

Stefan Wild

Enhancing Security in Managing Personal Data by Web Systems

Doctoral Dissertations in Web Engineering and Web Science
Volume 3

Prof. Dr.-Ing. Martin Gaedke (Series Editor)

Stefan Wild

**Enhancing Security in Managing
Personal Data by Web Systems**



TECHNISCHE UNIVERSITÄT
CHEMNITZ

Universitätsverlag Chemnitz
2017

Impressum

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Angaben sind im Internet über <http://dnb.d-nb.de> abrufbar.

Technische Universität Chemnitz/Universitätsbibliothek

Universitätsverlag Chemnitz

09107 Chemnitz

<http://www.tu-chemnitz.de/ub/univerlag>

readbox unipress

in der readbox publishing GmbH

Am Hawerkamp 31

48155 Münster

<http://unipress.readbox.net>

ISSN 2199-5354 print - ISSN 2199-5362 online

ISBN 978-3-96100-010-4

<http://nbn-resolving.de/urn:nbn:de:bsz:ch1-qucosa-217284>



TECHNISCHE UNIVERSITÄT
CHEMNITZ

Department of Computer Science
Distributed and Self-Organizing Systems Group

Enhancing Security in Managing Personal Data by Web Systems

Dissertation

submitted in fulfillment of the
requirements for the degree of

Doktoringenieur (Dr.-Ing.)

by

Dipl.-Inf. Stefan Wild

*August 27, 1983 in Reichenbach

December 9, 2016

Dissertation Committee:

Prof. Dr.-Ing. Martin Gaedke

Prof. Dr. rer. nat. Hannes Hartenstein

Dipl.-Inf. Stefan Wild

Enhancing Security in Managing Personal Data by Web Systems

Dissertation Committee:

Prof. Dr.-Ing. Martin Gaedke (Technische Universität Chemnitz),

Prof. Dr. rer. nat. Hannes Hartenstein (Karlsruher Institut für Technologie)

Submitted on August 18, 2016

Defended on December 9, 2016

Technische Universität Chemnitz

Department of Computer Science

Distributed and Self-Organizing Systems Group

Straße der Nationen 62

09111 Chemnitz

In gratitude to my family
for their encouragement,
patience and support.

Abstract

Web systems have become an integral part in daily life of billions of people, with social web applications taking on an increasingly important role among them. *Social* is a key characteristic modern web projects need to feature in order to be successful in the social age. To benefit from an improved user experience, individual persons are continually invited to reveal more and more personal data to web systems.

With a rising severity of attacks on web systems, it becomes evident that their security is inadequate for the amount of personal data they accumulate. Numerous risk and threat reports indicate that social media has become a top-ranking attack target, with climbing impacts, with ramifications beyond individuals and with a booming black market to trade compromised user accounts and leaked personal data. Attackers hereby profit by a poor consideration of information security during design and runtime of web systems and involved applications and services. There is great uncertainty and low transparency about protection, circulation and use of personal data by third parties.

Motivated by the positive developments a solution to this problem would imply for individual persons, companies and governments, the purpose of

the dissertation is to enhance information security in managing personal data by web systems. Five research questions and a set of objectives operationalize the purpose. To holistically address these objectives with respect to the suitability of state-of-the-art technologies, the dissertation proposes a solution architecture and three dedicated research contributions. While the solution architecture establishes the foundation for a more secure management of personal data by web systems, the research contributions represent complementary components for protecting personal data against unwanted data disclosure, tampering and use without the actual data owner's intent or knowledge. These components enable seamless integration and combination, and contribute to assure quality and maintainability through relying on preexisting artifacts only. By conducting an objectives-based evaluation, this dissertation verifies to what extent the proposed solution as a whole has 1) achieved the purpose, 2) attained answers to the research questions, and 3) contributed to the overall objective of an increased protection of privacy and a reduction of personal data disclosures through attacks on web systems. The dissertation concludes with interpreting the evaluation results and providing an outlook towards future work.

Acknowledgment

I like to take the opportunity to express my gratitude to all who accompanied me on my journey of becoming a PhD, with special thanks to the following persons:

My supervisor Prof. Dr.-Ing. Martin Gaedke for offering me the chance to join his VSR research group, for giving me enough space to pursue and integrate my own ideas, and for guiding me through the doctoral studies until this point of achievement. His always enthusiastic, motivating and future-driven nature led me back to Chemnitz in order to successfully carry on with my research on the spot.

Prof. Dr. rer. nat. Hannes Hartenstein for inviting me to talk about my research results as well as for his willingness and effort to review this work.

All members of the VSR research group for the always open and friendly atmosphere, the thoughtful discussions, the constructive criticism and the many nice, personal and funny moments we shared. Here, I like to especially mention Sebastian Heil, Michael Krug, Dr.-Ing. Max Speicher, Alexey Tschudnowsky, Fabian Wiedemann, Bahareh Zarei Mohammadzadeh

as well as Dr.-Ing. Jörg Anders and Ralph Sontag, and, not to forget, my former co-workers Hendrik Gebhardt, Dr.-Ing. Matthias Heinrich, Sven R. Kunze and Frank Weinhold. Prof. Dr.-Ing. Martin Gaedke did a great job in putting the VSR team together.

Markus Ast, Falko Braune, Dominik Pretzsch, Michel Rienäcker and, last but in no way the least, Anna Scholtz for the work they accomplished in diverse student and research assistances. Not only did they contribute to proof-of-concepts, but also to research papers this dissertation partially incorporates. I like to honorably mention Marco Drechsel and Sebastian Müller for their last minute support actions that added to the excellent rating of the EC FP7 project OMELETTE, where this work has its origins in. Furthermore, I am grateful and proud that also M. Földner, O. Grund, C. Gürtler, R. John, P. Petrenko, M. Peuß, R. Plarre, C. Pusch, K. Reichel and M. Urban successfully completed their student projects in the context of this work.

My wife Sandra, my family and those of my friends I have not mentioned so far for their ongoing encouragement, appreciation, understanding, love and care.

Dissemination

In order to substantiate this work and comply with high academic standards, parts of the research results this dissertation is based on have been disseminated before. All relevant publications (one book chapter, four journal articles, twelve conference articles, and two technical reports) are listed in the following:

Peer Reviewed Book Chapters

Satzger, B.; Zabolotnyi, R.; Dustdar, S.; Wild, S., et al. (2014): “Toward Collaborative Software Engineering Leveraging the Crowd”. In: *Economics-Driven Software Architecture*. Ed. by Mistrik, I.; Bahsoon, R.; Kazman, R.; Zhang, Y. 1st Edition. Elsevier. Chap. 8, pp. 159–182. ISBN: 978-0-12-410464-8.

Peer Reviewed Journal Articles

Wild, S.; Gaedke, M. (2014): “Utilizing Architecture Models for Secure Distributed Web Applications and Services”. In: *it - Information Technology* 56.3: *Architecture of Web Application / René Peinl*. Ed. by Molitor, P., pp. 112–118. ISSN: 1611-2776.

Wild, S.; Wiedemann, F.; Heil, S.; Tschudnowsky, A.; Gaedke, M. (2015): “ProProtect3: An Approach for Protecting User Profile Data from Disclosure, Tampering, and Improper Use in the Context of WebID”. In: *Transactions on Large-Scale Data- and Knowledge-Centered Systems*. Lecture Notes in Computer Science 8990: *Special Issue on Big Data and Open Data XIX*. Ed. by Hameurlain, A.; Küng, J.; Wagner, R.; Bianchini, D., et al., pp. 87–127. ISSN: 0302-9743.

Ast, M.; Wild, S.; Gaedke, M. (2014): “Efficient Development of Progressively Enhanced Web Applications by Sharing Presentation and Business Logic Between Server and Client”. In: *Journal of Web Engineering* 13.3 & 4: *Component-Based, Client-Oriented Web Engineering. Issues, Advancements and Opportunities*. Ed. by Daniel, F.; Dolog, P.; Li, Q., pp. 223–242. ISSN: 1540-9589.

Chudnovskyy, O.; Wild, S.; Gebhardt, H.; Gaedke, M. (2012): “Data Portability Using WebComposition/Data Grid Service”. In: *International Journal on Advances in Internet Technology* 4.3 & 4, pp. 123–132. ISSN: 1942-2652.

Peer Reviewed Conference Articles

Wild, S.; Ast, M.; Gaedke, M. (2013): “Towards a Context-Aware WebID Certificate Creation Taking Individual Conditions and Trust Needs into Account”. In: *Proceedings of the 15th International Conference on Infor-*

- mation Integration and Web-based Applications & Services (iiWAS2013)*. Ed. by Weippl, E.; Indrawan-Santiago, M.; Steinbauer, M.; Kotsis, G.; Khalil, I. IIWAS '13. Vienna, Austria: ACM, pp. 532–541. ISBN: 978-1-4503-2113-6.
- Wild, S.; Braune, F.; Pretzsch, D.; Rienäcker, M.; Gaedke, M. (2014): “Tamper-Evident User Profiles for WebID-Based Social Networks”. In: *Web Engineering*. Ed. by Casteleyn, S.; Rossi, G.; Winckler, M. Vol. 8541. Lecture Notes in Computer Science. Springer, pp. 470–479. ISBN: 978-3-319-08244-8.
- Wild, S.; Chudnovskyy, O.; Heil, S.; Gaedke, M. (2013a): “Customized Views on Profiles in WebID-Based Distributed Social Networks”. In: *Web Engineering*. Ed. by Daniel, F.; Dolog, P.; Li, Q. Vol. 7977. Lecture Notes in Computer Science. Heidelberg: Springer, pp. 498–501. ISBN: 978-3-642-39199-6.
- Wild, S.; Chudnovskyy, O.; Heil, S.; Gaedke, M. (2013b): “Protecting User Profile Data in WebID-Based Social Networks Through Fine-Grained Filtering”. In: *Current Trends in Web Engineering*. Ed. by Sheng, Q. Z.; Kjeldskov, J. Vol. 8295. Lecture Notes in Computer Science. Springer, pp. 269–280. ISBN: 978-3-319-04243-5.
- Wild, S.; Gaedke, M. (2009): “WebComposition/EMS: A Value-Driven Approach to Evolution”. In: *ICWE'09 Doctoral Consortium*. Ed. by Rossi, G.; Iturrioz, J. CEUR Workshop Proceedings. ISSN: 1613-0073. Onekin Research Group; University of the Basque Country, pp. 39–43.
- Ast, M.; Wild, S.; Gaedke, M. (2013): “The SWAC Approach for Sharing a Web Application’s Codebase Between Server and Client”. In: *Web Engineering*. Ed. by Daniel, F.; Dolog, P.; Li, Q. Vol. 7977. Lecture Notes in Computer Science. Heidelberg: Springer, pp. 84–98. ISBN: 978-3-642-39199-6.

- Braune, F.; Wild, S.; Gaedke, M. (2014): “Using Linked Data for Modeling Secure Distributed Web Applications and Services”. In: *Web Engineering*. Ed. by Casteleyn, S.; Rossi, G.; Winckler, M. Vol. 8541. Lecture Notes in Computer Science. Springer, pp. 540–544. ISBN: 978-3-319-08244-8.
- Heil, S.; Wild, S.; Gaedke, M. (2014a): “Collaborative Adaptive Case Management with Linked Data”. In: *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*. WWW Companion '14. Seoul, Korea: International World Wide Web Conferences Steering Committee, pp. 99–102. ISBN: 978-1-4503-2745-9.
- Heil, S.; Wild, S.; Gaedke, M. (2014b): “CRAWL-E: Distributed Skill Endorsements in Expert Finding”. In: *Web Engineering*. Ed. by Casteleyn, S.; Rossi, G.; Winckler, M. Vol. 8541. Lecture Notes in Computer Science. Springer, pp. 57–75. ISBN: 978-3-319-08244-8.
- Rienäcker, M.; Wild, S.; Gaedke, M. (2014): “Building Bridges between Diverse Identity Concepts Using WebID”. In: *Web Engineering*. Ed. by Casteleyn, S.; Rossi, G.; Winckler, M. Vol. 8541. Lecture Notes in Computer Science. Springer, pp. 498–502. ISBN: 978-3-319-08244-8.
- Scholtz, A.; Wild, S.; Gaedke, M. (2015a): “Scope-Aware Delegations in Distributed Social Networks”. In: *Engineering in the Web in the Big Data Era*. Ed. by Cimiano, P. et al. Vol. 9114. Lecture Notes in Computer Science. Springer, pp. 709–712. ISBN: 978-3-319-19890-3.
- Scholtz, A.; Wild, S.; Gaedke, M. (2015b): “Systematic Composition of Web-based Applications with Focus on Security”. In: *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services*. iiWAS '15. ACM, pp. 637–641. ISBN: 978-1-4503-3491-4.

Technical Reports

Chowdhury, S. R.; Daniel, F.; Tschudnowsky, A.; Wild, S.; Gaedke, M., et al. (2013): *Final Specification of Mashup Description Language and Telco Mashup Architecture. Deliverable 2.3*. URL: http://www.ict-omelette.eu/c/document_library/get_file?p_l_id=48742&folderId=165188&name=DLFE-12333.pdf (visited on July 1, 2016).

Tschudnowsky, A.; Wild, S.; Gaedke, M., et al. (2013): *Final Dissemination and Standardization Report. Deliverable 8.5*. URL: http://www.ict-omelette.eu/c/document_library/get_file?p_l_id=48742&folderId=157989&name=DLFE-12320.pdf (visited on July 1, 2016).

Contents

1	Introduction	1
1.1	Situation	2
1.2	Central Problem	6
1.3	Motivation	12
1.4	Purpose	14
1.5	Outline	17
2	Challenges	23
2.1	Foundations	24
2.2	Problem Analysis	31
2.3	Objectives	53
2.4	Summary	66
3	State of the Art	67
3.1	Categorization	68
3.2	Criteria	70
3.3	Analysis Results	78
3.4	Summary	118
4	Enhanced Security in Managing Personal Data	121
4.1	Design	122

4.2	Architecture and Process	132
4.3	Key Components for Enhanced Security	141
4.4	Proof-of-Concept Platform	149
4.5	Summary	150
5	Context-Aware Control	153
5.1	Analysis	154
5.2	Development	162
5.3	Evaluation	176
5.4	Summary	179
6	Tamper-Evidentness	181
6.1	Analysis	182
6.2	Development	187
6.3	Evaluation	199
6.4	Summary	204
7	Fine-Grained Filtering	207
7.1	Analysis	208
7.2	Development	213
7.3	Evaluation	224
7.4	Summary	228
8	Overall Evaluation	231
8.1	Characteristics	232
8.2	Procedure	232
8.3	Results	234
8.4	Summary	256
9	Conclusion	259
9.1	Contributions	260
9.2	Review	263

9.3 Outlook	266
A LFA Artifacts	269
Glossary	283
Bibliography	289
List of Acronyms	315
List of Figures	321
List of Listings	323
List of Symbols	325
List of Tables	329

Introduction

1

To introduce this dissertation, the first chapter begins with describing today's landscape of web systems with a particular emphasis on the social web in Section 1.1. Having situated this work through presenting the general domain our research focuses on, Section 1.2 then shows prevalent issues existing in this context and consolidates them to the central problem statement. On that basis Section 1.3 illustrates the impact of the central problem in order to highlight our motivation for finding a solution. Section 1.4 condenses the motivation to define the purpose of this work, which is further operationalized by a set of research questions and a set of anticipated research contributions. Finally, Section 1.5 outlines the organization of the work to achieve the purpose.

1.1 Situation

Web systems¹ have become an integral part in daily life of billions of people, with social web applications taking on a more and more important role among them in the last couple of years (Nielsen and NM Incite, 2012). Supporting a broad spectrum of user activities from social networking and self-presentation over content and opinion sharing to collaborative editing and crowdfunding, social web applications enable people to express themselves, communicate and team up with each other (Appelquist et al., 2010; Kietzmann et al., 2011). No matter whether dealing with products, contents or activities, *social* is a key characteristic modern web systems need to feature for being and remaining successful in the social age (Azua, 2009; Bell, 2009). With more data stored in the cloud and accessible online, it is required for individual persons, businesses and governments to handle massive quantities of personal data securely (Verizon, 2014). Personal data includes any information relating to a (human) entity, where this particular entity is either identified or identifiable without unreasonable effort (European Parliament, 1995; FRA, 2014).

With the advent of e-mail almost forty years ago, social application software started its triumph and steadily gained more ground down to the present day (Porter, 2008). The World Wide Web (WWW or Web) is a paragon of social software that laid the foundation for a multitude of applications, which are running on web servers and are accessible over networks via standardized interfaces by heterogeneous agents including web browsers and other services. Holding onto its original intention of being a facilitator of information exchange among users (Kaplan and Haenlein, 2010), the WWW completed its shift towards a platform whose applications foster

¹Throughout this dissertation, we use the term *web system* to denote an orchestration of entities, like web applications and web services, to provide one or multiple more complex services with extended functionality.

the creation of new content whilst taking account of how well content is received and shared with others (Kietzmann et al., 2011). Driven by the Web 2.0 movement, essential building blocks have been provided to facilitate both user participation and contribution rather than mere information consumption, which was the de facto standard before (Appelquist et al., 2010; Kaplan and Haenlein, 2010). With a significant and lasting impact on the way people inform themselves and acquire knowledge, user-generated content (UGC) emerged out of this movement as a prevalent and omnipresent element that deeply shaped the media landscape (Daugherty et al., 2008; Webster, 2010). Deployed on the technological basis introduced by the Web 2.0, social media applications serve more and more people to create and exchange UGC at a global scale (Nielsen and NM Incite, 2012). By increasing accessibility and lowering entry barriers for contributing and consuming UGC, the rise of the mobile sector largely accelerated this trend (Kaplan and Haenlein, 2010).

Beyond serving as a necessary mean for recognizing content contributors, a user's identity took on a special place in the developments towards the social web (Kietzmann et al., 2011). While user profile pages were first meant as a place to store personal data needed to use services more efficiently, e.g., auto-completion of address data, a growing number of web systems focused on identity data of individual persons as a central theme. Enabling to identify and establish relationships between people over the web, the social web provided a hotbed for many new applications in recent years (Appelquist et al., 2010). Through linking identity with activity, content and products, social web applications allow for user recognition, attribution and building trust (Bell, 2009). A prime example are online social networks (OSNs). Founded on the identity theme, the IT companies behind centralized OSNs gained high market coverage, manifested their business position through initial public offerings and rank among today's most valuable global players (Appraisal Economics, 2014; Winkels, 2013).

Judged by their ability to attract and retain individuals, both the current size and the growth potential of the user base determine the high valuations of social media companies (Anders, 2014; Appraisal Economics, 2014). Rather than selling physical products, this so-called *invisible economy* or *attention economy*, i.e., the economy constituted by social media companies, adds their value primarily from the amount of social capital it acquires and binds (Chayka, 2014; Webster, 2010). Social capital originates from the social connections between individuals, their access to resources, and their positions in a social network gained through self-presentation and interactions with other users, e.g., messaging and sharing (Burke et al., 2011; N. B. Ellison et al., 2011). Analyzing the consumption habits of their users enables to steadily provide them with new custom-tailored content, which is often generated by the users themselves (Daugherty et al., 2008). In order to benefit from more relevant content and, hence, an improved experience, individuals are invited to reveal further personal data (Delo, 2014; Krishnamurthy and Wills, 2008). Not only does this include information on their social capital, their opinions and their sentiments, but also personally identifiable information (PII), i.e., any information that allows for detecting the identity of an individual person. Along with content customizations, this assists in creating appealing advertisements that properly factor in the user's individual characteristics disclosed to or inferred by web systems (Baden et al., 2009; Webster, 2010). Seeking to rise the average revenue per user, social media companies accumulate personal data of their users (Appraisal Economics, 2014; Krishnamurthy and Wills, 2008). In particular young people provide web systems with large amounts of information relating to them as individuals (EC, 2011; Nielsen and NM Incite, 2012). The more personal data individuals expose, the more accurate the recommendations and the higher the expected revenue will be through targeted advertisements or through a potential acquisition of the company (Bria, 2012; Webster, 2010). Although the worth is not distributed equally among individual

persons, each active user contributes to the overall social capital managed by social web applications (Appraisal Economics, 2014).

Striving for keeping the valuations high through preventing the drain of up-to-date information on social capital and other personal data, especially social web applications did not pay much attention to data portability and, thus, largely evolved as personal data silos (Bojars et al., 2008; Darwell, 2012). Marked by the heterogeneity of web systems flourished on the social media landscape, individuals were encouraged to create new identities based on more or less the same main identity and enter related data over and over again (Appelquist et al., 2010; Webster, 2010). In the past, individual persons received *real* identity cards primarily from a modest number of organizations from the public or private sector, like government-issued or corporate identification cards, after providing solid proof and close examination of personal attributes. Today's users, however, have to cope with numerous digital identities specific to particular web systems. In consequence of establishing many redundant, distributed, application-specific stores for information like name, address or account data, users are repeatedly asked to provide credentials for their identity data in order to enable protection against unauthorized access and use. Most modern web systems rely on authentication via username/password pairs for this purpose and burden individual persons with the task to either remember their credentials or keep them safe from unwanted parties respectively (Florêncio and Herley, 2007). By consolidating user identities at a central place and enabling controlled access for web systems, federated identity tries to mitigate that matter (Hackett and Hawkey, 2012). This is particularly attributed to the option of enabling users to plausibly legitimize to a federation of multiple web systems through a single authentication, known as single sign-on (SSO). Social login solutions, as a variant of SSO, dominate today and gave further leverage to global players in the social media sector.

While many developments towards more sociality in the web are positive, they did not come along without negative implications as explained in the next section.

1.2 Central Problem

With a rising severity of attacks, it becomes evident that the security² of today's web systems is inadequate for the amount of personal data they accumulate. Numerous risk and threat reports, including (Horacek, 2013; Symantec, 2014; Verizon, 2014), show that social media has become a top-ranking attack target, with climbing impacts, with ramifications beyond single individuals and with a booming black market for trading compromised user accounts and leaked personal data. Accompanied by an increasing accumulation of personal data, we argue that an insufficient consideration of protective measures by web systems at design time and runtime is a major cause for the growing number, extent and impact of successfully orchestrated attacks. Not only do these attacks affect a couple of individuals over a short period, but a plethora of users with hard to foreseen consequences in the long-term.

In order to substantiate the problem statement, the following paragraphs give reasons for our claim by 1) illustrating the particular target characteristic of web systems caused by an accumulation of personal data obtained from their users, by 2) describing resulting attacks in terms of magnitude, scope, distribution, impact, spatial and physical independence, and by 3) showing that protective measures taken by web systems are only insufficient.

²Unless otherwise noted, we refer to *information security* when using the term *security*.

By analogy with banks that deposit monetary capital for their customers³, web systems store information on the social capital and other personal data their users entrusted them with. Promising customers to make payable investments by utilizing provided resources and capabilities, e.g., multiply their capital or expand their social network, both banks and web systems accumulate a specific type of capital on behalf of their customers. Like banking federations and major banks, also federations among web systems offer a number of benefits for their customers, e.g., unified access to a larger product portfolio, but bear risks associated to an increased accumulation and centralized management of capital.

An accumulation of capital, whether it is money or information, can represent an attractive as well as lucrative target for attacks as soon as the potential reward outweighs the necessary effort (Horacek, 2013). In the context of information security, an attack represents “any kind of malicious activity that attempts to collect, disrupt, deny, degrade, or destroy information system resources or the information itself” (CNSS, 2010). Similar to the real world, where bank robbers target the monetary capital customers committed to a bank, criminals target personal data, including information on social capital, web systems manage for their users through exploiting potential infrastructural, organizational or personnel weak points of the entity the capital is entrusted to (Müller, 2008). Due to specially focusing on their users’ personal data, social web applications present a top target in this context. The number of incidents relying on social tactics to expose data are steadily climbing since several years and are only surpassed by malware and hacking as of this writing (Verizon, 2014). In line with a growing amount of attacks on web systems, security companies discovered a rapid increase of breaches with more than 10 million identities revealed, including data on real names, birth dates, government-issued identification numbers, home

³Although characterized by domain-specific usages, we treat *customer* and *user* as synonyms in this work.

addresses, medical records, phone numbers, financial information, email addresses, user names and passwords, and insurance data (Verizon, 2014).

Information services like *DataBreaches.net*, *LeakedIn* or *‘-have i been pwned?’* provide an overview of an alarming extent of attacks (DataBreaches.net, 2016; Hunt, 2016; LeakedIn, 2016). Trying to make individuals aware about the risks of data loss and identity theft, such information services also show the sheer magnitude of breaches affecting both users and web systems (McCandless and Evans, 2013). The latter service, for example, lists breaches which altogether entailed more than 170 million publicly available data disclosures as of this writing.

While the motivation of criminals to enrich themselves or to harm entities holding the capital stays the same independent from the type of capital, transforming this very motivation into concrete actions is largely supported by the characteristics of the web. According to (Verizon, 2014), a significant share of attacks on web systems is chiefly driven by financial motives intending to convert disclosed data to money. Here, external threat actors are dominant, but internal ones do not play an inconsiderable role (Verizon, 2014). Criminals typically target user credentials for this purpose, i.e., user names, email addresses and passwords, by utilizing techniques like phishing, brute-force password guessing or application-level attacks that aim at breaking into the user management or bypassing authentication routines of web systems (Bogart, 2013; Reisinger, 2014). Among other things, attackers are assisted in their malicious practice of finding suitable victims by deeply exploiting information on identity and connections available within user profiles (Horacek, 2013; Verizon, 2014). Even though individuals are informed to mitigate risks by restricting the amount of personal data they make publicly available, it is doubtful whether web systems are also taking sufficient protective measures to safeguard personal data entrusted to them (EC, 2011; Verizon, 2014).

Attackers often benefit from a poor consideration of information security during design or runtime of web systems, e.g., storing email addresses and password hints as unencrypted plain text in databases (Hunt, 2014), using unsalted password hashes (Finkle and Saba, 2012), granting operators too many privileges (Ilyin, 2014), or missing to sustainably address known vulnerabilities, such in the case of cross-site scripting (XSS) (Brandom, 2014). While the high heterogeneity among web systems can be considered as positive, the associated lack of knowledge about underlying design principles and employed models does not help to ensure individual persons that their data is kept safe against exploitation they do not comply with. There is an insufficient awareness of the significance to ensure availability, confidentiality and integrity of information (systems) by protecting them against unauthorized access and exploitation (CNSS, 2010; Venter and Eloff, 2003). That is, security is often treated only as an afterthought and not as a first thought during design time and runtime of web systems and involved web applications and web services. The conduct of a considerable number of companies to encourage individuals to disclose as much personal data as possible via web systems is not in line with the frequently cited laws of identity principles published in (Cameron, 2005). There is great uncertainty and low transparency about the protection, circulation, and use of personal data by third parties (EC, 2011). Contrary to the perspective described in (Jordan et al., 2003) a decade ago, we think that 1) the absence of both physical and local presence of attackers, 2) an attack magnitude beyond a single unit, and 3) a fast and global distribution and exploitation of insights on latent weak points are additional support factors compared to other type of attacks, which entail further leverage that directly originates from the characteristics of the web.

Taking a more detailed look at the support factors for attacks on web systems, it is apparent that in comparison to bank robberies, where timing and location are critical factors, cybercriminals typically do not need to

have physical access or be present in person nearby the server(s) that are running the web system in question (Müller, 2008). Both the spatial independence of attacks and the freedom of locality for attackers allow for a more concealed preparation, execution and escape as well as a reduction of the attacker's inhibition level through an increased distance in a psychological, physical, geographical and legal sense.

The continuous evolution of computer systems and networks facilitates utilizing computational capabilities also for malicious purposes. This created a competition against the human will and capability to remember things, which results in increasingly challenging knowledge-based authentication. For example, as of writing this dissertation, specialized tools like *hashcat* (Steube, 2016) enable attackers to crack common passwords in half a day⁴ using conventional hardware (Lystad, 2013). Hereby, they profit from both central processing unit (CPU)- and graphics processing unit (GPU)-accelerated computation. The means thus provided assist in attacking multiple targets in parallel (Horacek, 2013), instead of one single target after another as in the case of bank robberies (Müller, 2008).

Today's possibilities for rapid and global distribution of information further contribute to potentially play knowledge on latent weak points of web systems into the wrong hands. According to (Horacek, 2013), the value associated with gaining access to social media accounts created a black market. Enabling attackers to monetize social media vulnerabilities as well as compromised user accounts, black market prices vary depending on the type of data disclosure, starting from basic identity data over health records to newly leaked credit card information (Holt and Smirnova, 2014).

⁴Eight character long passwords consisting of letters and numbers. Verification of 3,67 billion Secure Hash Algorithm (160 bits) (SHA-1) hash values per second.

The impacts of attacks on web systems are not limited to their particular domain, but potentially bear negative implications for both affected individuals and companies with respect to reputation, trust and finance (Dane, 2012; Horacek, 2013). Leaked PII can be matched and combined by criminals with existing data sets and exploited for further attacks involving penetration of protection phrases or social engineering tactics like impersonating users. Notwithstanding the non-materialistic character of knowledge fed by data disclosures, it can seriously impair other types of capital when used in the wrong hands, e.g., cause financial losses or persecution of minorities due to conflicting views or beliefs (Krishnamurthy and Wills, 2008; Reisinger, 2014). It is not unusual that the consequences of a declining reputation and trust in web systems manifest in a drain of users as they fear ramifications in the offline world (Horacek, 2013). While public relations (PR) managers were responsible yet reluctant on reporting incidents of web systems in a coordinated way in the past, today's social media allows for a more direct, fast, and uncoordinated spread of information by users of affected web systems and, therefore, reinforces implications of bad news (Shalal-Esa, 2012).

Unlike the protection of physical values, which is an intrinsic part of human nature, the measures to prevent personal data and, especially, social capital from unwanted exploitation lag behind the ones that are in place for safeguarding other types of capital like monetary capital or human capital. Further complicated by the fact that personal data is non-anonymous, non-physical and bound to the individuals it relates to, there is currently no proper mean to compensate affected people once personal data is leaked, copied or exploited for malicious purposes – it is inappropriate to create new identities, as it would greatly interfere with recognizability. Individuals are aware of this problem to some degree (EC, 2011), but they seem to resigned on that matter, probably due to the fact that they cannot fully anticipate the consequences the problem entails for their future. Not only are user data sets especially valuable for advertisement, marketing and

insurance companies to generate potential leads (Bria, 2012), but also for criminals following more dubious business models.

In conclusion, personal data including information on social capital has a particular value that needs to be protected adequately both by individual persons and by the web systems the data is entrusted to. While this problem description represents the starting point and the connecting link for the research and highlights the importance of the matter (Creswell, 2012; Leedy and Ormrod, 2010), the motivation statement presented in the next section shows why the central problem is worth addressing.

1.3 Motivation

The motivation to approach the central problem manifests in the positive developments a potential solution would imply for individuals, web companies and democratically elected governments⁵. The relevance for them⁶ becomes apparent, once we have outlined how the problem impacts them:

Individual persons are primarily challenged by the problem in the sense that their personal data can fall into wrong hands and used for purposes they do not agree to, like monetization by criminals on black markets (Holt and Smirnova, 2014; Horacek, 2013). The uncertainty about circulation and exploitation of leaked personal data makes it difficult for individual persons to anticipate the aftermaths of data disclosures with respect to moment, location, type, extent and involved people (EC, 2011; Müller,

⁵For the sake of brevity we just use the terms *company*, when referring to an information technology (IT) organization that provides certain services over the web in exchange for a compensation, and *government*, when referring to a government with democratic attributes.

⁶These groups have been identified through a stakeholder analysis with results detailed in Subsection 2.2.2.

2008). Organizations, which are formed by groups of persons to pursue collective purposes, are secondarily affected by the problem. On the one hand, there are companies that have created their business models around potential users. Not only can data disclosures cause a loss of hard-earned trust in companies and damage the reputation of brands, but also induce a decline of users and earnings through missing returns on advertisements or falling stock prices (Dane, 2012; Gangewere, 2013). On the other hand, there are governments of states or state unions, like the European Union (EU). They are challenged by the problem because data disclosures through attacks on web systems increasingly cast doubt on established legal frameworks and legally enforceable rights of individual citizens (EC, 2012). While governments are responsible for representing and protecting their population, a not inconsiderable ratio of citizens, e.g., 58% of Europeans, does not see alternatives to disclosing personal data to third parties when they want to use services or obtain products, regardless of associated risks (EC, 2011; WEF, 2011).

When the thus described impacts of the problem are not approached in time, the current negative trend of insufficiently taking account of information security in the design of web systems and during their runtime is expected to continue with negative effects rising in severity and provide further leverage for criminals.

Aiming at inverting these negative impacts, an ideal solution would offer individual persons control about use and safekeeping of their personal data, i.e., enabling them to decide whom exactly they want to make their data available to and detect who is actually accessing their data. Consequently, such solution would also enable individuals to regain ownership of their personal data. Relieving web systems from the need to store identity data they do not own would allow companies for refining their business models to convince individuals mainly by a useful and unique feature set rather

than the mere fact that many acquaintances are already using a certain application, probably due to missing migration options. Reducing the accumulation characteristic of web systems would therefore make them less attractive for criminals as they cannot retrieve such large amounts of information on social capital anymore. This is also in line with the ambitions of governments, like the European Commission (EC), to make companies accountable for dealing with personal data and to urge them to immediately inform users as well as authorities on incidents of web systems that compromised rights on personal data of their citizens (EC, 2012). Beyond notifying affected entities after a breach, the EC, for example, strives for enforcing the principles of “privacy by design” and “privacy by default” in order to prompt companies to build mechanisms to safeguard personal data into their products beginning from the earliest stage of development (EC, 2012).

Such ideal is difficult to reach in practice, yet it indicates the direction for possible endeavors to meet the challenges arising from the problem and it is therefore the leitmotif for this dissertation, which is formalized in the following section.

1.4 Purpose

Based on the motivation, we set the purpose of this work in enhancing security in managing personal data by web systems for the mutual benefit of individual persons, companies and governments. With this tangible target, we aim at approaching the central problem stated in Section 1.2, whilst taking into account the principles of managing personal data established in the privacy framework (OECD, 2013) by the Organisation for Economic Co-operation and Development (OECD) as well as the vision of the World Economic Forum (WEF) to value personal data of individuals similar to their monetary capital (WEF, 2011). Treating the security

of personal data as a core idea, protection has to be applied holistically by web systems at design time and during runtime. While security is a broad discipline with multiple dimensions, realms and layers, we set the scope of this dissertation by solely focusing on security-related aspects of web systems concerning personal data.

To operationalize the purpose, as suggested in (Creswell, 2012), we narrow it using five hypotheses to be answered by this dissertation:

Hypothesis 1: Modeling. There is an appropriate, interoperable, interpretable and supportive way to model web systems at different abstraction levels that also puts a strong emphasis on security involving trust and invocation relationships.

Hypothesis 2: Description and Identification. It is practicable to describe and identify web systems, applications and services, as a basis for applying protective measures, by means adequate to describe and identify other entities like individual persons.

Hypothesis 3: Ownership. There is a way to enable individuals to maintain ownership about their personal data, especially personally identifiable information, and allow controlled access for third parties incl. other users, web applications and web services.

Hypothesis 4: Delegation. It is feasible to authorize entities, including individuals as well as web applications and web services, to use web applications and web services within a defined scope on behalf of other entities in highly distributed environments.

Hypothesis 5: Protection. There is a way to protect personally identifiable information against data disclosure without the corresponding individual's knowledge or intent, against tampering and (identity) theft, and against misuse by third parties including users, web applications and web services.

While hypotheses are suitable for making predictions about expectations to be verified through quantitative research, research questions allow for raising issues to be addressed also through qualitative research (Creswell, 2012). A hypothesis deals with the mere possibility of the matter (a question of *what*). In contrast, a research question focuses on the exact attainment (a question of *how*). As a consequence, we can build a research question upon a hypothesis. That is, we will implicitly prove the underlying Hypotheses 1 to 5 by answering the following five research questions:

Research Question 1: Modeling. How to model web systems at different abstraction levels in an appropriate, interoperable, interpretable and supportive way that also puts a strong emphasis on security involving trust and invocation relationships?

Research Question 2: Description and Identification. How to describe and identify web systems, applications and services, as a basis for applying protective measures, by means adequate to describe and identify other entities like individual persons?

Research Question 3: Ownership. How to enable individuals to maintain ownership about their personal data, especially personally identifiable information, and allow controlled access for third parties incl. other users, web applications and web services?

Research Question 4: Delegation. How to authorize entities, including individuals as well as web applications and web services, to use web applications and web services within a defined scope on behalf of other entities in highly distributed environments?

Research Question 5: Protection. How to protect personally identifiable information against data disclosure without the corresponding individual's knowledge or intent, against tampering and (identity) theft, and against misuse by third parties including users, web applications and web services?

To attain answers to above research questions, this dissertation provides a set of four complementary research contributions. They are summarized below:

Research Contribution 1: Enhanced Security in Managing Personal Data. Enhancements to the security of personal data management by web systems, web applications and web services at design time and during runtime.

Research Contribution 2: Context-Aware Control. Increased awareness and control during identity creation, during identity use and in delegation scenarios.

Research Contribution 3: Tamper-Evidentness. Detection of a) identity theft, b) forgery of identity data and c) tampering of personally identifiable information.

Research Contribution 4: Fine-Grained Filtering. Creation, management and utilization of customized views on personal data depending on requesting entities.

These research contributions will assist in achieving the purpose of this work and, thus, contribute to increase the protection of privacy and to realize the secure by design vision, as further detailed in Chapter 2. The next section completes the first chapter by briefly introducing the research contributions and putting them in context with the other parts of this dissertation.

1.5 Outline

Chapter 1 introduced the subject of this work and, thus, the general domain our research focuses on. It showed our motivation for dealing with the central problem of web systems treating security only as an afterthought for

the amount of personal data they accumulate. Building on the motivation for solving the central problem, we set the purpose to hold out a tangible target for approaching the secure management of personal data by web systems, web applications and web services as a first thought throughout their entire life cycle. Not only did we operationalize the purpose by five hypotheses and five associated research questions but also by four dedicated research contributions. For organizing this dissertation, we employ renowned works on research methodology, including (Creswell, 2012; Leedy and Ormrod, 2010). We proceed from this point as depicted by Figure 1.1 and as indicated by the planned course of action in the following.

Chapter 2 clarifies the challenges this work has to meet. It therefore provides the necessary background and lays the foundations in order to ensure both an appropriate level of understanding and a systematic discussion within the dissertation. For analyzing the central problem, we first illustrate it by diverse scenarios and then divide it into more manageable units representing subproblems. Similar to the subproblems, which constitute the central problem, we present a set of corresponding objectives for achieving the purpose.

Chapter 3 analyzes the suitability of state-of-the-art technologies for solving the problem. For that reason, we transform the objectives into assessment criteria and verify their degree of fulfillment through the specific technologies. To reduce the verification effort and be prepared for future developments, we establish a categorization of relevant technologies which share similar characteristics.

Chapter 4 proposes a novel approach to enhance the security in managing personal data by web systems. In consideration of the analysis results obtained in Chapter 3, we devise a design that aims at addressing the challenges discussed in Chapter 2. Using the design we provide our main Research Contribution 1 in terms of the solution architecture and pro-

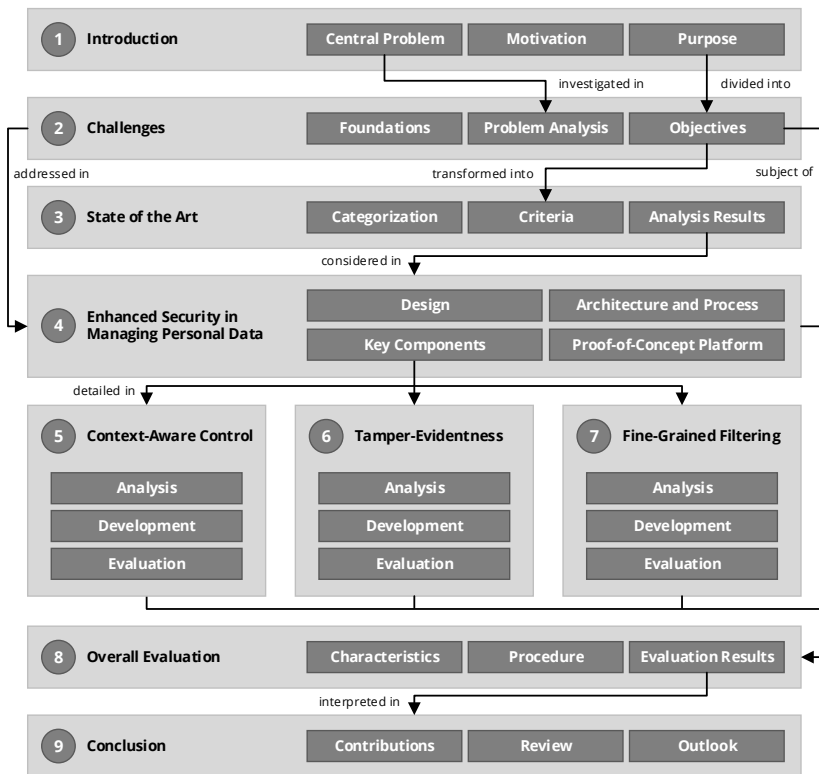


Figure 1.1: Organization of Dissertation

cess. On that basis we introduce three Research Contributions 2 to 4, each representing a particular component that extends the architecture by complementary facilities for enhanced information security. In order to create a technological foundation to implement, integrate and verify all research contributions, we present the proof-of-concept platform, on which we will rely on and build upon in the remainder of this work.

Chapters 5 to 7 substantiate the proposed solution by providing further explanations to Research Contributions 2 to 4. Each chapter describes a component encapsulating a research contribution with regard to domain-specific requirements and related work. For the development and the evaluation of each component, we take account of requirements as well as assessment results. Although all three components contribute to enhance security on their own, they allow for combination and complementation in order to increase their usefulness through exploiting synergies (cf. Chapter 8).

Chapter 8 evaluates to what extent the entire solution including all research contributions (Chapters 4 to 7) has achieved the objectives outlined in Chapter 2. To retrieve plausible evaluation results, we do not only define criteria to be validated and carefully select an appropriate procedure, but also consider all proof-of-concept implementations within a prototypical system that demonstrates the capabilities for enhanced security in managing personal data. Moreover, we examine how well all research questions have been attained and how well we innovated beyond the state of the art.

Chapter 9 concludes this dissertation by interpreting the evaluation results. Not only do we summarize the main research contributions and review what has been accomplished, but also offer an outlook towards the direction future work has to go.

Appendix A compiles the logical framework approach (LFA) artifacts we considered as necessary for conducting a logical analysis of the challenges in Chapter 2.

Characteristic for a clear, systematic and scientifically sound procedure, we catalog relevant terms used in the course of this dissertation in a glossary on page 283. In support of our argumentation, we furthermore give an overview of all underlying works in the bibliography on page 289. To distin-

guish between different levels of quality assured through methods such as peer review, we divide the bibliography into printed references and online references. Finally, we provide several type-specific indexes. By covering Chapters 1 to 9, appendices, glossary and bibliography, these indexes facilitate the lookup of acronyms on page 315, figures on page 321, code listings on page 323, mathematical symbols on page 325, and tables on page 329.

Challenges

2

To enable a systematic investigation and breakdown of the challenges arising from the problem stated in the Introduction, this chapter begins with providing fundamental information in Section 2.1. Thinking of a challenge as a difficult yet stimulating task (Merriam-Webster, 2016b) enables to distinguish two connected elements: a problem and an objective. While a problem refers to something difficult in the current situation, an objective refers to a stimulus of achieving the desired state in the future through addressing the problem. This concept promotes dealing with both elements individually in two separate yet interdependent sections. Being a “starting point for the research” and “a unifying thread that runs throughout all the elements of the research endeavor” (Leedy and Ormrod, 2010), Section 2.2 first illustrates the problem by diverse scenarios and then further details it by classifying related subproblems. As a counterpart to this problem classification, Section 2.3 details the purpose by classifying necessary objectives. Summarizing the challenges a solution has to meet, Section 2.4 concludes this chapter.

2.1 Foundations

Laying the groundwork for a solid discussion and examination of the challenges, Subsection 2.1.1 describes the research methodology and tools to approach the problem and 2.1.2 introduces important terms and definitions to foster a common understanding.

2.1.1 Research Methodology and Tools

To draw conclusions and answer the research questions presented in Section 1.4, we have to provide sufficient evidence (Ellis and Levy, 2009). Retrieving results as part of a study requires selecting a methodology that is suited for conducting research in consideration of the problem type (Leedy and Ormrod, 2010). As we can neither completely oversee nor universally control nor holistically drive a solution in the problem domain, we regard the central problem as a complex one, with uncertain future developments and with diverse stakeholders, who have partially contrary views on the matter. Responding to this type of problem requires applying a methodology that allows for 1) breaking the central problem down into more manageable subproblems to handle complexity, 2) making use of assumptions to deal with uncertainty at least to some degree, and 3) taking the needs of different stakeholders into account to offer an adequate solution.

Manifested in its direction toward the purpose defined on page 14, the objective-oriented nature of this work allows for regarding and treating it as a project. Being a “temporary endeavor undertaken to create a unique product, service, or result”, we can systematically approach the problem to achieve the purpose by making use of proven project management methodology, i.e., applying knowledge, skills, techniques and tools (PMI, 2009). While there are diverse approaches for managing projects, like PRINCE2®

or Project Management Body of Knowledge (PMBOK®) by PMI, we employ the project cycle management (PCM) in association with the logical framework approach (LFA) as a method for managing the research process. Here, LFA represents “an analytical tool for objectives-oriented project planning and management” that assists in facing problems characterized by complexity, uncertainty and conflicting stakeholder interests (NORAD, 1999). Devised in the 1960’s to support planning, management and evaluation of projects, LFA has been continuously evolved and is an obligatory part of EC’s PCM since 1993 (EC, 2004; Miklič, 2008). LFA is predestined for the domain of this work because it contributes to 1) clarify the purpose, 2) specify key elements, and 3) measure the success of the project (NORAD, 1999). Involving concepts for systematic analysis, sound organization, weakness identification and informed decision making, LFA defines how to run projects in a logical way (Miklič, 2008).

LFA, as a research tool, integrates into PCM by supporting four out of five stages within its cycle of operations, i.e., identification, formulation, implementation, evaluation and audit (EC, 2004). Even though LFA does not cover the programming stage in PCM, the elements⁷ included in this stage have already been discussed in Chapter 1 through situating this work and highlighting prevalent shortcomings. In addition to that, they are elaborated further in Chapters 8 and 9 through drawing conclusions from the evaluation results that potentially contribute to the subsequent programming stage. As we consider management and tool support for both the implementation stage and the audit stage as non-essential for projects at the scale of this work⁸, we primarily use LFA for identification, followed by formulation

⁷The actual strategy development elements are largely shaped by both institutional long-term visions and scientific standard procedure. They are therefore not fully elaborated in this dissertation’s scope.

⁸For example, there is no indispensable need for extensive contracting and progress monitoring.

and, finally, for evaluation. That is, we perform the implementation independently yet in the sequence defined by the cycle of operations in PCM.

Employing LFA in three designated stages enables to 1) analyze the existing situation to recognize potential objectives during identification, 2) plan the project by setting clear objectives and measurable outcomes during formulation, and 3) assess the degree of achieving the objectives during evaluation (EC, 2004). The remainder of this chapter addresses the identification and formulation, whereas Chapter 8 covers the evaluation stage.

The two main phases of LFA, namely analysis and planning, relate to the stages of identification and formulation in PCM (EC, 2004). In order to enable creating a sustainable project design based on the “most direct and essential causal relationships” identified before, the analysis phase precedes the planning phase in the cycle of operations as per PCM (NORAD, 1999). Figure 2.1 depicts the way how we proceed by taking PCM and LFA into account. We report analysis results obtained by identifying and characterizing potential major stakeholders and related key problems in Section 2.2, and discuss derived objectives and an appropriate strategy to achieve them in Section 2.3. In line with the strategy selection, the latter section also presents the “documented product of the analytical process”, which we use as a mean to organize and plan the further course of action (Miklič, 2008).

2.1.2 Terms and Definitions

In order to ensure a proper application of LFA, (EC, 2004) recommends establishing a common understanding about terminology before proceeding further. Building on (ITU, 2010; Sakimura, 2013), we therefore define important terms in the meaning we use them in this work and illustrate the relations between them in Figure 2.2.

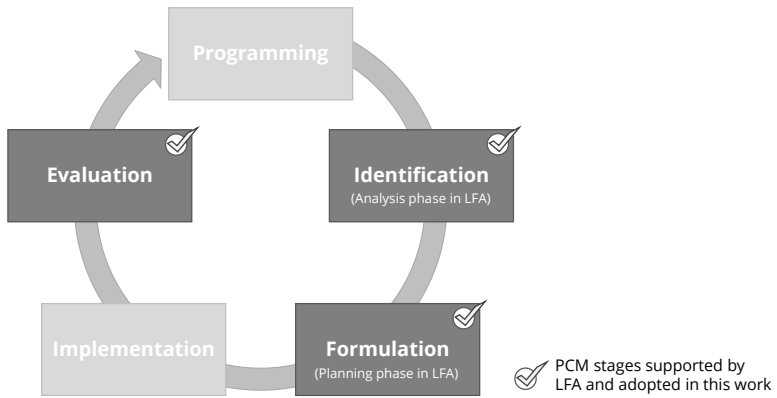


Figure 2.1: Methodology to Approach the Problem Using PCM and LFA

An *entity* $e \in E$ denotes “something that has separate and distinct existence” (ITU, 2010), where E is the set of all entities. That is, E includes different classes of both subjects and objects like persons, devices, accounts, software applications and services (Barker et al., 2012). It follows from the foregoing that a *user* is any entity that makes use of a *resource*, which in turn provides something of value (CNSS, 2010). When an entity acts on another’s behalf, it is referred to as an *agent* (ITU, 2010). A *user agent*, like a web browser, is therefore an entity that acts on behalf of a user for making use of resources. Particularly with regard to *delegation*, agents entrusted with authority, responsibility or a function are known as *delegates* and entrusting entities are called *delegators* (ITU, 2010). In this sense, trust refers to the conviction in the “ability and disposition of an entity to act appropriately, within a specified context”, whereas trust can also stand for “reliability and truth of information” in a more common conception (ITU, 2010).

Although there is no direct way of perceiving an entity (Sakimura, 2013), *attributes* A allow for binding information through making statements about

an entity. With A being the set of all attributes, an attribute $a \in A$ can specify a characteristic of an entity via a type/value statement (ITU, 2010), e.g., age is 42 or gender is female. An *identity* $i \in \mathcal{I}$ represents an entity within context $c \in C$ using a set of context-specific attributes $A_c \subset A$ (ITU, 2010), where \mathcal{I} is the set of all identities and C is the set of all contexts⁹. Determining the variety of attributes assignable to an entity, the *context* stands for an “environment with defined boundary conditions” (ITU, 2010). *Identification* is about recognizing an entity within context through a set of attributes, where a particular subset $A_i \subseteq A_c$ is also known as an *identifier* (ITU, 2010). *Anonymity* denotes the state of being unable to identify a particular entity among other entities (ITU, 2010).

One and the same entity e can have several identities $I \subset \mathcal{I}$, e.g., husband, friend or co-worker (Sakimura, 2013). Any such contextual identity results from a certain *self-perception* of a (humanoid) entity and is expressed through a set of attributes (Barker et al., 2012; Sakimura, 2013). Given that it is virtually impossible to *fully* represent each entity by one holistic identity (ITU, 2010), it is also not feasible to completely eliminate issues in recognition. The *third-party recognition* of an identity may differ from the self-perception it originated from (Pronin et al., 2004), e.g., because not all attributes are recognizable, unexpressed attributes are inferred or the context is regarded diversely.

There are two types of identity assertions: *self-asserted identity*, which an entity declares to be its own (ITU, 2010) and *third-party asserted identity*, which a trusted authority assigns an entity after successfully confirming all identity claims made by the entity. For this purpose, there are particular *service providers (SPs)* that either manage identities for other entities or put entities in the position to do so on their own. Such special SPs are

⁹As a so-called *holistic identity*, i.e., a complete representation of an entity by all characterizing attributes, has no basis in practice, we use $A_c \subset A$ rather than $A_c \subseteq A$.

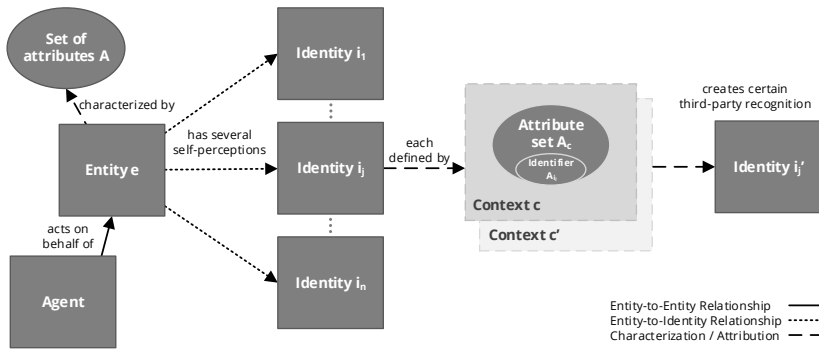


Figure 2.2: Essential Terms and Relationships

called *identity (service) providers (IdPs)*, whereas a SP is just any entity that offers a service via web-based means. A *relying party (RP)* is a SP that puts confidence in the identity claimed by a *requesting entity* within some context (ITU, 2010). Here, identity management (IdM) subsumes processes and facilities dedicated to identity-related tasks including creation, assignment, maintenance, termination, verification and assurance of entity/identity bindings (Dinger and Hartenstein, 2008).

A *principal* is “an entity whose identity can be authenticated” (ITU, 2010). Through comparing identity claims with information already verified during initial enrollment, the *authentication* aims at achieving “sufficient confidence in the binding between the entity and the presented identity” (ITU, 2010; Sakimura, 2013). Furnishing evidence for a claimed identity thus allows for successfully authenticating a requesting entity (ITU, 2010). The term *credential* refers to such evidence. There are three types of credentials: knowledge-based credentials like passwords, ownership-based credentials like certificates and inheritance-based ones like fingerprints (Bertino and Martino, 2007).

Protection refers to the state of keeping something safe from adverse effects. With reference to information and information systems, *security* or more precisely *information security* denotes the protection against adverse effects in terms of “unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability” (CNSS, 2010). The process of granting a principal the privilege to pass through protective measures taken against entities having insufficient permissions is called *authorization* (Barker et al., 2012). That is, a *privilege* is “a right that [...] permits [an] entity to perform an action” (ITU, 2010). In this regard, *access control* determines which principals have what kind of access to which resources based on a set of rules (Anderson, 2008; ITU, 2010). Distantly related to access control and put on a more abstract level, *privacy* is a privilege of individuals to control what personal data may be accessed by third parties for various purposes including collection, use and distribution (ITU, 2010).

In specifying the terms privacy and personal data, we fully comply with the fair information practices OECD established in their privacy framework by formulating eight basic principles of regulating management of personal data, including collection and use limitations, data quality and purpose, security safeguards, openness as well as individual participation and accountability (McCallister et al., 2010; OECD, 2013). As a consequence, the basic characteristics of the entities described in these guidelines match those of the stakeholders identified in the remainder of the chapter.

Having defined both the research methodology and the terminology used in this work, the following section continues with investigating the central problem.

2.2 Problem Analysis

Identifying causes as well as effects that surround the central problem this dissertation is dealing with, Subsection 2.2.1 describes today's handling of personal data using a set of scenarios, Subsection 2.2.2 characterizes major stakeholders involved in these scenarios, and, finally, Subsection 2.2.3 extracts and classifies prevalent problems impacting these stakeholders.

2.2.1 Scenarios

By employing following scenarios, we aim for illustrating the management of personal data that we considered as characteristic in today's landscape of web systems and, furthermore, for hinting at negative aspects, e.g., shortcomings, risks and limitations, which possibly arise from the situation described.

Scenario 2.1: Online Shopping. As her old camera got irreparably damaged by water during the last vacation, Alice wants to buy a new product for her upcoming vacation. She therefore invested a significant amount of time to inform herself and compare suitable products with each other. After hours of searching, Alice finally found a candidate matching her expectations. Taking advantage of an online price comparison portal, she hopes to discover an online shop that offers the desired product for a reasonable price. Alice actually makes an attractive find, yet it is a recently launched online shop she never used before. As part of the deal of buying the product, she has to provide some of her personal data, e.g., phone number, address and banking account data, to this SP for reasons not completely clear to her at the moment. Here, Alice can choose between registering by entering her data manually or relying on one of today's popular social logon options offered by a major social platform she signed up some time ago.

While she knows about the benefits of not having to enter her data over and over again and just rely on her usual password, she is also aware of the fact that she often had to complement thus provided data by additional details about her in order to fully utilize the service. Alice also considers that such shares of personal data, e.g., her favorite delivery address, are specific to particular services and cannot be transferred back to the provider of her preferred social logon option because it only supports a self-contained set of attributes to describe her identity. Moreover, Alice is troubled about what she read on tracking, analysis and exploitation of site visits and interactions among online users by some IdPs. So, she decides to go without social logon and sign-up manually yet in knowledge of the likely limited possibilities of exchanging, reusing and managing personal data she provides the particular service with.

Scenario 2.2: Password Trouble. In addition to completing the form presented by the SP with her personal data, Alice also needs to create a password as part of the registration process of the online shop. Here, she remembers the broad media coverage on security leaks and the large number of requests she received from diverse SPs in the last couple of months asking for updating her passwords. Alice recollects the time-consuming, repetitive effort to change the passwords of user accounts she maintained at potentially affected SPs. She had to log in with her old password, create a new one, confirm it by e-mail, and verify that the password change was successful. In doing so, Alice was confronted with miscellaneous user interfaces for managing certain functions of the underlying access control facilities. At some point in time, she lost track of whether she has really updated all passwords or missed some of them. Alice knows that even old passwords in the wrong hands might do harm because back then she was too sluggish to create a unique password per SP.

Recalling her last visit in person at the bank to put something in her safety deposit locker, she noticed the obvious security measures—like cameras,

guards and the bank vault—taken to protect her financial capital and other valuable objects. Imagining the hypothetical event of a bank robbery or a bankrupt, Alice feels confident that she could count on her insurance or the deposit protection funds established by the government.

Sensing that both attacks and successful breaches on web systems are increasing in frequency and severity, Alice wonders what measures are put in place by SPs, like the online shop or the IdPs offering the social logon option, and by governments in order to protect her personal data. While she received announcements that some services offer an improved protection, Alice doubts these promises as she can hardly find evidences, solid assurances or evaluation results.

Scenario 2.3: Holiday Replacement. For her upcoming vacation Alice needs to name a holiday replacement at work, which represents her business interests in her absence. She authorizes her co-worker Casey to act on her behalf during the vacation. They agreed on the scope of Casey's new function and tasks in an extensive conversation. Being her holiday replacement, Casey has to access some web applications Alice is regularly using in her work routine. Even though she does not want Casey to know the passwords she is using to authenticate herself to those web applications, Alice does not see another option as she is the only one who can make certain transactions. The thus acquired access to her user accounts let Alice feel uncomfortable because they also contain personal data, like her private phone number, which would be disclosed to Casey. While Alice could remove or alter such attributes describing her identity and re-entering them after her vacation, she shies away from this effort. When returning to work, she will change her passwords and verify that her personal data was not modified by mistake or on purpose, since she knows Casey as a guy who occasionally makes some bad jokes at other people's expense.

From the scenarios, we extract stakeholders in the following subsection.

2.2.2 Stakeholders

Being the first artifact of the analysis phase in LFA, we summarize below the stakeholder matrix shown in Table A.1 of Appendix A. As stakeholders are entities that have a stake in the outcome of a project (Miklič, 2008), their analysis is of particular importance for the success of this work. Independently yet in line with related work¹⁰, e.g., (WEF, 2011) and (Bria, 2012), we identified three groups of stakeholders, namely individual persons (citizens, people), companies (corporation, firms, private sector), and governments (agencies, public sector, regulators). While the main stakeholder groups have already been broached as part of the motivation in Section 1.3, this subsection details their characterization, their concerns in the context of the central problem and their motivation for improving the current situation.

Individual Persons

Like protagonist Alice, we became acquainted with during above scenarios, individual persons belong to those human entities that employ web systems for reaching their personal short-term objectives, like assistance for shopping or work activities. They therefore consume and produce various types of content in a customized way. A subset of these so-called *web users* are users of online social networks, who do not only publish and manage their user-generated contents with functions provided by OSN services, but who also want to stay informed by keeping themselves up-to-date about activities of other users.

Individual persons critically regard the sphere of influence of large web companies and are concerned about protection of their privacy especially due to frequent news coverage on data disclosures (WEF, 2011). There are

¹⁰Stakeholders mentioned by related work are enclosed in parentheses.

several issues with regard to personal data management that bother individual persons. Not only do these issues include the unspecific appropriation, unclear ownership, and less transparent exploitation of personal data by third parties (Halpin, 2014; Jahid et al., 2012), but also the constrained options of choosing personal data operators and the restricted means of extending, managing and maintaining quality (like validity and consistency) of personal data beyond application-specific limits (Dinger and Hartenstein, 2008). It is difficult for individual persons to maintain control over access and use of their personal data, to exchange and migrate data across boundaries of SPs, and to estimate risks and aftermaths of data disclosures with respect to moment, location, type, extent and involved people (EC, 2011; Müller, 2008). Individual persons are aware that disclosure of personal data to unwanted parties means that it is possibly disclosed forever without having any proper mean of compensation in place. So, they fear that personal data can fall into wrong hands and is used for purposes they do not agree to, like impersonating the real identity owners or monetization by criminals on black markets (Holt and Smirnova, 2014; Horacek, 2013). The very risks associated to such incidents force individual persons to questionable actions, yet without exact knowledge about security measures companies integrated into web systems to keep personal data safe. This results in syndromes of individual persons such as *password fatigue* and following frequent calls for global password resets evoked by incident reports (Florêncio and Herley, 2007; WEF, 2011), which is rather a degeneration into fighting symptoms than an attempt to systematically address the actual problem.

These issues conflict with the endeavors of individual persons to have an open, free and customizable user experience when surfing the web, where they are enabled to maintain ownership and control about use and safekeeping of their personal data.

Companies

By providing particular services to individual persons and other organizations through web systems, companies seek for some kind of return on their investment, e.g., revenues from advertisements (Halpin, 2014). Here, users take a central role in the business strategy of many companies (cf. *online shopping* and *holiday replacement* scenarios). Aiming for increasing the return, companies want to maintain the current number of active users and attract new ones. They therefore invest by continuously adding more and more functions to web systems. Moreover, some companies also benefit from acquiring information about certain user characteristics, particular with regard to personal data, and making thus acquired information accessible to third-party businesses in return for payment (Bria, 2012; Jahid et al., 2012). That is, such businesses cooperate with companies, like OSN service providers, in order to gain access to particular aspects of personal data which is valuable to them, e.g., for creating appealing context-sensitive advertisements that are customized to individual persons (Halpin, 2014).

The central problem affects companies in several regards. Firstly, with web systems getting more and more complex, they become harder to maintain and evolve for developers. Among other things, this bears the risk of taking only insufficiently account of security during development and, thus, introducing security weak spots by mistake, which increases the vulnerability of web systems for attacks that potentially target the personal data of their users (cf. *password trouble* scenario). Secondly, conducting forensic analysis after security breaches and fixing discovered security issues in a timely and permanent manner is consequently business critical for all involved parties, yet the costs incurred put additional weight upon the balance of affected companies (Gangewere, 2013; Riddell, 2011). Thirdly, companies are concerned about risks associated to a decline in the number of users, with effects on core businesses as well as partners,

which are responsible for further exploitation or for related tasks, like storage or marketing. They know that today's users are put in the position to vociferate their grievances by swiftly and globally spreading news on incidents, breaches and data disclosures, which are recognizable to them. Victims do not have to await official statements by involved companies, but they can utilize social media for word of mouth information exchange. In this regard companies also fear that offering functions to ease migration and exchange with competing SPs could result in a drain of users and, therefore, a reduction of social capital and possible revenues.

These issues conflict with the endeavors of companies to get an advantage over competitors by optimizing diverse parameters such as growing the number of active users, reducing costs, accelerating the development process with respect to maintenance and evolution of web systems, or obtaining and retaining access to high quality, up-to-date personal data of users with minimal effort. Lowering the opportunities for security breaches by design would relieve companies from higher investments in damage control and enable users to regain trust in the management of personal data by web systems.

Governments

Democratically elected governments are political organizations which exercise control and make decisions for those individual persons, called citizens, who usually occupy and legally belong to specific territories, called states, state federations, or political unions of states, as in case of the EU (Merriam-Webster, 2015). Governments are appointed to represent, protect and fulfill interests of their citizens.

In doing so, governments are concerned about the fact that data disclosures through attacks on web systems increasingly clash with mandated

responsibilities to protect personal data of citizens (cf. *password trouble* scenario) (EC, 2012). While citizens often do not see any alternative to sharing their personal data with third parties (cf. *online shopping* scenario) before they are enabled to employ certain services provided by companies (EC, 2011; Halpin, 2014), governments want to assure civil rights and make companies accountable for managing personal data of citizens. Governments are furthermore affected by the central problem as the prosecution of cybercriminals, which are suspected of attacks on web systems, is a significant administrative burden as well as an expensive, difficult and time-consuming matter, which is also owing to fragmented legal environments with divergent and inconsistent protection rules across countries (EC, 2012; WEF, 2011).

In line with efforts to make access, transfer and control over personal data general accepted fundamental citizens' rights, including the "right to be forgotten", governments attempt to force the principles of "privacy by design" and "privacy by default". They hereby intend to urge companies for integrating mechanisms to safeguard personal data into their products beginning from the earliest stage of development (EC, 2012). Not only would implementing such initiatives, in the context of the secure by design vision, assist in ensuring individual persons that web systems, web applications and web services provided by companies act responsibly towards managing and protecting personal data, but also contribute to reduce costs and administrative burden for organizations, including both companies and governments.

After the main stakeholders have been described, the following subsection continues with discussing prevalent problems that impact them.

2.2.3 Prevalent Problems Impacting Stakeholders

In line with the logical framework approach, we extracted several negative aspects from the scenarios and the affected stakeholders. Even though it does not lay claim for being complete, we consider the selection of negative aspects characteristic for describing the problem domain. Relating these problems with each other resulted in a so-called problem tree. It represents the current negative situation exemplified by Scenarios 2.1 to 2.3 as a set of cause-effect relationships (EC, 2004). While Figure A.1 in Appendix A illustrates this second LFA artifact completely, we report on the findings of the associated analysis individually in the remainder of this subsection.

There are various problem causes that pile up to the central problem, which we differentiate in first, second etc. level causes. Level one causes primarily induce the central problem, whereas secondary problem causes provoke them and so on. Reusing this structure, we continue with listing the problem effects we identified as negative consequences of the central problem. In order to provide a bigger picture of the impact of the central problem, we once again divided the effects into primary, secondary etc. ones. Each problem cause and each problem effect represents a problem on its own. Figure 2.3 schematically depicts this structure as an overview, which we also rely on in the following discussion.

Causes of the Central Problem

To detail causes and root causes of the central problem, we traverse them beginning at the first level and proceeding in depth-first order:

Problem Cause 1: Security of Web Systems Treated as Afterthought.

From the *password trouble* scenario, it is apparent that a high number of incidents entail serious consequences through data disclosures, which

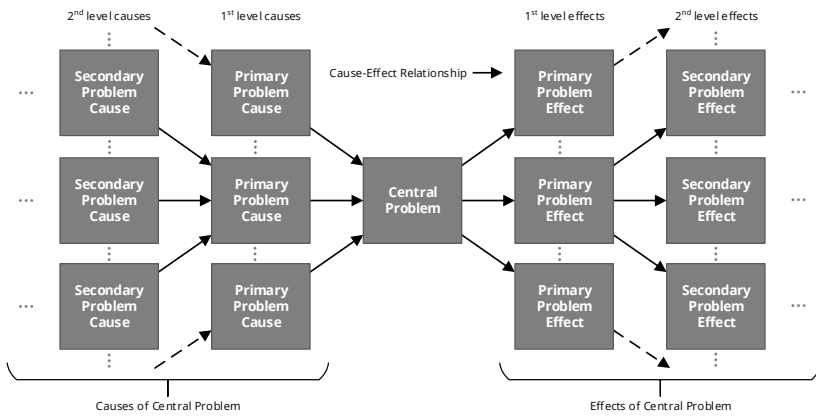


Figure 2.3: Structure of Causes and Effects of the Central Problem

especially impedes individual persons as stakeholders. This is furthermore an indication that companies do not treat security of personal data managed by web systems, including involved web applications and web services, with the same care as other parts of the business (Horacek, 2013; Symantec, 2014; Verizon, 2014), like acquiring new users through swiftly releasing new features. Figure 2.4 illustrates this primary problem cause and its root cause as a part of the problem tree, where other direct causes and the central problem are included to provide context. Here, the lack of means for modeling secure web systems induces Problem Cause 1, as described next.

Problem Cause 1.1: Lack of Means for Modeling Secure Web Systems.

Similar to the incomplete means offered to users for controlling and protecting their personal data (cf. *online shopping* and *password trouble* scenarios), there is also a deficit in proper modeling approaches (Papazoglou et al., 2007). Without adequate support for modeling security aspects, developers will continue to pay only minor attention to shield sensible parts of web systems, e.g., web applications and web services, from criminal activities.

Such modeling approaches would allow software developers employed by companies for designing, evolving and maintaining web systems that take security as a key element into account (Wild and Gaedke, 2014).

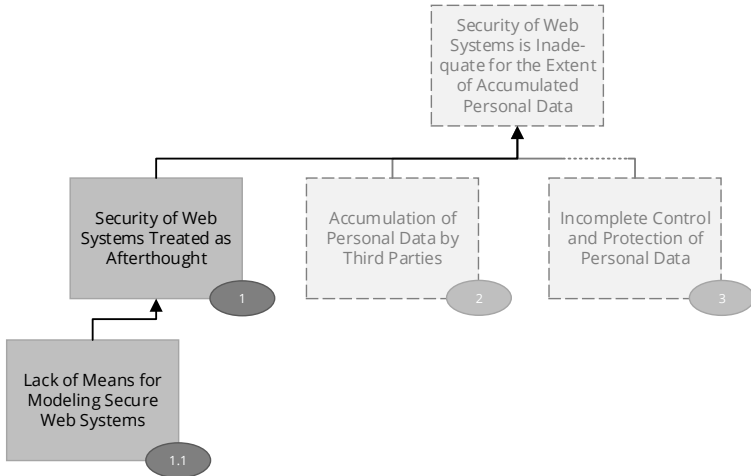


Figure 2.4: Problem Cause 1 and Associated Root Cause

Problem Cause 2: Accumulation of Personal Data by Third Parties. From the *online shopping* and *password trouble* scenarios, it follows that in addition to an insufficient consideration of security in the design and during runtime of web systems also the aggregation of personal user data by third parties is another direct cause of the central problem. This is due to the fact that such aggregation of personal data creates an unnecessarily lucrative target for attacks, which promises convincing risk-reward-ratio for cybercriminals (Horacek, 2013). Figure 2.5 illustrates this primary problem cause and its root causes as a part of the problem tree, where other direct causes and the central problem are included as context information. Here, the customer and data lock-in induces Problem Cause 2, as detailed in the following.

Problem Cause 2.1: Customer and Data Lock-in. In order to foster long-term customer retention, companies make it difficult for individual persons to switch to web systems of competitors (Halpin, 2014). Companies do this by refusing to equip products with necessary migration options. Furthermore, companies invisibly increase the barriers for migration by making users dependent on specific characteristics of provided services (Bria, 2012), e.g., offering many OSN members to communicate with or expressive user profiles for self-presentation. Several factors withhold users from migrating manually on their own, including the potential effort and the loss of entered data (Yeung et al., 2009). With the thus gained access to an everlasting source of up-to-date information from and about users, companies aim

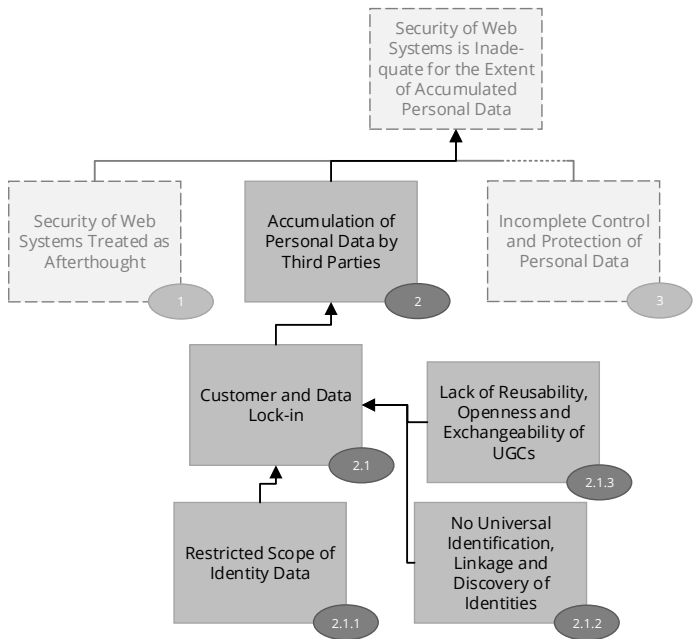


Figure 2.5: Problem Cause 2 and Associated Root Causes

at exploitation of their users' social capital for self-serving purposes, like revenues through advertisements (Bielenberg et al., 2012; Jahid et al., 2012). The unclear ownership of personal data acquired by third parties makes this matter even more severe (WEF, 2011), as highlighted in the *online shopping* scenario. The restricted scope of identity data, no universal identification, linkage and discovery of identities, and the lack of reusability, openness and exchangeability of UGCs furthermore provoke this secondary problem cause, as outlined next.

Problem Cause 2.1.1: Restricted Scope of Identity Data. Despite the advantages of today's heterogeneous landscape of web systems, this variety also yielded diverse ways of managing user data (Jordan et al., 2003). Intending to enable user identification and fast data processing by allowing users for attaching data, e.g., contact details, to their identities, companies introduced a multitude of separate identity management systems (IdMSs) over the last decades. Without a wide-ranging, fully satisfying and mandatory SSO solution in place, companies designed such systems primarily to fulfill particular needs of the web applications and web services they provide. For example, it would be pointless from the perspective of an online camera shop to allow users for stating which cars they drive as part of their identity data, when users are solely employing the web application for buying camera equipment. While focusing on a certain field of application is both justified and beneficial from the company's point of view, users are left with multiple application-specific identities. Such application-specific identities are represented by sets of attributes that are restricted to particular domains. However, this prevents users from employing identities for web systems different to the originally intended ones (cf. *online shopping* scenario) (Bria, 2012). In addition to identities per se, corresponding management functions are also restricted to domain- and application-specific limits.

Problem Cause 2.1.2: No Universal Identification, Linkage and Discovery of Identities. With diverse systems for IdM in use, users have to rely on several identities for authentication, which narrows identifications, linkage and discovery to particular application fields (cf. *online shopping* scenario) (Halpin, 2014). For example, it is difficult to detect by means of a known identity whether an individual person, like Alice, is member of two competing OSNs, which rely on different IdMS. Consequently, also interlinking identities across fields of application is problematic. This also interferes with the idea of having a social web with universal identification and discovery.

Problem Cause 2.1.3: Lack of Reusability, Openness and Exchangeability of UGCs. Not only is the customer lock-in issue intensified by identity usage restrictions but also by missing options for reusing and exchanging UGC in an open way (Bria, 2012). While web systems, like a video sharing website, offer facilities for comfortably adding UGCs, like self-made video clips, they are often sparing with adequate functions for exporting once provided contents in order to prevent drain of social capital to competing product offerings. This confinement raises questions and expectations by users about service reliability and use of personal data not only in the possible event of a hostile takeover by a competitor.

Problem Cause 3: Incomplete Control and Protection of Personal Data. From the *online shopping* and *holiday replacement* scenarios emerges another cause of the central problem in terms of the lack of clarity about the protection of personal data. Accompanied by the non-transparency to individual persons how their personal data is safeguarded against disclosure, tampering and unintended use, when managed by third parties (Bria, 2012), there is also an incomplete set of means in their hands which hinders users to self-manage access, modification and delegated use of personal data by other. This is largely due to the fact that today's companies adopted a silo-centric approach of managing personal data with characteristics specific to

particular SPs (Halpin, 2014). Figure 2.6 illustrates this primary problem cause and its root causes as a part of the problem tree, where Problem Causes 1 and 2 and the central problem are included to provide context. Here, the insufficient control of identity based on individual context, the risk of identity theft and tampering of personal data, and the incomplete range and granularity of access control induce Problem Cause 3, as explained in the following.

Problem Cause 3.1: Insufficient Control of Identity Based on Individual Context. Today’s prevalent application-specific character of identities restricts a broader use and control by individual persons outside the field of application specified by SPs (cf. 2.1.1) (Halpin, 2014). While an identity represents an entity within context, those contexts are usually not defined by the real identity owners but by service providers. Not only does this prevent identity owners from extensively controlling read/write access to their identities including associated attributes sets as part of their personal data

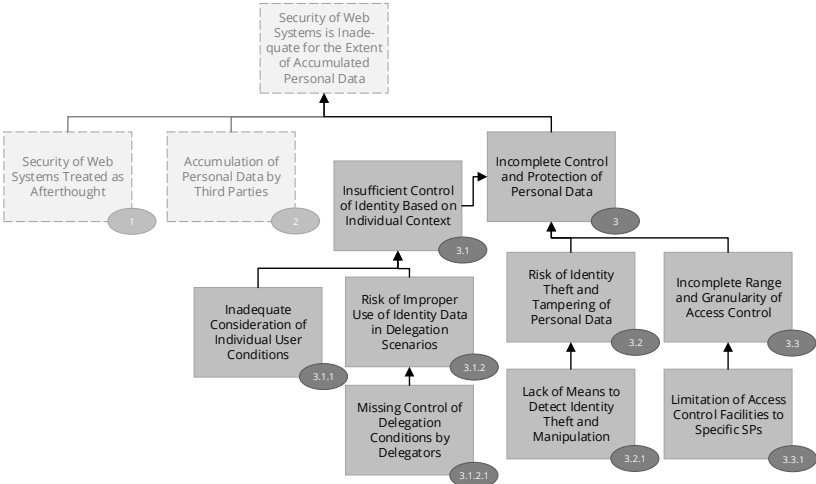


Figure 2.6: Problem Cause 3 and Associated Root Causes

(cf. *holiday replacement* scenario), but it also impedes further exploitation in other application fields on the identity owners' own expectations (Bria, 2012). The inadequate consideration of individual user conditions furthermore provokes this secondary problem cause, as described next.

Problem Cause 3.1.1: Inadequate Consideration of Individual User Conditions. Individual persons have different expectations towards privacy and protection of personal data (WEF, 2011), e.g., whether to share their date of birth, phone number or email address. Nevertheless, they employ similar web systems for different purposes. They therefore rely on diverse devices and web browsers, and use them in various environments, like at work or at home. When issuing, managing and using identities, such individual conditions and preferences are, however, only insufficiently considered and, thus, intensify Problem Cause 3.1. Creating multiple identities allows for partially resolving this issue, yet at the expense of burden identity owners with having to maintain redundant identity attributes and associated credentials (WEF, 2011).

Problem Cause 3.1.2: Risk of Improper Use of Identity Data in Delegation Scenarios. In some scenarios, like holiday replacement, it is necessary that an entity acts on another's behalf (Wild et al., 2015). To do so, the so-called delegator provides the delegate with certain privileges. These might include access to personal data, especially identity data, of the delegator, e.g., allowing the delegate for replying to messages on the delegator's behalf while employing her profile-bound business contact list. Although accessing and utilizing the delegator's identity data by a delegate might be less problematic in trustworthy environments and relationships, e.g., a familiar context, it bears risks towards an exploitation of the delegator's personal data beyond the original intention (cf. Problem Cause 3.1). That is, delegates can misuse the delegator's identity and data for inappropriate purposes, e.g., ordering goods in the name of the delegator or impersonating the delegator in business or private arrangements. The missing control

of delegation conditions by delegators furthermore induces this tertiary problem cause, as outlined below.

Problem Cause 3.1.2.1: Missing Control of Delegation Conditions by Delegators. In contrast to Problem Cause 3.2, which deals with tampering and identity theft certainly without prior authorization by identity owners, Problem Cause 3.1.2 is effected by a too informal authorization through the delegator (cf. *holiday replacement* scenario). That is, delegators are not put in the position to clearly specify the scope of a delegation in terms of limitations, with SPs having to obey these limitations to avoid conflicting with the delegator's intention.

Problem Cause 3.2: Risk of Identity Theft and Tampering of Personal Data. It is apparent that individual persons count on today's web systems for properly managing and protecting the different types of personal data they entrusted them with. Even though users hand over more and more personal data to third parties, threat reports indicate that companies cannot keep pace with this development without jeopardizing high quality standards, particular with regard to security (WEF, 2011). Such transfer of responsibility and credit of trust by individual persons to companies bear risks towards management of personal data by web systems different to what users have expected. Consequences include unwanted data disclosure (cf. Problem Cause 3.3) as well as tampering of personal data and even identity theft (WEF, 2011). Here, risks are not limited to external aggressors, but also involve insiders that work for companies. The lack of means to detect identity theft and manipulation furthermore effects this secondary problem cause, as explained in the following.

Problem Cause 3.2.1: Lack of Means to Detect Identity Theft and Manipulation. Aside from missing means to control the conditions of delegations (cf. Problem Cause 3.1.2.1 and the *holiday replacement* scenario), there is another shortage of measures affecting the protection of personal data also in non-delegation scenarios (WEF, 2011). Without extensive

measures enabling identity owners to protect their data against tampering and even identity theft, this increases the risk that attacks remain undiscovered by both identity owners and SPs regardless of whether malicious actions happened temporarily or permanently. Temporal attacks carry a special risk as they allow aggressors for tampering identity data, employing manipulated data for their purposes and disguising the tampering from later exposure by rolling changes back.

Problem Cause 3.3: Incomplete Range and Granularity of Access Control. There is a high diversity of access control facilities, which companies employ in their products. Different web systems provide users with different measures, which again vary in scope and granularity of controlled access (cf. *password trouble* scenario). As a consequence, both the inconsistency and the incompleteness of access control measures hampers users from properly protecting their personal data (Bria, 2012). Related to Problem Cause 3.2, a lack of holistically controlling access to personal data, including PII, on a fine-grained basis negatively contributes to the incomplete control and protection of personal data (cf. Problem Cause 3). Moreover, the limitation of access control facilities to specific SPs induces this secondary problem cause, as described below.

Problem Cause 3.3.1: Limitation of Access Control Facilities to Specific SPs. Completing the picture of a deficit of means to fully safeguard the identity owner's interests against various attacks (cf. Problem Cause 3.1.2.1 and Problem Cause 3.2.1), the range and granularity of today's prevailing access control strongly depends on specific SPs. Such diversity in access control facilities hampers identity owners in holistically protecting the variety of their data against acquisition by unwanted third parties (cf. Problem Cause 3.3). As companies integrate different types of access control in their web systems, which again offer different levels of granularity in protection, the restrictions defined by users are consequently only effective for users employing the particular application the access is controlled for.

Staff maintaining the applications, like administrators, are usually left out in this consideration¹¹. Users, who want to migrate their personal data to another SPs, are further impaired as specified access control settings are often unavailable for migration (cf. Problem Cause 2.1).

Having classified the causes of the central problem at different levels, we proceed with its primary and secondary effects.

Effects of the Central Problem

To detail the effects and aftereffects of the central problem, we traverse them starting with the first level and continuing in depth-first order.

Problem Effect 1: Rise in Successful Attacks on Web Systems. Various threat reports show that the number of attacks on web systems is increasing with negative impacts on both companies and individual persons through breaches and disclosures of personal data (Horacek, 2013; Symantec, 2014; Verizon, 2014). Figure 2.7 illustrates this direct effect of the central problem and its aftereffects as a part of the problem tree, where other primary problem effects and the central problem are included to provide context. Caused by too weak security for too much personal data entrusted by too many users to too few SPs, Problem Effect 1 secondarily provokes the risk of declining reputation, revenues and trust in web systems, the monetization of personal data by criminals, and the uncertainty about circulation and exploitation of personal data, as described in the following.

Problem Effect 1.1: Risk of Declining Reputation, Revenues and Trust in Web Systems. An increase of successful attacks on web systems (cf. Problem Effect 1) entails serious risks for affected companies. They have to fear damages in brand reputation, revenues and trust of users in provided services (Dane, 2012). Companies would have to cope with a fading user

¹¹Unless the identity owner having the only key applies encryption.

base as well as declining revenues, falling stock prices and lower valuations (Gangewere, 2013). This bears the potential of endangering business models and, consequently, the very existence of involved companies.

Problem Effect 1.2: Monetization of Personal Data by Criminals. Apart from negative impacts on companies, individual persons are also hit hard through attacks on web systems that resulted in data disclosure (Horacek, 2013). Aggressors aim for monetizing stolen personal data of users on specialized black markets (Halpin, 2014; Holt and Smirnova, 2014). Through the uncertainty about circulation and exploitation of personal data as an associated consequence (cf. Problem Effect 1.3), this problem has an adverse impact on affected individual persons.

Problem Effect 1.3: Uncertainty about Circulation and Exploitation of Personal Data. Data disclosures through severe attacks on web systems worry individual persons in terms of the risks resulting from stolen and resold personal data (Holt and Smirnova, 2014; Jahid et al., 2012). Those concerns are especially fed by the unpredictability when which parts of

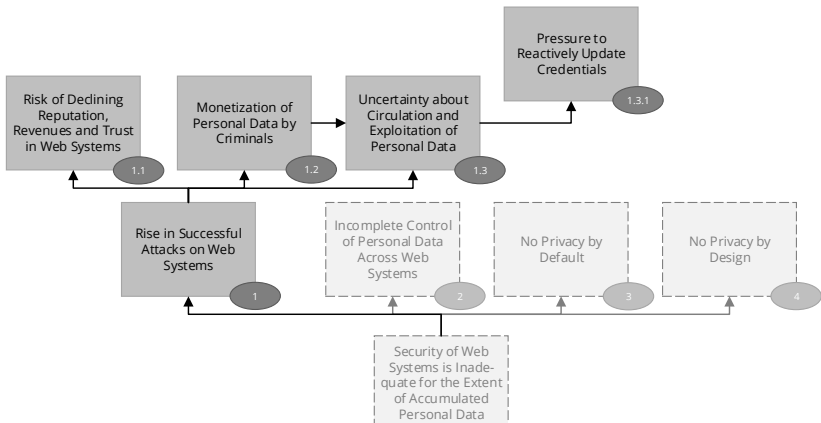


Figure 2.7: Problem Effect 1 and Associated Aftereffects

stolen or illicitly retrieved personal data will be circulated and exploited by whom to what extent and in which context (WEF, 2011). Moreover, the reluctance of companies to release information on incidents directly to the public further complicates that matter through missing transparency (Dane, 2012). This problem additionally increases the pressure to reactively update credentials, as explained next.

Problem Effect 1.3.1: Pressure to Reactively Update Credentials. In consequence of the uncertainty surrounding use and circulation of personal data (cf. Problem Effect 1.3), individual persons are asked by SPs for regularly controlling access and protection of their data, e.g., keep credentials up-to-date (WEF, 2011). While this can be considered as beneficial for maintaining security, users obey such recommendations without exact knowledge whether protective measures are really effective against aggressors. Fighting symptoms rather than tackling underlying challenges does not only shift the problem away from companies towards individual persons, but also creates problematic side effects, like password-fatigue (Florêncio and Herley, 2007). The fact that once stolen personal data is potentially disclosed forever aggravates this situation further.

Problem Effect 2: Incomplete Control of Personal Data Across Web Systems. As different web systems employ different measures for IdM and for protection to a varying degree of extent and quality, it is difficult for users, specifically identity owners, to holistically control their personal data without gaining profound knowledge in confidently handling such variety of measures. That is, web systems require that individual persons quickly familiarize themselves with specific measures in order to enable them to express what they want in terms of identity attribute editing, access control, and delegation of usage rights. Missing options to reuse personal data and associated protection preferences throughout applications represent an inherent part of this problem effect. As individual persons cannot keep up with redundantly maintaining their personal data at multiple SPs that

employ proprietary facilities for IdM and associated separate data silos, it becomes also difficult for companies to rely on accessing correct, consistent, coherent, and up-to-date personal data. Figure 2.8 show this problem effect as a part of the problem tree, where Problem Effect 1 and the central problem are included as context information. Here, the lack of privacy is an aftereffect of this problem, as outlined next.

Problem Effect 2.1: Lack of Privacy. Not taking sufficient account of treating privacy as a significant design principle, of making privacy a default for managing personal data, and of providing means for controlling access to personal data throughout web systems causes a lack of privacy with implications for all involved stakeholders (Bria, 2012).

Problem Effect 3: No Privacy by Default. Despite the presence of diverse privacy initiatives, like the proposal for reforming the data protection rules (EC, 2012), there is no all-encompassing default setting currently implemented by web systems, including involved web applications and web services, that would comprehensively ensure the privacy of individual persons. Such idealistic *privacy by default* setting for disclosing as few personal data to third parties as possible, however, interferes with the business models many companies have in use today. As a consequence, companies are reluctant in offering a default privacy setting to users (Jahid et al., 2012). Provoked by the central problem, this effect also contributes to the lack of privacy (cf. Problem Effect 2.1), as shown in Figure 2.8.

Problem Effect 4: No Privacy by Design. With security of web systems being inadequate for the extent of accumulated personal data (Bria, 2012), not only the condition of *privacy by default* is difficult to accomplish, but also the idealistic *privacy by design*. The latter is another initiative aiming for more privacy (EC, 2012). However, unless protective measures are considered sufficiently, how can privacy be considered by design? That is, only with adequate protective measures in place, it is manageable to consider privacy as an integral part of the design of web systems. Being a

negative impact of the central problem, this issue further adds to the lack of privacy (cf. Problem Effect 2.1), as depicted in Figure 2.8.

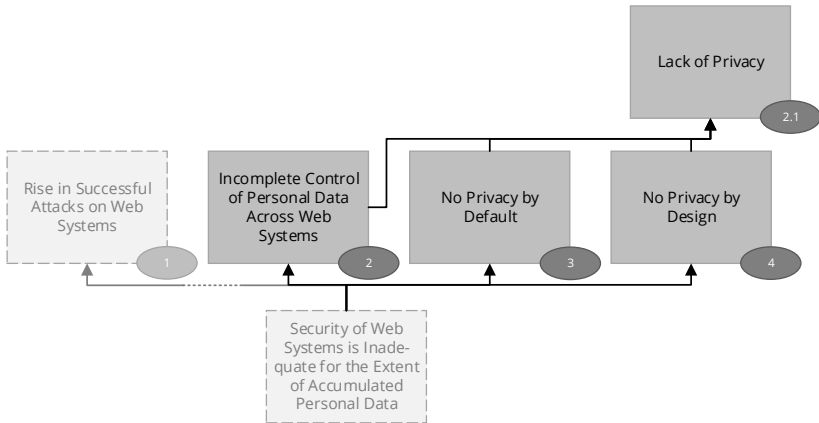


Figure 2.8: Problem Effects 2 to 4 and Associated Aftereffect

Now that the problems predominant in the context of the central problem are analyzed, we continue with transforming them into objectives in the next section.

2.3 Objectives

According to LFA, we transformed the negative statements that describe the problem domain into a positive reverse image by involving rewording, revision and consolidation (Miklič, 2008). Converting the problems classified in the previous subsection into desirable yet realistic and attainable achievements led up to an objective tree. Contrary to the problem tree, the objective tree represents the desired future situation as a set of means-ends relationships, where identified problems have been addressed success-

fully (EC, 2004). Figure A.2 in Appendix A illustrates this third LFA artifact completely, whereas Figure A.3 depicts the objective tree, in which certain closely related objectives have been consolidated for the sake of clarity.

In the consolidated version of this objective tree, we summarize all means necessary to achieve the purpose in three tangible first level objectives and five associated second level objectives. While the term *results* refers to these first level objectives, the term *activities* represents the second level objectives. Aiming for reversing effects and aftereffects of the central problem altogether, we also outline the consolidated *overall objective* we want to contribute to, yet in the knowledge that its achievement does not depend on the success of this project alone. Figure 2.9 shows the transformation of problems into objectives from a terminological point of view.

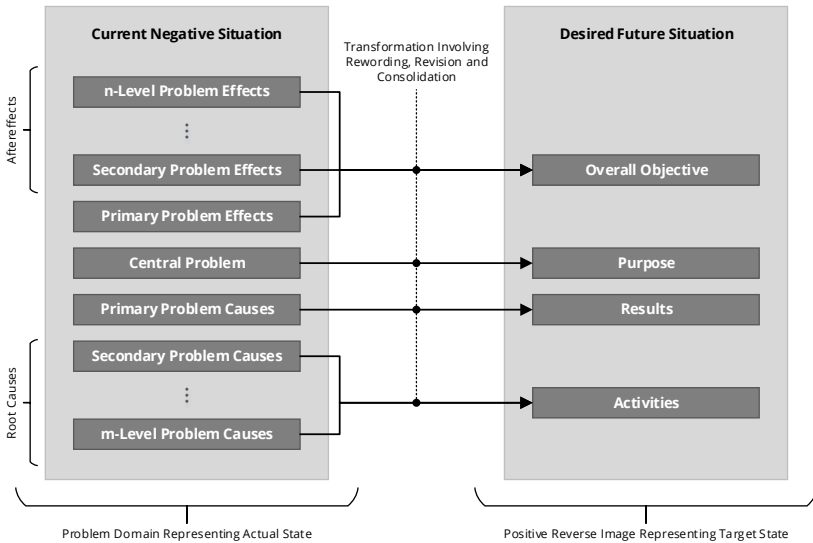


Figure 2.9: Transformation of Problems into Objectives

The remainder of this section reports on the findings of the objective analysis in greater detail, starting with results in Subsection 2.3.1, followed by activities in 2.3.2 and the overall objective in 2.3.3.

2.3.1 Results

To identify the results to be delivered for attaining the purpose, we describe the objectives we obtained by transforming the primary causes of the central problem in the following. The description of each result focuses on what is to be done and deliberately postpones details on the actual implementation to later chapters.

Result 1: Improved Modeling of Security Aspects for Web Systems. To address the problem that the security of web systems is treated as an afterthought (cf. Problem Cause 1), we have to enable companies for taking greater account of security in the engineering of web systems, including involved web applications and web services. This necessitates dealing with the associated secondary problem cause by undertaking following activity:

- *Activity 1.1:* Extend Means for Modeling Secure Web Systems

Result 2: Reduced Need for Accumulation of Personal Data by Third Parties. To address the problem of a hardly regulated accumulation of personal data by third parties (cf. Problem Cause 2), we must reduce the necessitation for companies to aggregate larger amounts of individual persons' data as part of their business model. This requires approaching the corresponding secondary problem cause by running the following activity:

- *Activity 2.1:* Offer Alternative to Customer Lock-in

Result 3: Extended Means for Control and Protection of Personal Data. To address the problem of an incomplete control and protection of personal data (cf. Problem Cause 3), we need to provide individual persons with appropriate safeguard measures for increasing control and protection of

their personal data. This involves addressing all relevant secondary problem causes by implementing the following activities:

- *Activity 3.1:* Improve Control of Identity Based on Individual Context
- *Activity 3.2:* Mitigate Risk of Identity Theft and Tampering of Personal Data
- *Activity 3.3:* Increase Range and Granularity of Access Control

Having explained the three results necessary to achieve the purpose, the next subsection proceeds with listing the activities to be carried out to produce them.

2.3.2 Activities

To tell which activities are needed to deliver the identified results, we convert the secondary causes of the central problem into the following five consolidated activities. Similar to the description of the results, we focus on outlining the content of each objective but leave information on the actual execution of the activities for later chapters.

Activity 1.1: Extend Means for Modeling Secure Web Systems. To deliver Result 1 by removing the lack of means for modeling secure web systems (cf. Problem Cause 1.1), we have to increase the support for developers to engineer web systems, including involved web applications and web services, that take particular account of security-related aspects.

Activity 2.1: Offer Alternative to Customer Lock-in. To create Result 2 by addressing the customer and data lock-in issue (cf. Problem Cause 2.1), we must provide companies with a suitable alternative to this widely applied procedure. Such alternative must not interfere with current business models of fostering customer retention. In doing so, this activity implicitly enables individual persons to regain ownership of their personal data. Car-

rying out this activity requires creating a positive reverse image to Figure 2.5 by a) increasing descriptiveness of identity attributes beyond the scope of individual web systems, b) improving identification, linkage and discovery of identities so that individual persons can use them across specific application fields defined by SPs, and c) enabling individual persons to employ their user-generated content in other usage scenarios through increasing the reusability, openness and exchangeability of UGC. Figure 2.10 shows such positive reverse image as a subtree of Result 2 within the objective tree, where the proposed measures to tackle Problem Causes 2.1.1 to 2.1.3 are directly subordinated activities of Activity 2.1 and the purpose as well as Results 1 and 3 are included as context.

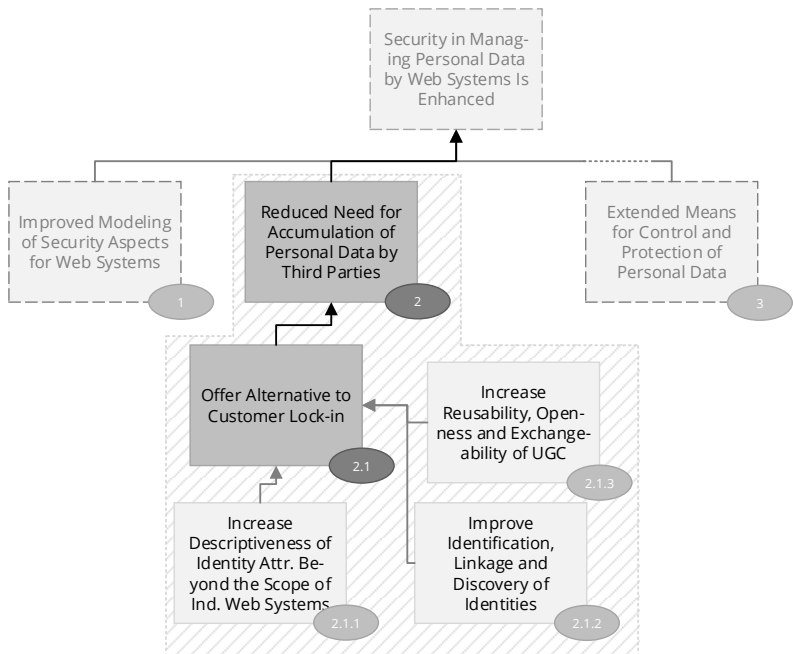


Figure 2.10: Activity 2.1 and Directly Subordinated Activities

Activity 3.1: Improve Control of Identity Based on Individual Context.

For contributing to produce Result 3 by approaching today's insufficient control of identity based on individual context (cf. Problem Cause 3.1), we are committed to increase the contextual scope of action for individual persons in terms of their identities. Undertaking this activity necessitates creating a positive reverse image to the left subtree in Figure 2.6 by a) enhancing consideration of individual conditions of users by web applications and, thus, enable an improved experience that is more custom-tailored to individual preferences of users, b) alleviating risks of improperly using personal data, especially with regard to PII, in delegation scenarios, and c) providing delegators with more control about the conditions of delegations in order to enable them to clearly specify the scope in which a delegate is allowed to act on the delegator's behalf. Figure 2.10 shows such positive reverse image as the left subtree of Result 3 within the objective tree, where the proposed measures to tackle Problem Causes 3.1.1, 3.1.2 and 3.1.2.1 are directly subordinated activities of Activity 3.1 and the purpose as well as Results 1 and 2 are included as context.

Activity 3.2: Mitigate Risk of Identity Theft and Tampering of Personal Data.

For further contributing to deliver Result 3 by mitigating the risk of identity theft and tampering of personal data (cf. Problem Cause 3.2), we are obliged to diminish both the chances and the impact of malicious manipulation of personal data, especially identity data. Executing this activity involves creating a positive reverse image to the centrally positioned subtree in Figure 2.6 by offering protective means to detect identity theft and tampering of personal data, with special regard to identity data, by identity owners, by SPs and by entities that request personal data. Figure 2.11 shows such positive reverse image as the centered subtree of Result 3 within the objective tree, where the proposed measure to tackle Problem Cause 3.2.1 is a directly subordinated activity of Activity 3.2.

Activity 3.3: Increase Range and Granularity of Access Control. For finally contributing to create Result 3 by approaching today’s incomplete range and granularity of access control (cf. Problem Cause 3.3), we have to extend as well as refine the filtering measures applicable to personal data. Working on this activity requires creating a positive reverse image to the right subtree in Figure 2.6 by reducing the dependency of individual persons on the access control mechanisms offered by SPs in order to facilitate holistically applying protection of personal data by individual persons throughout different web systems, web applications and web services. Figure 2.11 shows such positive reverse image as right subtree of Result 3 within the objective tree, where the proposed measure to tackle Problem Cause 3.3.1 is a directly subordinated activity of Activity 3.3.

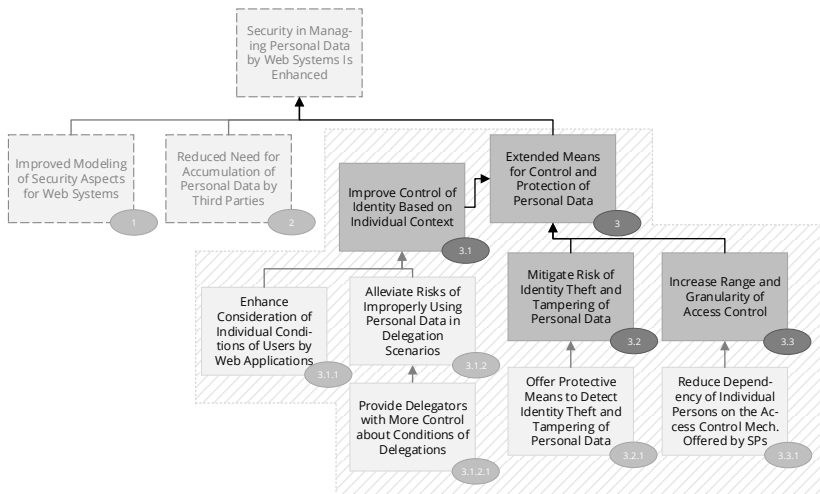


Figure 2.11: Activities 3.1 to 3.3 and Directly Subordinated Activities

After classifying the activities that represent the means to be provided for delivering the results and, thus, for achieving the purpose, the next subsection proceeds with its ends in terms of the overall objective.

2.3.3 Overall Objective

To estimate the positive long-term contributions of attaining the purpose (Miklič, 2008), we transformed all effects of the central problem (cf. Subsection 2.2.3) into objectives and, then, merged these objectives in one consolidated overall objective:

To Contribute to Increase the Protection of Privacy and Reduce the Risk of Personal Data Disclosures. With achieving the purpose, we intend to mitigate the problem effects by contributing to a reduction of attacks on web systems that would otherwise have entailed breaches and data disclosures with negative consequences for both individual persons and companies. Furthermore, we aim for improving the overall protection of privacy in the context of the web to the advantage of all stakeholders involved. This would also have a positive bearing on the mentioned problem effects by:

- Improving the overall control of personal data by individual persons throughout web systems, web applications and web services
- Supporting the accomplishment of the *privacy by default* initiative
- Helping to realize the *privacy by design* vision
- Minimizing risks for companies, with regard to losses in reputation, users and revenues, as well as for individual persons, with regard to a declining trust in web systems
- Reducing risks associated with monetization of personal data by aggressors
- Increasing certainty and awareness of individual persons about several usage aspects concerning their personal data, e.g., moment, origin and extent of access

- Decreasing the necessitation of updating credentials reactively once breaches affecting security of web systems, web applications or web services are reported to users

Now, that we have described all objectives that are either means of the purpose, in the case of results and activities, or ends of the purpose, in the case of the overall objective, the next subsection continues with explaining the strategy to achieve them.

2.3.4 Strategy

In this project, we embark on the strategy of processing the objective hierarchy in a bottom-up way. To achieve the purpose of an enhanced security in managing personal data by web systems for the benefit of individual persons, companies and governments, we therefore have to deliver the results by carrying out all identified activities. Due to the complexity of those activities, their success does not only depend on systematically approaching them within this work, but also on factors outside our direct control. Several assumptions have to hold true in order to allow for covering these factors and, thus, for successfully attaining the objectives with the necessary condition to verify their completion. That is, we contribute towards achieving an objective by fulfilling both inferior objectives and their associated assumptions (EC, 2004). Figure 2.12 schematically depicts this strategy, which operates on the documented product of the objective analysis, i.e., the so-called “Logframe matrix”. While Table A.2 in Appendix A represents the Logframe matrix completely, we summarize included assumptions, success indicators and verification sources in the following.

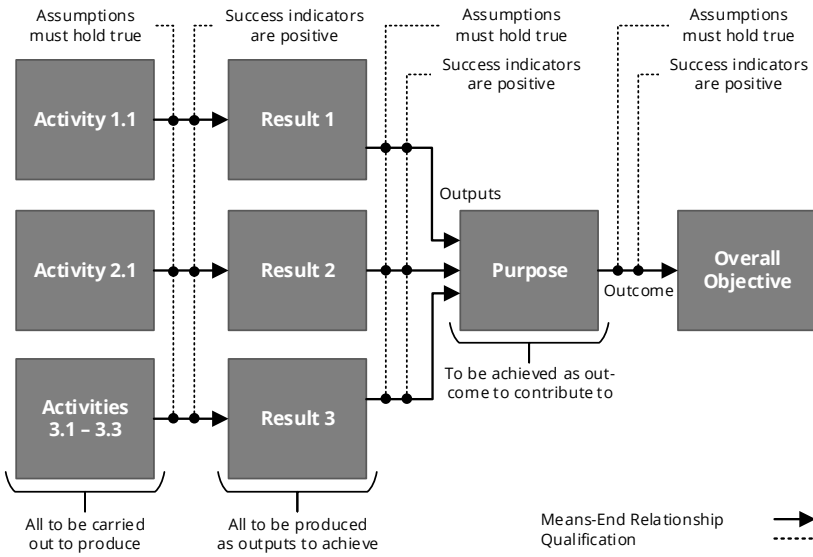


Figure 2.12: Strategy for Processing the Objectives

Assumptions

Clarifying under which circumstances it is possible to achieve each objective, assumptions take account of external factors that might impact the overall outcome of the project (NORAD, 1999). That is, assumptions help to deal with uncertainties of future developments outside the direct control of this project by determining the conditions which need to be present. In contrast to hypotheses, such as those outlined in Chapter 1, which are unproven falsifiable theories to be verified through investigation, assumptions are statements taken for granted, i.e., they are believed to be true but possibly cannot be proven (Merriam-Webster, 2016a; Merriam-Webster, 2016c). In the following, we list identified assumptions in relation to specific objectives and stakeholders at the level of purpose, results and activities.

Firstly, we assume that individual persons are willing to:

- apply security enhancements for personal data management (Purpose, Result 3),
- regain ownership and self-manage their personal data by maintaining a larger set of identity attributes yet with a lower amount of distributed copies (Activity 2.1),
- support detection and use of individual context and conditions (Activity 3.1),
- employ measures for extended access control, integrity protection, scope definition of delegations to safeguard their personal data (Activities 3.1 to 3.3), and
- make additional efforts for protecting their identity data (Activities 3.1 to 3.3).

Furthermore, we assume that companies are willing to:

- employ enhancements towards reducing necessitation of personal data accumulation, protecting personal data, and taking greater account of security aspects during design and runtime of web systems (Purpose, Results 2 and 3),
- use extended means for modeling web systems with special focus on security-related aspects (Result 1 and Activity 1.1),
- open up for universal identification, extended identity attributes, linkage and discovery across web systems, web applications and web services (Activity 2.1),
- settle with just obtaining access to user-generated contents rather than demanding storage within own premises and, hence, also giving up *own* application-specific facilities for controlling access to personal user data (Activities 2.1 and 3.3), and
- integrate means into their web systems for integrity protection and delegated access whilst obeying the scope defined by delegators (Activities 3.1 and 3.2).

Finally, we assume that governments are willing to:

- keep up with their support towards enabling more self-determined control about personal data for individual persons (Purpose, Activity 2.1).

It is evident that a large proportion of the assumptions are directed towards companies. Especially producing Result 2 requires a broad adoption and a rethinking of companies in terms of their business model, which is expected to be a long-term endeavor. While this result is essential for achieving the purpose of this work, it is not independently manageable as part of this dissertation. Yet, we intend to present an alternative to the current business model employed by companies, which would allow individual persons for controlling the conditions at which companies can access personal data.

In addition to knowing what needs to happen for achieving specific objectives, we must have some indicators that show when certain objectives are met based on what sources. This information is provided in the following.

Success Indicators and Verification Sources

Not only do success indicators and verification sources increase “clarity and specificity of objectives”, but they also assist in providing a “monitoring and evaluation framework” (EC, 2004). Indicators should therefore be specific, measurable, available, relevant and time-bound. For collecting necessary information to assess success, the project management team is eligible to perform acceptance tests, implement proof of concepts and make observations. Determining whether, when and to what extent we achieved objectives (NORAD, 1999), objectively verifiable indicators are installed at different levels in the objective hierarchy. They enable verifying that 1) all activities necessary for delivering results have been carried out, that 2) all outputs required for attaining the purpose are present through producing

the results, and that 3) the outcome of achieving the purpose contributes to the overall objective, which in turn creates certain impacts on stakeholders.

To assess the impact of contributing to the overall objective, we consider user studies as well as reports on analyzed risks and threats, like (Horacek, 2013), as appropriate verification sources. Although these sources would allow attesting both an increase in privacy protection and a reduction of breaches and data disclosures through successful attacks on web systems, the achievement of the overall objective is a joint long-run effort outside the direct control of this project and, therefore, not profoundly verifiable as part of this dissertation.

To assess the outcome of achieving the purpose, we think that integration and operational acceptance tests are suitable sources to verify not only the availability of all required results, but their prototypical integration into a demonstrator that proves the feasibility of the underlying concepts and illustrates the benefits of an enhanced security in managing personal data.

To assess the output of producing the results, we see proof of concepts as well as unit and operational acceptance tests as proper sources for verifying the successful implementation of all necessary activities in terms of extended protective means to safeguard personal data, modeling support with strong focus on security-related aspects, and an approach for accessing personal data that enables more control for individual persons by taking account of their preferences towards protection.

After discussing how to approach identified objectives, deal with factors outside our direct control, and verify progress as well as successful achievement of objectives, the following section proceeds with concluding this chapter.

2.4 Summary

Based on the foundations established by selecting PCM/LFA as an appropriate research methodology and by clarifying frequently occurring terms and definitions, we started investigating the central problem. From three scenarios illustrating the problem context, we extracted three relevant stakeholder groups and detailed how they are affected through the central problem. In order to describe the challenges we need to tackle, we further divided the central problem into first, second etc. level problem causes as well as primary, secondary etc. problem effects. Using this problem classification, we transformed all causes and effects into a consolidated means-ends hierarchy represented by several objectives. This hierarchy consists of five activities to be carried out to produce three results necessary for achieving the purpose, which in turn contributes to the overall objective. Having formulated a suitable bottom-up strategy to systematically meet the challenges, the next chapter examines to which extent already existing technologies bear the potential to facilitate achieving the objectives and fulfilling the assumptions.

State of the Art

3

To analyze state-of-the-art technologies with regard to their suitability of contributing to meet the challenges, this chapter first introduces a categorization of technologies that appertain to the scope of this work in Section 3.1. Based on the objectives obtained through investigating the research problem in the previous chapter, Section 3.2 specifies the analysis criteria by setting significance indicators, deriving requirements and specifying a rating system. Section 3.3 then discusses the analysis results to show the degree to which the requirements have been fulfilled by prior art. For creating an overview of existing technologies upon which we can build our research, as suggested in (Creswell, 2012), we employed systematic literature review and field testing of technologies. Finally, Section 3.4 summarizes the outcome of the state of the art analysis and the thus established basis for constituting our research contributions.

3.1 Categorization

The analysis of the problem in Section 2.2 showed that the reasons for today's insufficient consideration of security during development, runtime and evolution of web systems are manifold. However, Problem Cause 1 on “*security of web systems treated as afterthought*” and its secondary problem cause also pointed out that these reasons are at least partially attributed to an incomplete support of web engineers through appropriate development methods, models and tools. Leaving aside the variety of individual, pure implementation-driven approaches for manually developing web applications and web services, we aim at the engineering of web systems throughout their entire life cycle in an organized way. Here, web engineering is an eligible branch of software engineering that takes account of the particular conditions and needs of development efforts in the context of the web.

Web engineering is about the “application of systematic and quantifiable approaches (concepts, methods, techniques, tools) to cost-effective requirement analysis, design, implementation, testing, operation, and maintenance of high-quality web applications” (Kappel et al., 2006). That is, web engineering approaches (WEAs) intend to offer high adaptability and reusability, while reducing costs potentially caused through new developments and maintenance work (Gaedke, 2000; Koch, Meliá-Beigbeder, et al., 2008). By providing abstraction, formalization, separation of concerns and understanding through models, model-driven development (MDD) applied to the web serves this discipline of systematically building web systems (Saleem et al., 2014). Web engineering in conjunction with MDD furthermore enables to postpone the technical implementation on the basis of semi-automated model transformations until later engineering stages.

The discussion of analysis results on existing approaches for web engineering starts in Subsection 3.3.1, where we specially regard security aspects and use of models during the engineering process. Another important factor to be considered are users that employ web systems for their daily business. That is, users rely on web systems to properly take care of personal data entrusted to them. With web applications, web services and (web) users being essential entities as per definition on page 26, we continue the discussion in Subsection 3.3.2 with means for identifying and distinguishing those entities through appropriate web-based digital identities.

For further organizing this discussion, we extend the classification of IdMSs proposed in (Jøsang et al., 2007). Not only do we examine digital identities per se, but also other aspects of IdM including storage, integration and processes (Dinger and Hartenstein, 2008). While not necessarily predefined, many systems for IdM come along with a preferred form of authentication and a default set of attributes for describing entities, with implications as highlighted by Problem Cause 2 on “*accumulation of personal data by third parties*” and its subordinated problem causes. To cover identity creation and identification for both human and non-human entities, the options available for describing characteristics of such entities need to be sufficiently expressive and support various contexts. Subsection 3.3.3 investigates this matter by analyzing identity description languages (IdDLs).

In addition to modeling web systems as well as identifying and describing entities present therein, protection with focus on security in managing personal data takes an essential role. Applying protective measures, however, largely depends on the forgoing findings. By outsourcing the analysis of measures for protection and control to the related work sections of the solution components in Chapters 5 to 7, we enable detailed, self-contained and domain-specific reviews. Directing these technology reviews on either one of the areas of context-aware control, protection against tampering, and

access control assists in area-wise dealing with the root causes of Problem Cause 3 on “*incomplete control and protection of personal data*”.

Having outlined the categorization in web engineering approaches, identity management systems and identity description languages, the next section details criteria for examining the representatives of these categories.

3.2 Criteria

To set the criteria for analyzing the state of the art, Subsection 3.2.1 defines intrinsic significance indicators to weight specific aspects of a requirement, Subsection 3.2.2 then describes the requirements each technology category must satisfy, and, finally, Subsection 3.2.3 outlines the rating system to quantify the degree of fulfilling requirements.

3.2.1 Significance Indicators

For consistently determining the significance of diverse aspects of a requirement, we rely on both the request for comments (RFC) 2119 and the so-called *MoSCoW analysis* (Bradner, 1997; IIBA, 2009). As different characteristics of requirements have different implications on the success of this project, we interrelate each aspect with one out of the three levels listed below:

Must Fulfilling a certain aspect of a requirement is mandatory.

Should Fulfilling a certain aspect of a requirement is recommended.

Could Fulfilling a certain aspect of a requirement is optional.

Now that we have defined the significance indicators, we use them as keywords to weight each aspect of the requirements described in the following subsection.

3.2.2 Requirements

Software quality determines how well products fulfill certain requirements under particular conditions (ISO/IEC, 2011), yet it focuses on processing by the software and not on produced data. Software products, however, are often chained with each other in practice, i.e., the output data of one product is the input data of another. A prime example are mashup applications, which use, connect and combine data produced by other web applications and web services to create an improved experience tailored to certain scenarios (Chudnovskyy, Nestler, et al., 2012). Data quality therefore strongly determines how well software products can deal with inputs to generate outputs. Consequently, it is more difficult to generate high quality data from low quality inputs, e.g., incorrect, incomplete, inconsistent or incoherent data.

Based on the problem and objective descriptions, we derive several requirements per identified category in consideration of established standards, particularly the specifications of the software quality model *9126-1:2001* as well as its successor *25010:2011*, and the data quality model *25012:2008* by the International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) (ISO/IEC, 2001; ISO/IEC, 2008; ISO/IEC, 2011). Even though the ISO/IEC quality models taken by themselves would enable an extensive and differentiated analysis based on the numerous requirements they specify, we intentionally restrict this large set to five requirements per category for ensuring brevity, clarity and understandability. For selecting the requirements, we particularly

pay attention to aspects concerning the management of personal data, e.g., access, creation, extension, modification, ownership, and last but not least protection. These collective requirements consolidate many requirements according to ISO/IEC using five generic concepts. Such consolidation of requirements is not an unusual approach to adjust the focus, e.g., (Rafique et al., 2012) proposed a similar combination.

Requirements on Web Engineering Approaches

To improve the consideration of security aspects for web systems (cf. Result 1 and associated Activity 1.1), we make use of *interoperability*, *processability*, *scope*, *security* and *serviceability* as collective requirements on web engineering approaches including related processes, tools and models. Each requirement is elaborated on in the following:

Interoperability A WEA must produce artifacts, particularly web system models, that are usable independently from specific SPs (WEF, 2011). To prevent data and customer lock-in, a web engineering approach must foster interoperability for models, metamodels, transformations and tools (Koch, Meliá-Beigbeder, et al., 2008). The exchange between different modeling approaches should be supported through relying on open standards, which would also contribute to increase compatibility and reusability of modeled artifacts (WEF, 2011).

Processability A WEA must enable machine-independent accessibility, understandability and modifiability for produced models (ISO/IEC, 2011). That is, these models should be obtainable, interpretable and processable by human and non-human entities on both syntactic and semantic level, which includes that reliable language processors are available for enabling efficient operation and supporting human entities in their endeavors (WEF, 2011; Wild and Gaedke, 2014).

Scope A WEA must be sufficiently accurate, expressive and flexible to cover diverse scenarios and stages of development. Promptly responding to changing conditions requires adaptability and extensibility of existing software (Wild and Gaedke, 2014), which also contributes to ensure scalability, evolvability and maintainability of web systems (Koch, Meliá-Beigbeder, et al., 2008).

Security A WEA must support modeling security aspects at different abstraction levels by taking account of web systems as a whole and their individual components like web services (Wild and Gaedke, 2014). This requires facilities to 1) identify involved entities, 2) describe the thus obtained identities by appropriate attributes, e.g., interface definitions, and 3) specify the means for their protection, e.g., through access control (OECD, 2013). The necessity of identifying web systems and components throughout different development stages enables clarifying responsibilities and paves the way for authentication and accountability in later stages (ISO/IEC, 2011; OECD, 2013).

Serviceability A WEA and related tools must be available, reliable and technically accessible to support entities in modeling web systems (ISO/IEC, 2011; Wild and Gaedke, 2014), which includes options for efficient creation and adaption. Contributing to understandability and usability of architecture models by human entities asks for simple yet powerful graphical representation and editing capabilities (Koch, Meliá-Beigbeder, et al., 2008).

Requirements on Identity Management Systems

To reduce the need for accumulation of personal data by third parties on the one hand and to extend the means for control and protection of personal data on the other hand (cf. Results 2 and 3 and associated Activities 2.1 to 3.3), we employ *adequacy*, *control*, *openness*, *protection* and *transfer-*

ability as five collective requirements on identity management systems, with each requirement being detailed below:

Adequacy An IdMS must not only be technically accessible, but it should also be supportive, self-explanatory and easy to operate for the entities that employ it (ISO/IEC, 2011; Papazoglou et al., 2007; The White House, 2011). Along with supplying facilities for IdM, this includes allowing efficient processing of identity-related activities with special regard to authentication and access to identity data, and using human-memorable identifiers (El Maliki and Seigneur, 2007; Halpin, 2014). In addition to serviceability for users, an IdMS should be maintainable by developers and administrators to ensure its reliability and availability (ISO/IEC, 2011; The White House, 2011).

Control An IdMS must enable appropriate user control, imperative user consent and minimal disclosure of personal data, especially identity data, only to justifiable parties for a constraint use (cf. laws of identity) (Cameron, 2005; OECD, 2013). It must allow for claiming and maintaining ownership of personal data by actual data owners to prevent unauthorized accumulation and unwanted lock-in by third-parties, but without strongly interfering with companies' current models of fostering customer retention (OECD, 2013). Together with identity creation, SSO, attribute retrieval, and identity selection, an IdMS should also support single logout and federation establishment (Barisch, 2012). To properly take account of privacy needs, an IdMS must facilitate anonymity and unmappability¹², and support multiple identities per entity. Each identity should be made available either as self-asserted identity or as third-party asserted identity, e.g., issued by companies or government agencies (Cameron, 2005; Halpin, 2014; The White House, 2011). Supported by high descriptiveness of identities, an IdMS must handle different application areas to facilitate a consistent expe-

¹²The capability to conceal an entity's connections from discovery (Halpin, 2014).

rience across contexts (Barisch, 2012; Cameron, 2005), like in delegation scenarios (cf. Activity 3.1).

Openness An IdMS must offer global identification of entities in order to enable linkage and discovery unbound of specific IdPs (OECD, 2013; Papazoglou et al., 2007). This avoids data and customer lock-in, and facilitates connectivity to other entities. Moreover, an IdMS must be interoperable to ease cooperation with other IdMSs and to enable compatibility and reusability of personal data, especially identity descriptions (Barisch, 2012; ISO/IEC, 2011; The White House, 2011).

Protection An IdMS must provide confidentiality, integrity and privacy to allow for protecting against unauthorized read/write access to personal data (OECD, 2013; The White House, 2011). Furthermore, it should facilitate non-repudiation and accountability for clarifying responsibility of activities carried out by entities (ISO/IEC, 2011; OECD, 2013). Finally, an IdMS must enable authenticating human and non-human entities (ISO/IEC, 2011), which makes support of ownership-based authentication necessary¹³.

Transferability An IdMS must be portable to different environments to allow for broad adoption by users. Employing open standards on the protocol, interface and data type level facilitates transferability through compatibility and interoperability (Bria, 2012; ISO/IEC, 2011; WEF, 2011). High modularity of IdMSs conduce to their installability and adaptability.

¹³Unlike human, machines cannot keep secrets in mind, like passwords. Authentication of non-human entities by biometrics, i.e., by intrinsic characteristics such as temperature, utilization or fan rotation, is still an insufficiently explored field of research and, thus, immature for productive use. Consequently, non-human entities have to rely on some kind of persistent data, like a key or an access token.

Requirements on Identity Description Languages

To further contribute to reduce the need for accumulation of personal data by third parties (cf. Result 2 and associated Activity 2.1), we present *accuracy*, *appropriateness*, *assurance*, *interpretability* and *portability* as five collective requirements on identity description languages, which are described in the following:

Accuracy An IdDL must allow for representing entities according to their preferred level of detail in certain contexts in a consistent, precise and up-to-date way by relying on appropriate vocabularies (ISO/IEC, 2008; OECD, 2013; Rafique et al., 2012). Here, consistency should cover both the syntactic and the semantic level.

Appropriateness An IdDL must offer a rich set of expressive language constructs to appropriately represent entities according to their specification in certain contexts in a fairly complete manner (ISO/IEC, 2008; OECD, 2013). This requires that descriptions are extensible, which involves capabilities for expanding, linking and merging data sets (Halpin, 2014).

Assurance An IdDL should facilitate providing assurances in terms of availability, confidentiality, credibility, recoverability and traceability (ISO/IEC, 2008). It should not restrict distribution of parts of identity descriptions (e.g., as distinct web resources for enabling access control, availability or recoverability) and metadata attachment (e.g., signatures for fostering confidentiality, traceability or credibility).

Interpretability An IdDL must yield identity descriptions that are obtainable and processable by human and non-human entities, i.e., the resulting identity descriptions have to allow for machine-independent accessibility, understandability, and modifiability (ISO/IEC, 2008; Rafique et al., 2012). This necessitates the availability of language processors to enable efficient operation and support human entities in their endeavors.

Portability An IdDL must allow for migration, exchange and reusability of identity descriptions beyond the scope of specific IdPs in order to foster openness and, thus, reduce barriers for entering and exiting IdPs, and prevent discrimination caused by customer and data lock-in (Bria, 2012). Therefore, identity descriptions should be portable by relying on open, widely accepted standards (ISO/IEC, 2008).

After outlining the requirements, we continue with describing how to assess their fulfillment.

3.2.3 Rating System

Applying a point-based rating system for specifying the degree of fulfillment by a technology (group) in terms of a requirement allows for precise valuation, yet such absolute assessment is difficult to accomplish in practice (Hubbard, 2009). It involves taking the complete spectrum of technologies in a certain field into account in order to determine the minimum and maximum valuations per requirement a priori to the actual assessment. Furthermore, such absolute rating system would suggest doubtful precision also for subjective criteria assessments (Röder et al., 2009), like usability (Speicher, 2015), and, thus, falsely encourage comparability of technologies by characteristics via point and point differences. This, in turn, bears the risk of erroneous conclusions, e.g., a criterion assessed with value 14 is twice as good as one with value 7.

For these reasons, we rely on a relative rating system for determining the degree technologies fulfill the requirements given above. This relative rating system relies on a simplified cost-utility analysis and does not imply any exact conversion factor among certain degrees of fulfillment, but its ordinal scale provides a coherent four-step differentiation as described next.

- **Completely fulfilled** Technology does completely fulfill requirement.
- **Largely fulfilled** Technology does largely yet not completely fulfill requirement.
- **Partially fulfilled** Technology does fulfill requirement to some extent yet not largely.
- **Insufficiently fulfilled** Technology does not or not sufficiently fulfill requirement.

Now that we derived requirements and provided a rating system applicable to all technology assessments in this work, the next section continues with presenting the findings.

3.3 Analysis Results

Employing the criteria defined in the previous section, we discuss the results obtained by analyzing web engineering approaches in Subsection 3.3.1, identity management systems in Subsection 3.3.2, and identity description languages in Subsection 3.3.3. A summary containing a technology recommendation concludes each subsection.

3.3.1 Web Engineering Approaches

By assisting engineers in the systematic design, development and evolution of web systems, web engineering approaches intend to reduce error-prone and time-consuming activities associated with the manual construction and maintenance of web applications and web services (Koch, Meliá-Beigbeder, et al., 2008). WEAs aim at facilitating the engineering of web systems through models, processes and tools. In order to pursue this purpose, WEAs provide features like abstraction from implementation details, sepa-

ration of concerns, fostering flexibility and reusability, cost reduction, and automation the code generation. Here, models and components allow for such abstraction but with varying objectives, directness and affinity towards the technical implementation.

While models represent certain attributes of entities at different levels of details, components encapsulate technical or functional related parts in self-contained units and make functionality accessible through well-defined interfaces. Furthermore, models can act as blueprints for components. Both types of abstraction enable composition, with the option that models integrate other models and components consist of subcomponents.

Through adapting principles of conventional component-based software engineering to the characteristics of the web, engineers can interlink functional building blocks according to an underlying composition approach in order to establish component-based architectures such as today's prevalent service-oriented architectures (SOAs) (Gaedke and Rehse, 2000). In a SOA, web applications and web services represent components that provide web-accessible interfaces and might involve public or private subsidiary components. Not only do many WEAs enable engineers to model application aspects such as navigation and presentation, but also offer model-to-model and model-to-code transformations to automatically build web systems through appropriate tools (Koch, Meliá-Beigbeder, et al., 2008).

In order to discuss the analysis results of WEAs, we rely on the classification established in (Kappel et al., 2006) and updated in numerous publications including (Meinecke, 2008) and (A. Heil, 2012). This classification distinguishes between four primary orientations of web engineering approaches: data, hypertext, object and software (Kappel et al., 2006). While data-oriented approaches offer advantages when modeling data-intensive web systems due to their heritage from database systems, hypertext-oriented ap-

proaches focus on navigational aspects of web systems (Kappel et al., 2006; Meinecke, 2008). Object-oriented approaches make use of congeneric modeling languages, whereas software-oriented ones adapt many principles of conventional software engineering (A. Heil, 2012). In addition to these four groups, we also elaborate on the more implementation-directed group of component-based web engineering approaches.

Sorting analyzed approaches according to their focus into five groups allows for conducting the assessment on a more general level and for classifying new approaches without undermining conclusions drawn for those groups. With modeling of security aspects taking an important role, the discussion of analysis results starts with the group of data-oriented web engineering approaches, proceeds with hypertext-, object- and software-oriented ones, and ends with component-based WEAs. For each group, we provide a characterization, an assessment using the requirements (cf. Subsection 3.2.2) and examples.

Data-Oriented Approaches






Not just since the era of big data, data-intensive software systems make up a large share of all web-based solutions (Meinecke, 2008). This group of web systems focuses on presentation and modification of data and, therefore, heavily relies on one or more central data repositories (A. Heil, 2012). Here, users gain access to specific parts of the underlying data sets through appropriate interfaces. Data-oriented WEAs assist in the systematic development of web systems, with the set of applications spanning data-driven domains like e-commerce and social networking. In line with their orientation, approaches of this group primarily involve data-centric models such as the entity-relationship (ER) model (A. Heil, 2012).

Prime examples of data-oriented WEAs include the Web Modeling Language (WebML) (Ceri et al., 2000) and Hera (Houben et al., 2003). Beyond its original intention as a conceptual language for designing data-intensive web systems, several extensions increased the capabilities of WebML to cover modeling service, workflow, semantic and context-awareness aspects (Brambilla and Fraternali, 2014; Ceri et al., 2000). Based on the spiral model, WebML also proposes a design-centric, non-incremental process model for engineering web systems that consists of seven stages (A. Heil, 2012). WebML enables to specify structure, composition, navigation and presentation of web systems. Furthermore, its process model supports these specifications through three dedicated design stages that take account of data, hypertext and architecture. Specialized tools like WebRatio support the engineering process through features such as model-to-code transformations (Brambilla and Fraternali, 2014). As noted in (Meinecke, 2008), WebML can consider IdM aspects during modeling as a foundation for personalization and access control.

Similar to other data-oriented WEAs, WebML forces a strong binding between application and data model, which also includes security models like the identity model. This results in creating data models that are compatible to ER and Unified Modeling Language (UML) diagrams on the one hand, yet these models are specific to certain applications or domains on the other hand. By fostering SP-centric models to IdM, this entails deficits towards using, extending and managing data, with particular regard to personal data, across application domains (cf. remote silo model in Subsection 3.3.2). As an effort to resolve such limitation, Hera supports integrating diverse data repositories through relying on transformations based on semantic models (Vdovjak et al., 2003). However, the definition of integration engines increases complexity and represents a compromise towards the greatest possible intersection between data models (Meinecke, 2008).

Although data-oriented WEAs provide advantages for engineering data-centric web systems, they show deficits in creating and using models for heterogeneous domains, which limits the scope. Models and tools of this group are appropriate and lead to a reasonable level of serviceability, with benefits introduced by them being primarily evident during the stages of design and code generation, but lag behind during runtime. The definition and management of security-related aspects is specific to particular domains (or SPs respectively) and, thus, restricts universal processability and interoperability. Table 3.1 summarizes the results for data-oriented WEAs and corresponding tools using the symbols defined on page 77.

Table 3.1: Analysis Results of Data-Oriented Web Engineering Approaches

Interoperability	Processability	Scope	Security	Serviceability
				

Hypertext-Oriented Approaches

Focusing on navigational aspects, hypertext-oriented WEAs specially support modeling structural characteristics of web systems on the conceptual foundation established through hypertext (Nelson, 1994). Based on content models, engineers can describe available nodes and edges using specialized notations, where nodes represent certain information fragments and edges link interrelated nodes together according to underlying requirements (Kappel et al., 2006). This allows for creating different perspectives on the information space for different viewers.

Prime examples of hypertext-oriented approaches are the Hypertext-Design Model (HDM) (Garzotto et al., 1993) and the Web-Site Design Method (WSDM) (Troyer and Casteleyn, 2003). By providing a platform-independent hypertext model, HDM supports the design phase

of engineering hypermedia web systems. HDM divides the design in navigation structure and documents, and enables to specify diverse types of references including structure, perspective and application (Gaedke and Gräf, 2000). Similar to HDM, WSDM is primarily intended for engineering web systems that allow users to consume and navigate through the information space. WSDM implements a user-centric modeling approach that assists engineers in conceptual design and physical design (Troyer and Casteleyn, 2003).

Partially attributed to an incomplete support by models, tools and guidelines throughout the web engineering process, both examples do, however, show deficits in developing complex web systems or web systems that do not represent information systems (Gaedke and Gräf, 2000). Here, HDM does not define an own process model, yet it is combinable with conventional software engineering approaches (Garzotto et al., 1993). Not relying on systematic and comprehensive process models, HDM and WSDM neglect reusability and, thus, reduce their applicability with implications on interoperability, processability, scope and serviceability. These issues led to further developments, like Object-Oriented Hypermedia Design Method (OOHDM), which involve dedicated process models with increased coverage of other engineering stages.

Hypertext-oriented WEAs offer only insufficient means to develop web systems with focus on security. By primarily addressing navigational characteristics in engineering non-complex web systems, they fall short in other engineering stages. In consequence, hypertext-oriented approaches became obsolete and were superseded by other WEA groups, which incorporated concepts of hypertext-orientation. Through employing specialized proprietary models, domain-specific languages (DSLs) and tools for describing structural aspects of web systems, hypertext-oriented approaches furthermore lack in largely fulfilling other collective requirements, like interoperability

and processability. As summarized in Table 3.2, this group of web engineering approaches has little to offer for achieving the purpose of this work.

Table 3.2: Analysis Results of Hypertext-Oriented Web Engineering Approaches

Interoperability	Processability	Scope	Security	Serviceability
◐	◐	○	○	◐

Object-Oriented Approaches

In order to provide a holistic approach to engineer web systems, object-oriented WEAs rely on the conceptual foundation of objects that comprise structure as well as behavior (Kappel et al., 2006). By enabling to model both structural and behavioral aspects of web systems at the levels of content, hypertext and presentation through appropriate diagrams, UML acts as a general-purpose language (GPL) many modeling methods are based on (A. Heil, 2012; Kappel et al., 2006). Furthermore, UML facilitates creating human-interpretable design artifacts, which can serve as generally accepted specification and documentation of the engineering process.






OOHDM (Rossi and Schwabe, 2008) and UML-based Web Engineering (UWE) (Koch, Knapp, et al., 2008) represent prime examples of object-oriented approaches. Based on UML, UWE describes guidelines for using general modeling constructs and maps them to the particular requirements of engineering web systems (Meinecke, 2008). Employing special profiles, it covers aspects like navigation and presentation at a highly abstract level, and supports modeling content objects through UML class diagrams. Like UWE, OOHDM includes a dedicated process model, which consists of conceptual, navigation and interface design as well as implementation (Rossi and Schwabe, 2008). In addition to utilizing UML for

design matters, OOHDM enables to semantically specify conceptual models of web systems (Rossi and Schwabe, 2008).

However, UWE and OOHDM lack support for integrating content objects from third-party sources like web services. For controlling access to specific nodes of the navigation model, UWE allows for specifying digital identities and setting authorization preferences within the model. Despite this advantage in modeling security aspects, UWE and OOHDM do neither detail the implementation of such aspects nor foster the application-independent use of identities and access control preferences (Meinecke, 2008). Both prime examples focus on non-implementation characteristics of web systems.

Object orientation in the analyzed approaches of this group applies primarily to models and only secondarily to actual implementations, whereby code generation supports creating object-oriented implementation artifacts. All evolutionary steps, e.g., evoked by changing requirements, entail possibly complex adjustments to the models before code modifications can happen, which limits the scope. Moreover, a large set of complex models as necessary requirement complicates interpretability and processability by human entities and makes security matters difficult to incorporate properly. With object orientation being a common paradigm in software engineering, there is a solid level of standardization and broad support by tools, e.g., for modeling. This results in largely fulfilling the collective requirements regarding processability and serviceability, where the latter is, however, impaired by insufficient assistance during implementation. Table 3.3 shows these findings as an overview.

Table 3.3: Analysis Results of Object-Oriented Web Engineering Approaches

Interoperability	Processability	Scope	Security	Serviceability
				

Software-Oriented Approaches

On the methodological basis of conventional software engineering, recent software-oriented WEAs pursue creating model-driven architectures (MDAs) by involving multiple models at different abstraction layer (Meliá and Gómez, 2005). Approaches of this group are therefore especially suited for implementing projects with clear functional requirements, so that benefits from structured models, separation of concerns and standard platforms have a clear effect (Koch, Meliá-Beigbeder, et al., 2008). When approached systematically, a MDA allows for addressing interoperability, model evolution and adaption, and for bridging the gap between high level design by computational independent models (CIMs) and platform independent models (PIMs) described using DSLs and low level implementation by platform specific models (PSMs) and source code (Koch, Meliá-Beigbeder, et al., 2008).






The Web Software Architecture (WebSA) (Meliá and Gómez, 2005) is a prime example of a software-oriented approach. Unlike other approaches, WebSA does not introduce own DSLs and modeling methods for describing aspects such as navigation, presentation and content, but it reuses existing approaches like UWE. To develop web systems on the MDA foundation, WebSA utilizes UML for describing PIMs as well as transformation engines for converting these models to PSMs and source code (Kappel et al., 2006). WebSA offers an architectural perspective in engineering web systems, which distinguishes between coarse-grained subsystem models and fine-grained configuration models (Meliá and Gómez, 2005).

Similar to component-based approaches, which are discussed next, WebSA can represent building blocks of a web system as components (Meliá and Gómez, 2005). However, it does not elaborate on how models and components reflect changes made through maintenance and further development.

Although a so-called integration model enables to join various model types, they have to be from known sources, i.e., there is a lack in support of external web services (Meinecke, 2008). WebSA does not inherently cover IdM.

By combining advantages of models and components, software-oriented web engineering approaches that follow the MDA concept strive for providing a passable compromise between abstraction and implementation focus. With security treated as an afterthought during engineering, scope impaired through missing capabilities for driving evolution and change management appropriately, and processability as well as serviceability reduced by external dependencies, only interoperability can score owing to the (non-semantic) model basis, as outlined in Table 3.4.

Table 3.4: Analysis Results of Software-Oriented Web Engineering Approaches

Interoperability	Processability	Scope	Security	Serviceability
				

Component-Based Approaches

In contrast to the methodologies presented so far, component-based web engineering (CBWE) approaches assist in rapid development of web systems by focusing on implementation through software building blocks that are ready to use with none or only a few adjustments. The four other WEA groups involve thoroughly analyzing the problem domain before creating domain-specific models based on analysis results, whereas CBWE does not require such exhaustive investigation of the problem domain and subsequent complex designs. Components offer a lower level of abstraction compared to models, but they facilitate adaptability, reusability and composition, especially when a large set of building blocks is available. Component-based approaches enable composing web systems

by orchestrating reusable, existing components, like web services, towards a more complex service offering (Schill and Springer, 2012). However, models and components are not mutually exclusive, as the group of software-oriented approaches has shown.






Prime examples of component-based approaches include WebComposition (Gellersen and Gaedke, 1999), the portlet-centric web development approach (Díaz et al., 2008) and the Open Mashup Enterprise service platform for LinkEd data in the Telco domain (OMELETTE) approach (Chudnovskyy, Nestler, et al., 2012). WebComposition describes an incremental, iterative development process adopted from the spiral model that takes account of changing requirements of web systems (Gaedke, 2000). This so-called WebComposition Process Model (WCPM) involves analysis, design, implementation and evolution (Gaedke and Gräf, 2001). Being both a reusability- and a life-cycle-centric approach, WebComposition defines a repository of design artifacts and organizes the evolution into three cyclic stages, which cover analysis, design and implementation (Gaedke, 2000). In (Trujillo et al., 2007), the authors propose a MDD process to create web applications, named portals, by making use of portlets as building blocks. While portlets offer dedicated services like components do, they additionally “encapsulate the presentation layer and all navigation that goes with it” and, thus, are primarily suited for building user-facing applications (Díaz et al., 2008). OMELETTE describes an open reference architecture, where engineers can build web-based solutions upon, and assists them by integrating component repositories and by providing tools like the automatic composition engine to accelerate the engineering process (Chudnovskyy, Nestler, et al., 2012).

Employing specific models and languages, WebComposition supports 1) formalizing web-specific characteristics based on object-oriented and technology-independent component-based modeling with the WebComposition Component Model (WCCM), 2) describing components in

a non-semantic manner with the eXtensible Markup Language (XML)-based WebComposition Markup Language (WCML), and 3) modeling interconnected building blocks of web systems in consideration of federation aspects with the WebComposition Architecture Model (WAM) (A. Heil, 2012; Meinecke and Gaedke, 2005). Using a UML-like notation, WAM allows for characterizing essential entities and their associations within architectures of web systems (Meinecke and Gaedke, 2005). An architecture model described by WAM can represent different connection types, “organizational zones of control over networks, hardware and software system” called security realms, and entities classes including web applications and web services, in both disassociated and federated contexts. Enabling to swiftly incorporate changes, WAM is a proven yet proprietary formalism for modeling distributed solutions in consideration of trust and security aspects on a technology-independent, service composition level (Meinecke and Gaedke, 2005; Meinecke et al., 2007). As part of the reference architecture, OMELETTE defines a dedicated IdP that can integrate identities of external domains in addition to performing common IdM activities.

Components are a generic, uniform concept for defining artifacts of web systems. Unlike the models used in other web engineering approaches, components are stronger bound to specific platforms and, thus, restrict interoperability and require further means for description, interpretability and processability. Facilitating the composition of web systems, CBWE approaches provide sufficient yet limited scope and serviceability, while taking aspects of security into account. Nevertheless, questions in terms of identification, discovery, linking, access and protection of modeled components remain not completely answered. Table 3.5 presents a summary of the analysis results for CBWE approaches.

Table 3.5: Analysis Results of Component-Based Web Engineering Approaches

Interoperability	Processability	Scope	Security	Serviceability
				

Summary of Web Engineering Approaches

Approaches that rely on models for driving the engineering of web systems intend to provide adaptability and reusability through abstraction in the design phase, yet they yield solutions that are tailored to particular scenarios or use cases, and remain below their potential during runtime. Even though such tailoring offers advantages, like an increase in efficiency, for domain-specific engineering efforts, data-, hypertext, object- and software-oriented WEAs bear deficits in interoperability through customizing models, including security models, to the requirements of particular domains. Web engineering models can describe web systems independently of platform and implementation language, yet these PIMs do not have this level of abstraction in common in order to represent CIMs (Koch, Meliá-Beigbeder, et al., 2008). WEAs permit building similar types of models, but employ different graphical notations for them. The variety of existing domain-specific and often proprietary model description languages, e.g., UML with additional notations, makes interoperability difficult, as stated in (Koch, Meliá-Beigbeder, et al., 2008).

Limiting domain-independent use of models entails negative consequences for processability, scope, security and serviceability, which becomes especially evident during runtime. Interoperability issues of models and associated tools implicitly impair serviceability particularly for human actors because they need to adjust to varying conditions, capabilities and features, which requires additional training efforts. The web engineering community just started addressing the topic of security in the last couple of years and

has not yet provided an integral and widely adopted solution. By rejecting a security approach covering multiple domains, IdM is bound to individual applications by design, which in turn affects various runtime aspects like the expressiveness of identities through limited attribute sets, restricted access control settings or non-uniform management interfaces. This does not only promote lock-in of users and their personal data, increase redundancy and weaken (re-)usability, but it also interferes with Results 2 and 3. Data-, hypertext, object- and software-oriented WEAs impede rapid prototyping and agile development through necessitating a complex foundation of models before code generators can create implementation fragments. That is, engineers that manually make adjustments or add extensions directly to the source code have to properly reflect these changes within models in order to avoid their automatic removal during future code generation based on outdated models (Saleem et al., 2014). Unlike methodologies that focus on obtaining customer feedback as early as possible through running software, the plurality of web engineering approaches postpone the technical implementation to a late stage in engineering.

Endeavors such as (Rivero et al., 2012) try to combine advantages of different engineering methodologies; others like (Wimmer et al., 2007) aim at increasing interoperability by establishing migration paths among different WEAs (Vallecillo et al., 2007), yet they introduce model transformations rather than aligning the underlying engineering process (A. Heil, 2012). In (Koch, Meliá-Beigbeder, et al., 2008), the authors recommend introducing a common metamodel to 1) partially meet the challenges with respect to increasing universality and processability, and to 2) bear positive effects on various engineering aspects like having holistic descriptions and simplifying identification of arbitrary entities. Notwithstanding these efforts, today's data-, hypertext, object- and software-oriented WEAs only represent a compromise, which does neither fully serve engi-

neering universally applicable nor completely custom-tailored web systems that sufficiently focus on security.

Component-based web engineering approaches like WebComposition go without a large set of detailed models in order to facilitate development, maintenance and evolution of web systems and, thus, be in line with agile software development methodology. While component-based methodology shows advantages in scope, security and serviceability, they lack in profoundly addressing interoperability and processability through a holistic approach of describing and identifying building blocks of web systems at the individual and composite level. Enriching CBWE by a light-weight yet sound foundation of models bears the potential for not only improving diverse characteristics such as specification, search and selection of suitable components, but also for providing an integral perspective on security for both human and non-human entities.

Before the next subsection continues with discussing the outcome of the IdMS analysis, Table 3.6 summarizes the results for web engineering approaches and associated tools using the criteria specified in Section 3.2.

Table 3.6: Analysis Results of Web Engineering Approaches and Corresponding Tools in Terms of Collective Requirements

WEA	Collective Requirements					
	Orientation	Interoperability	Processability	Scope	Security	Srv.-ability
Data	●	●	●	●	●	●
Hypertext	●	●	○	○	○	●
Object	●	●	●	●	●	●
Software	●	●	●	●	●	●
Component	●	●	●	●	●	●

3.3.2 Identity Management Systems

For analyzing IdMSs, we adapted and extended the classification proposed in (Jøsang et al., 2015; Jøsang et al., 2007), with the result of distinguishing overall eight models to IdM, where five models are primarily SP-centric and three models are primarily user-centric. SP-centric models are designed to fulfill the needs of service providers in the first place and, thus, SPs are in charge of storing and controlling personal data of users. In contrast, user-centric models empower individual entities to self-manage their personal data by enabling them to exercise control about storage and access.

The discussion of analysis results starts with SP-centric models and proceeds with user-centric models. For each model, we provide a characterization, an assessment with regard to the requirements stated in Subsection 3.2.2 and examples.

Remote Silo Model

By enabling SPs to authenticate users without requiring external IdPs, the remote silo model is a SP-centric model, where each SP also acts as IdP (Jøsang et al., 2007). In addition to their actual service offering, SPs therefore provide custom-tailored facilities for IdM, e.g., handling of authentication tokens and identity data of users. For historic reasons, it is a widely adopted model. Based on (Jøsang et al., 2007), Figure 3.1 illustrates this the remote silo model using the WAM notation, as discussed in Subsection 3.3.1 and described in (A. Heil, 2012). Here, an entity e requires separate credentials for each individual identity managed at different SPs ($SP_1, \dots, SP_j, \dots, SP_n$). Each of those SPs acts in its own domain or, to be more precise, security realm (SR) ($SR_1, \dots, SR_j, \dots, SR_n$) with an inherent IdP ($IdP_1, \dots, IdP_j, \dots, IdP_n$) and a silo ($Silo_1, \dots, Silo_j, \dots, Silo_n$) storing personal data associated with an identity ($i_1, \dots, i_j, \dots, i_n$) of entity e .

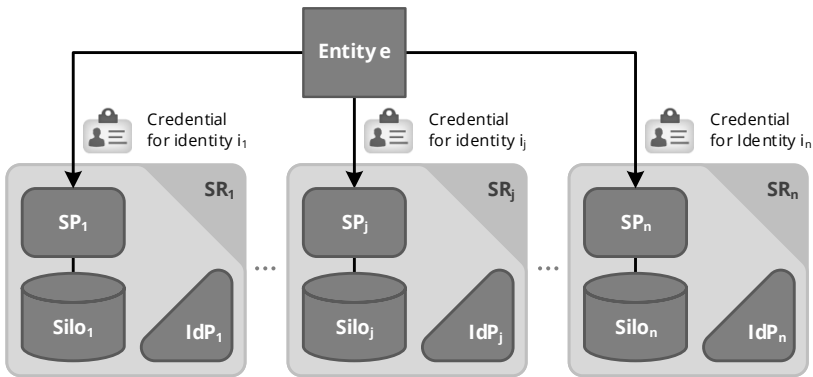







Figure 3.1: Remote Silo Model (adapted from (Jøsang et al., 2007))

Prime examples include e-commerce applications and online banking applications. The remote silo model is simple in deployment, but it is designed for the individual conditions of specific SPs, e.g., an OSN would have different needs to IdM in comparison to an online book store application. Its use is adequate within so-called “walled gardens” and it allows for sharing personal data only with SPs intended by a user (Yeung et al., 2009).

On the downside, it bears problems in handling many identifiers and associated credentials which users can only employ for individual SPs with their particular namespaces (Jøsang et al., 2007). This can entail so-called “password fatigue”, which has implications on control and protection (Sun et al., 2010). While password fatigue is evident for all SP-centric models to IdM, especially the remote silo model promotes this negative effect due to a vast number of passwords users need to remember. This in turn can provoke poor password quality and high password redundancy (Florêncio and Herley, 2007). Users need to get along with a variety of individual IdM facilities, which most likely differ between SPs. Having to deal with different interfaces that offer different options for IdM decreases adequacy

in terms of serviceability, particularly for users that rely on many SPs. Moreover, the silo character of this model largely impairs openness and transferability, e.g., by limiting identity linkage and discovery only to a specific IdP (or SP) and fostering proprietary IdM solutions, which in turn are hardly portable. Table 3.7 summarizes the results for the remote silo model to IdM using the symbols introduced in Subsection 3.2.3.

Table 3.7: Analysis Results of Remote Silo Model

Adequacy	Control	Openness	Protection	Transferability
				

Common Domain Model

In contrast to the remote silo model, SPs and IdPs are separate entities in the common domain model. Here, an IdP acts as central authority and, thus, is responsible for several SPs sharing a namespace. That is, a central authority carries out the IdM for a common domain. This simplifies management of authentication tokens as well as enables to unify or reduce identifiers and corresponding credentials, yet authentication per se is done by SPs individually (Jøsang et al., 2007). Figure 3.2 depicts the common domain model. It is obvious that entity e has to employ only one credential for identity i_h to authenticate to several SPs in a common domain SR_h with a shared IdP_h and SP-specific data silos.

Prime examples include public key infrastructures (PKIs). When based on certificate authorities (CAs) as central authorities, a PKI represents a centralized trust model that uses hierarchically organized authority chains (Caronni, 2000). It is a “set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates” (Stallings, 2010), which defines CAs for issu-

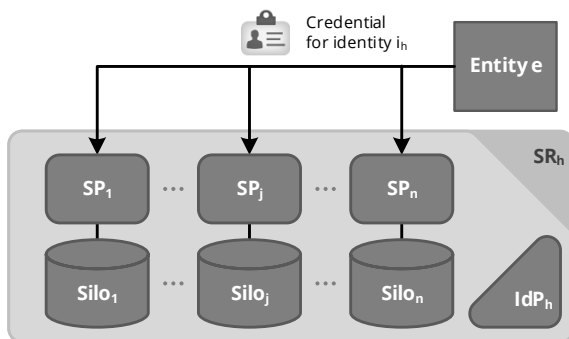







Figure 3.2: Common Domain Model (adapted from (Jøsang et al., 2007))

ing digital certificates, establishing trustworthy certification chains, and providing the infrastructure to verify the integrity of signed certificates. That is, a PKI associates digital certificates with physical entities after passing a strong review process. Here, an entity does only obtain a third-party asserted identity, supported by a CA-signed certificate, after it has intensely proven the claims made.

A PKI, however, bears several risks as discussed in (C. Ellison and Schneier, 2000), e.g., the centralization of authority, security and trust. If used in the wrong hands, this can facilitate discrimination and espionage. These risks do not solely apply to PKI, but to the common domain model. On a global scale, this SP-centric model shares drawbacks in terms of control, openness, protection and transferability with the remote silo model, albeit in a somewhat weaker form. Such mitigation results from the consolidation of IdPs, with the corresponding domains creating wider yet still walled gardens (Yeung et al., 2009). The common domain model is inappropriate when it comes to identity-related interactions across domains, such as reuse of identifiers and identity data. Table 3.8 outlines these analysis results.

Table 3.8: Analysis Results of Common Domain Model

Adequacy	Control	Openness	Protection	Transferability
				

Centralized SSO Model

Similar to the common domain model, a central authority represented by an IdP serves multiple SPs per domain sharing the same namespace in the centralized SSO model. However, authentication is not individually done by SPs, but happens in a centralized manner through an IdP per domain (Jøsang et al., 2007). IdPs consequently have to send security assertions, i.e., assertions to be checked in a security architecture, to SPs either directly or indirectly via user agents. While this requires a mutually agreed upon policy on IdM within a domain, it allows for SSO and, thus, slightly contributes to adequacy and protection. That is, the model is suited for more complex yet clearly defined environments with multiple SPs, like in large organizations of the public or private sector. Figure 3.3 illustrates the centralized SSO model for a single domain SR_h , with entity e authenticating either explicitly (as depicted) or implicitly to a domain-wide IdP $_h$ using an appropriate credential for identity i_h . After successful authentication, entity e can access SPs in that specific domain by means of security assertions obtained by IdP $_h$.

Kerberos is a prime example of an implementation of this SP-centric model (Jøsang et al., 2007). As summarized in Table 3.9, the centralized SSO model shares many risks in open environments with the common domain model, which is again due to the fact that many SPs would have to trust few authorities represented by IdPs.

Table 3.9: Analysis Results of Centralized-SSO Model

Adequacy	Control	Openness	Protection	Transferability
◐	◐	◐	◐	◐

Multi-Domain SSO Model

Complementing the centralized SSO model, the multi-domain SSO model involves support for multiple IdPs, where each is responsible for a particular domain. The SP-centric model disallows direct interactions between IdPs and SPs, i.e., all communication is redirected through user agents. As the model is similar to the centralized SSO model for the most part, the already introduced Figure 3.3 sufficiently illustrates the multi-domain SSO model with the addition that entity e needs to employ separate credentials for separate identities maintained at diverse domains.

A prime example of this model is InfoCard (Jones and McIntosh, 2008). The ownership-based InfoCard approach tries to consistently represent digital identities by so-called information cards, which are “analogous to

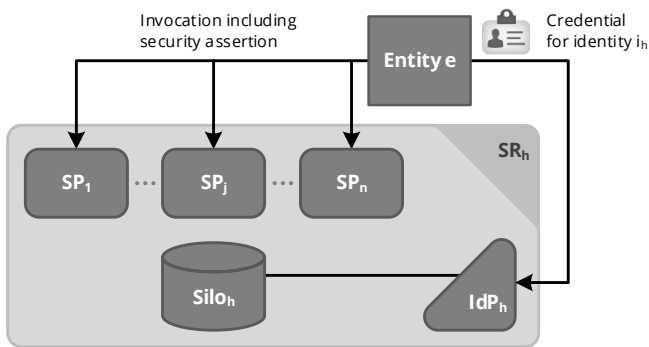


Figure 3.3: Centralized SSO Model (adapted from (Jøsang et al., 2007))

[...] physical identity and membership cards” (Hackett and Hawkey, 2012; Maler and Reed, 2008). Such information cards either self-contain a set of identity attributes or link to them, when stored on IdPs (Jøsang et al., 2007). Although Microsoft decided to stop development for its InfoCard implementation (Windows CardSpace) in 2011, the InfoCard approach remains a valuable IdMS contribution (Barisch, 2012), which open-source projects like *Higgins* (Trevithick, 2016) and *Open InfoCard* (Openinfocard, 2016) adopted and evolved further. InfoCards can represent both self-asserted identities and third-party asserted identities (Maler and Reed, 2008). For selecting an InfoCard-based identity, users need a special client tool (Dhamija and Dusseault, 2008).

Although these specialized agents contribute to improve control for users with regard to identity selection, confirmation and redirection of identity data or tokens that reference IdPs storing relevant personal data, they also add dependencies that limit the applicability of the multi-domain SSO model in different environments. To implement this model, operating system or browser vendors would have to integrate appropriate support (Jøsang et al., 2007). This in turn impedes transferability and openness, where the latter is further weakened—in case of InfoCard—through relying on non-web-compliant, binary identifiers that restrict universal linkage and discovery (Jones and McIntosh, 2008). In (Barisch, 2012), the authors outline deficits in adequacy that are caused by “complex software development [of InfoCard] for IdPs and [RPs]”. Like in the centralized SSO model, users need to entrust personal data to third-party IdPs, where in case of InfoCard such data consists of a neither extensible nor machine-readable attribute set (Jøsang et al., 2007; Maler and Reed, 2008). Table 3.10 presents an overview of the analysis results for the multi-domain SSO model.

Table 3.10: Analysis Results of Multi-Domain SSO Model

Adequacy	Control	Openness	Protection	Transferability
●	●	●	●	●

Federated SSO Model

In addition to independently authenticating users for purposes that relate to an individual SP, the federated SSO model enables SPs to recognize identities through accepting security assertions on pre-authenticated users from trusted SPs/IdPs (Cantor et al., 2005b; Jøsang et al., 2007). As of this writing, the federated SSO model is a widely applied SP-centric model that allows for mapping domain-specific identifiers upon each other and, thus, form a federated domain, where each SP remains responsible for its own namespace. This assists in referring to the same entity, even though different identifiers are present. As a consequence, the model is compatible to the remote silo model and also facilitates SSO in virtually open environments, i.e., within federations of arbitrary size. Figure 3.4 depicts the model, where SPs mutually trust each other; however, this is not stringently required.

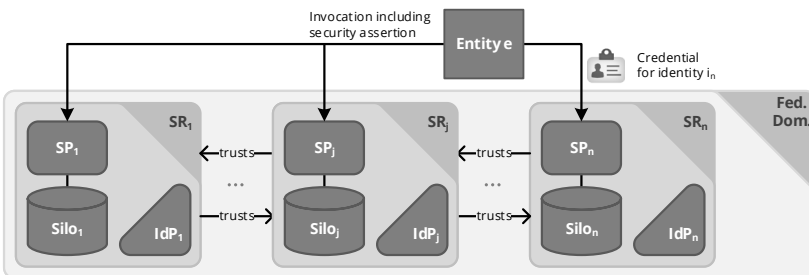


Figure 3.4: Federated SSO Model (adapted from (Jøsang et al., 2007))

OpenID (Fitzpatrick et al., 2007) and Shibboleth (Cantor et al., 2005a) are prime examples that actualize the federated SSO model. Other efforts for assembling federated web applications include the yet abandoned Liberty federation and the identity federation system (IdFS). In order to enable SSO in accordance with the WS-Federation specification, IdFS provides an infrastructure consisting of IdPs, security token services and modules for linking web applications and web services (Meinecke and Gaedke, 2005). Moreover, social login providers such as Google adapted OAuth 2.0 (Hardt, 2012), which is an authorization protocol in the first place, for authentication and especially for SSO purposes (Hühnlein, Wich, et al., 2014). Here, RPs have to trust these social login offering IdPs (Jøsang et al., 2015). As with OpenID Connect (Sakimura et al., 2014) OAuth and OpenID are merged, OAuth is not separately taken account of in the analysis of this model.

Both OpenID and Shibboleth cover similar use cases, but differ in their individual implementation in various aspects, e.g., valid initiators of an authentication, support for bindings and for single sign-out. Shibboleth is based on the Security Assertion Markup Language (SAML), which is the “most prominent standard for identity federation” (Jøsang, 2014) that consists of a “set of specifications describing security assertions that are encoded in XML, profiles for attaching the assertions to various protocols and frameworks [...]” (Hodges et al., 2005). On the other hand, OpenID represents a “lightweight alternative to SAML-based systems” (Barisch, 2012), which focuses on decentralization, easy setup and reduction of the complexity of SAML. For instance, OpenID tolerates only a subset of identifiers allowed in SAML, yet it provides Uniform Resource Identifier (URI)-based identification of primarily human entities (Cantor et al., 2005b; El Maliki and Seigneur, 2007). Although the OpenID specification does not detail protection and trust matters on purpose, it is typically deployed using passwords as proof of a user’s identity for the sake of applicability and compatibility (Barisch, 2012; Maler and Reed, 2008). Unlike Shibbo-

leth, OpenID does neither require an already established trust relationship between IdP and RP nor it involves central authorities to approve new RPs (Fitzpatrick et al., 2007; Hackett and Hawkey, 2012). With large IT companies offering IdPs and enabling skilled users to do so on their own, the adoption of OpenID swiftly increased, so that it is considered as well-adopted by users today (Hackett and Hawkey, 2012). For many SAML-based implementations (Mayer, 2013), like Shibboleth, the same holds true in more constrained environments, e.g., organizations of the public sector.

For effective operation, the federated SSO model is designed to map identifiers among SPs. This, however, bears risks towards correlation of user information within a federation (Jøsang et al., 2015). Moreover, such mapping of identifiers is only insufficiently supported in practice in case of OpenID, with the result that some RPs confuse users by offering numerous choices for IdP-specific authentication, which is also known as the *NASCAR problem* (Messina, 2009). That is, users have to trust SPs and IdPs of a federation for properly handling and protecting their personal data. This is contradicted by the limited inherent capabilities OpenID and Shibboleth provides to protect personal data against unauthorized access (like integrity protection) and to assign, process and exchange self-explanatory attributes. Even though extensions aim at resolving the latter issue for OpenID 2.0 (Hardt et al., 2007), the direct application of these IdMSs remains restricted to certain environments. Notwithstanding that Shibboleth only permits strict XML-based attributes and OpenID Connect, as successor of OpenID, relies on lightweight JavaScript Object Notation (JSON), they both require clearly defined attribute sets (Hughes et al., 2005; Sakimura et al., 2014). Without pre-negotiating vocabularies or introducing mediators, this lack of control and openness in terms of flexibility and extensibility makes these representatives of the federated SSO model inappropriate for attaching further descriptive data to identifiable

human and non-human entities in a holistic fashion. The analysis results for the federated SSO model are condensed in Table 3.11.

Table 3.11: Analysis Results of Federated SSO Model

Adequacy	Control	Openness	Protection	Transferability
●	●	●	●	●

Omitted Silo Model

In order to elevate user control, the omitted silo model provides authentication that involves exchange of only a minimal attribute set characterizing an identity. For the sake of minimal disclosure of personal data, the model does neither consider advanced attribute handling nor storage of identity data within silos. Figure 3.5 illustrates the omitted silo model with entity e authenticating to a trusted security token provider (in this case a certificate provider) that involves IdP*. Having obtained a security assertion after verifying the credential for identity i , entity e can access diverse model-compliant SPs as an authenticated user.

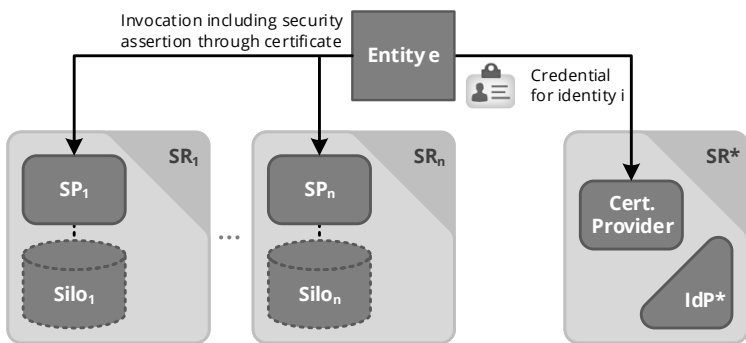







Figure 3.5: Omitted Silo Model

BrowserID (Mozilla, 2013) is a prime example of this user-centric model. It represents a hybrid knowledge/ownership-oriented IdMS that relies on email addresses as identifiers and internally hands out certificates to authenticated users (Akhawe et al., 2013; Bai et al., 2013). To prove ownership of an email address, a BrowserID-based IdP issues a certificate to a user after successful password-based authentication. Within a default time frame of 24 hours, users can employ such certificates without requiring any further IdP interaction (Bamberg et al., 2013). This contributes to control and protection through making user tracking by IdPs difficult (Dietz and Wallach, 2014). While (existing) email providers are primarily intended as certificate issuers, Mozilla acts as a fallback. BrowserID is designed for performing cryptographic operations on the client side, which makes support for JavaScript obligatory (Hackett and Hawkey, 2012). Before each authentication attempt, BrowserID asks for explicit user consent, which allows for anonymity, but it also intentionally prevents automatic SSO (Hackett and Hawkey, 2012). Characteristic for the omitted silo model, attribute handling and exchange is of little relevance in BrowserID.

Although the omitted silo model per se enables comprehensive control and protection, the abandonment of an inherent attribute management bears negative consequences for openness and adequacy with regard to serviceability. As a result of these missing capabilities, the model compels users to store attributes individually per SP, i.e., which is similar to the remote silo model with regard to attribute handling. Through relying on JavaScript, BrowserID shows additional deficits in protection and transferability, e.g., it appears vulnerable to phishing attacks due to login web page manipulation (Hackett and Hawkey, 2012). Furthermore, the precondition of an all-encompassing support for efficient, client-side JavaScript execution is not fully satisfied, especially when taking account of aspects such as older devices, different web browsers or opposing user preferences. Table 3.12 summarizes these results.

Table 3.12: Analysis Results of Omitted Silo Model

Adequacy	Control	Openness	Protection	Transferability
				

Local Silo Model

Providing extensive control for users similar to the omitted silo model, the local silo model involves an external, isolated and trusted device to verify identity claims and (optional) store personal data (Jøsang et al., 2015). Such personal authentication device (PAD) allows for authenticating a user locally at first and then performing the login to a RP on behalf of the pre-authenticated user either automatically or with explicit user consent. A PAD can therefore be seen as some sort of personal IdP, which enables a universal SSO solution (Jøsang et al., 2007). Figure 3.6 shows the local silo model, where entity e authenticates towards the PAD once. A SR represents the PAD in this figure. It also includes an IdM application, a data silo storing personal data as well as credentials for the SPs used by entity e , and an authentication service acting in concert with an IdP. After authenticating to the PAD, entity e can invoke diverse SPs of multiple domains, with having access conditions negotiated automatically between PAD and each SP.

OffPAD is a prime example of this user-centric model (Jøsang et al., 2015). Other examples include FutureID (Hühnlein, Schmolz, et al., 2014) or SkIdentity (Hühnlein, Hornung, et al., 2014), which also involve smart cards, but additionally make use of mediators for bridging among diverse IdM concepts (Hühnlein, Wich, et al., 2014). For the sake of focus on the local silo model, these examples are neglected in the analysis. Undertaking the management of identities and credentials decoupled from the environment usually employed for accessing web contents, OffPAD strengthens security and improves usability (Jøsang et al., 2015). OffPAD supports user

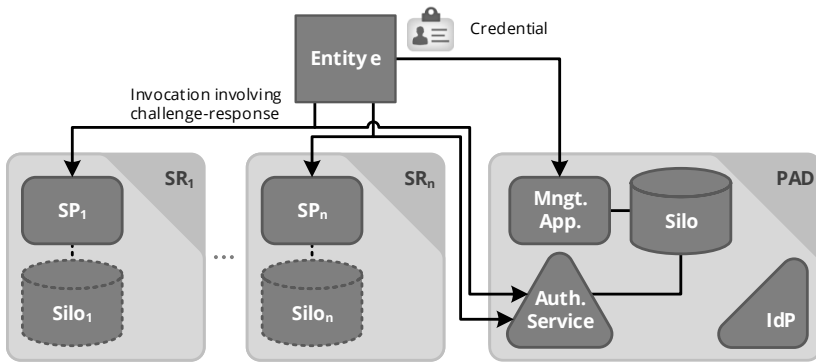







Figure 3.6: Local Silo Model (adapted from (Jøsang et al., 2007))

authentication through extending the HyperText Transfer Protocol (HTTP) challenge-response framework, so that security-relevant operations are outsourced to the PAD (Jøsang et al., 2015). To improve privacy, OffPAD only permits access to attributes for the time defined by the device. Furthermore, OffPAD offers a common user interface to IdM independently from SPs.

Notwithstanding the advantages in control and protection, the local silo model shows deficits in terms of openness, transferability and also adequacy particularly in scenarios that deal with heterogeneous environments. For authentication, a user needs to carry an additional device in form of a PAD, where existing mobile and desktop platforms, operating systems or web browsers have to be capable interacting with it in a trustworthy and transparent manner. This bears the risk of introducing potential weak points due to having to support various systems. What is a blessing for privacy introduced by the PAD can be a curse for RPs with regard to accessibility to identity attributes when the user in question is unavailable or offline. Moreover, it is questionable how non-human entities can employ the local silo model. Finally, the model does not specify what attributes can be stored in which way on the device or smart card and how they can be exchanged

with RPs, e.g., the so-called “eID” feature of the new german identification card only allows for exchanging a couple of attributes (Gutwirth, 2015). This might entail that SPs keep maintaining individual data silos. Table 3.13 outlines the findings for the local silo model.

Table 3.13: Analysis Results of Local Silo Model

Adequacy	Control	Openness	Protection	Transferability
				

Open Silo Model

Like the local silo model also the open silo model involves a mean under exclusive user control that allows for storing personal data. Rather than being dependent on 1) availability, on 2) online state and on 3) a device users need to carry along, the open silo model focuses on enabling universal access to personal data independently from a user’s individual conditions and approvals, yet in line with pre-defined needs towards privacy. Therefore, the open silo model necessitates that identity owners store their personal data at public domains they either (preferably) own or at least trust. Characteristic for all user-centric models, also the open silo model fosters distribution of personal data per user or per (relatively small) group of users with the consequence of reducing the impact of successful attacks. Although such impact reduction does not help affected individuals, it relieves the vast number of users which would be affected through data disclosure when relying on a SP-centric model instead.

Figure 3.7 illustrates this user-centric model, where entity e can store personal data in a dedicated silo within a trusted or owned domain represented by SR^* . There, IdP^* enables creating new identities and managing existing ones. When accessing a SP, entity e decides on employing one of those

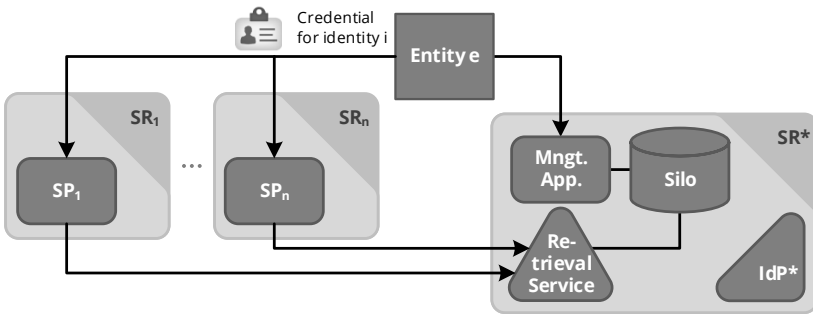


Figure 3.7: Open Silo Model

identities, e.g., identity i . While a web service associated with IdP* allows third parties for retrieving data related to identity i in a controlled way, a web application assists entity e in managing personal data.

WebID (Sambra et al., 2014) is a prime example of the open silo model devised by the World Wide Web Consortium (W3C). Representing a distributed IdM approach, WebID enables users to globally authenticate themselves, connect to each other and manage their identity data at self-defined places. Therefore, it uses three interrelated artifacts: URIs to identify arbitrary entities, profiles to semantically store personal data of identity owners, and certificates to enable ownership-based authentication. Here, WebID URIs also link to WebID profiles, which are resources that store personal data in a flexible, extensible and machine-readable way using Resource Description Framework (RDF)-based vocabularies like Friend of a Friend (FOAF) (Brickley and Miller, 2014; Sambra et al., 2014). It is a crucial design aspect that such personal data silos are owned by individual identity owners or entities they trust. Instead of authorities, WebID empowers identity owners to maintain control of their data. Forming relationships between WebID profiles through WebID URIs facilitates establishing the basis for creating distributed OSNs, similar to alternatives like Diaspora (Bielenberg et al., 2012; Jahid






et al., 2012; Tramp, Frischmuth, et al., 2012). Furthermore, WebID certificates are X.509v3-compliant, have no static expiration time as in BrowserID and contain WebID URIs referring to WebID profiles that also store public keys with the corresponding private keys being verified during the Transport Layer Security (TLS) handshake (Cooper et al., 2008; Dierks and Rescorla, 2008; Inkster et al., 2014). By signing certificates appropriately, WebID enables both self-asserted identities and third-party asserted identities.

Even though it would not be an implementation effort to enable SSO by automatically selecting a favorite identity in form of a client certificate, WebID does not offer SSO by default and, thus, it allows for anonymity if desired (Hackett and Hawkey, 2012). That is, it requires explicit user consent before each authentication attempt, which is similar to BrowserID (Hackett and Hawkey, 2012). Unlike PKIs as an example of the common domain model, WebID allows for implementing the web of trust (WoT) concept, which represents a flat hierarchy only relying on individuals (Caronni, 2000). The open silo model strengthens responsibility of users for their personal data. With personal data being available as Linked Data, i.e., accessible via URI and retrievable using established standards like SPARQL Protocol and RDF Query Language (SPARQL) (Harris and Seaborne, 2013), a controlled exploitation of profile data remains possible for SPs, e.g., to improve customer services or user experiences through customization specific to authenticated users (Wild, Chudnovskyy, et al., 2013a). Moreover, WebID does not burden users with creating and remembering strong passwords, but it necessitates that web browsers support X.509v3 client certificates. By reusing the certificate selection dialog of web browsers, WebID does not need a separate client for identity selection, as in case of CardSpace (Dhamija and Dusseault, 2008).

Notwithstanding that the open silo model sufficiently fulfills all collective requirements, as summarized in Table 3.14, it bears risks originating from

the individual yet consolidated storage of personal data. For instance, outages or successful attacks, e.g., tampering of personal data, would impair IdM at once for all RPs an identity owner is associated with. Therefore, the model requires that users either have enough expertise to protect the systems used for storing their personal data or are supported by appropriate tools respectively. With regard to applying fined-grained access control and detecting misuse or tampering of profile data, WebID shares similar deficits with other IdMSs like OpenID or BrowserID (Hackett and Hawkey, 2012). However, the open silo model and, consequently, WebID facilitates flexibly extending profile data, e.g., add cryptographic signatures. Such extension is not applicable to many other IdMSs due to their centralized, closed, remote or restricted handling of personal data.

Table 3.14: Analysis Results of Open Silo Model

Adequacy	Control	Openness	Protection	Transferability
				

Summary of Identity Management Systems

The analysis of IdMSs shows that user-centric models are superior to SP-centric models especially with regard to control, openness and protection. This is partly due to the fact that users explicitly have to show more responsibility for their personal data. While SP-centric models score in adequacy, both groups of IdMSs are coequal in terms of transferability. The federated SSO model sufficiently fulfills the collective requirements and ranks on top of the group of SP-centric models, whereas the open silo model is the user-centric model with the overall best results. None of the analyzed models fulfills all requirements completely, yet the open silo model, with WebID as a prime example, bears the greatest potential for enhancing control and protection by appropriate extensions made possible through its

open approach. Before proceeding with investigating IdDLs in the next subsection, Table 3.15 provides an overview of the analysis results.

Table 3.15: Analysis Results of Identity Management Systems in Terms of Collective Requirements

IdMS Model	Collective Requirements				
	Adequacy	Ctrl	Openness	Protection	Transfer.
Remote Silo	◐	◐	○	◐	○
Common Dom.	◐	◐	○	◐	◐
Centralized SSO	◐	◐	◐	◐	◐
Multi-Dom. SSO	◐	◐	◐	◐	◐
Federated SSO	◐	◐	◐	◐	◐
Omitted Silo	◐	●	◐	◐	◐
Local Silo	◐	●	◐	●	◐
Open Silo	◐	◐	●	◐	◐

3.3.3 Identity Description Languages

To discuss the suitability of IdDLs, we employ a four-element grouping consisting of proprietary, data interchange, markup and semantic languages. Relying on criteria defined in Subsection 3.2.2, we characterize, assess and illustrate each language type by examples starting with proprietary languages.






Proprietary Languages

While not limited, diverse proprietary languages for describing identities are prevalent in connection with the remote silo model and the common domain model (cf. Subsection 3.3.2). This is especially due to the fact that systems implemented on the basis of these models were developed in-house and evolved independently from each other, i.e., without hav-

ing a mutually agreed language specification for identity data in place. Since SPs treated IdM as an afterthought yet as a necessary mean for the actual service offering (cf. Section 1.2), e.g., order processing in online shopping applications like Amazon, it is not unusual that languages and functions related to IdM were designed for individual SP-centric use cases. In consequence of SPs employing own DSLs, accuracy, appropriateness and assurance capabilities of associated identity data fit only their specific needs instead of the needs of the users. Together with storing identity descriptions in internal databases rather than making them universally accessible through web-based application programming interfaces (APIs), this impairs portability of personal data to third parties.

Although there are efforts for making structured data on identities accessible or exportable, like in case of (Facebook, 2015) accomplishing a federated SSO model with Facebook Login (cf. Subsection 3.3.2), they lack thorough implementation. It is obvious that the absence of semantics and schema specifications for languages used to describe identities affects interpretability, particularity for non-human entities. In contradiction to the sophisticated data storage and processing technologies utilized by SPs internally, they provide users with simplified identity descriptions, e.g., plain Hyper-Text Markup Language (HTML) contents in case of Facebook exports (Berjon et al., 2014). Here, these contents are designed for human entities, which is furthermore affirmed by the fact that only identity owners can manually trigger most export features. Despite such ambitions, proprietary languages do only insufficiently meet the collective requirements. Table 3.16 summarizes the results for proprietary identity description languages using the symbols of the rating system established at the beginning of this chapter.

Table 3.16: Analysis Results of Proprietary Languages

Accuracy	Appropriateness	Assurance	Interpretability	Portability
				

Data Interchange Languages

To tackle the problems of proprietary languages especially with respect to their restricted appropriateness, interpretability and portability, data interchange languages offer both domain-specific and general-purpose alternatives in form of DSLs and GPLs. Although both types differ in the scope of covered use cases, they aim at fostering exchange of data among various providers. By relying on common languages for describing data, including personal data on identities, SPs can benefit from standardized tool sets made available with such languages for data structuring and processing, e.g., through schema support or query processors.

Prime examples of data interchange languages include vCard (Perreault, 2011a) and JSON (Bray, 2014). While vCard represents a DSL for describing personal data of individuals, JSON is a GPL capable of expressing miscellaneous data structures like arrays and dictionaries. The two languages are widely supported standards and have their own Multipurpose Internet Mail Extensions (MIME) types. Like all GPLs, also the common applicability of JSON comes at the cost of a slightly increased overhead in expression and processing in relation to a DSL, when only taking separate domains into account. However, the overhead of JSON is smaller compared to markup languages, e.g., by shorter delimiters (Bray, 2014). Additionally, JSON is extensible, simple to process by human and non-human entities, and compatible to JavaScript with only a few exceptions (Holm, 2015). There are also efforts for standardizing schema definitions through JSON Schema (Galiegue et al., 2013) and query languages, e.g., JSONPath (Goessner, 2007).

On the downside, semantics and extent of DSLs like vCard are pre-defined by a domain, whereas GPLs like plain JSON are applicable to multiple domains with extensible attribute sets, but the latter lacks—in case of JSON—inherent semantics and capabilities for intrinsic and extrinsic object references¹⁴. Here, the meaning of structural elements results from their relative position in the data set rather than from an explicit labeling. Finally, data interchange languages allow for specifying assurances, yet their interpretation highly depends on external knowledge and specialized processors. Table 3.17 outlines these analysis results.

Table 3.17: Analysis Results of Data Interchange Languages

Accuracy	Appropriateness	Assurance	Interpretability	Portability
●	●	●	●	●

Markup Languages

Extending plain data interchange languages by annotations, markup languages indicate structural elements in data sets using explicit labels. So they do not only enable to distinguish structure from text, but also permit human entities for extracting hints on the meaning of thus annotated data.






A prime example of a markup language is XML (Bray et al., 2008), which relies on a hierarchical model for organizing data. Being a human- and machine-readable GPL, XML focuses on support for various applications, easy creation and usability over the Internet (Bray et al., 2008). As XML, however, permits representing the same information in different ways, there is a strong need for schemas that specify the recommended structure. XML schema definition (XSD) (Fallside and Walmsley, 2014) complies

¹⁴Even though URIs within JSON data can link to external resources, a default language processor is incapable of handling them without further instructions, e.g., whether to merge or add data.

with this need, whereas query support is available through technologies such as XQuery (Boag et al., 2010) and XPath (Clark and DeRose, 1999). Furthermore, there are XML representations for describing identities that denote human entities, like xCard for vCard (Perreault, 2011b). On the other hand, XML derivatives like the Web Application Description Language (WADL) (Hadley, 2009) or the Web Service Description Language (WSDL) (Chinnici et al., 2007) facilitate describing identities of non-human entities in case of web applications and web services.

Compared to general-purpose data interchange languages like JSON, XML is not that concise and reasonably clear as it is intended to be (Bray, 2014; Bray et al., 2008). Moreover, (Tauberer, 2014) notes that XML is “not particularly suited for distributed, extensible information unless that XML looks a lot like RDF”. Although markup languages enable human entities for obtaining semantics from labels, this is not the case for non-human entities. Similar to data interchange languages, this has negative implications on accuracy, appropriateness, assurance and portability, as summarized by Table 3.18.

Table 3.18: Analysis Results of Markup Languages

Accuracy	Appropriateness	Assurance	Interpretability	Portability
				

Semantic Languages

As an effort to improve interpretation of descriptions by non-human entities without relying on extensive prior knowledge, semantic languages aim for associating data with further information and, thus, enable to determine underlying concepts and infer meaning. By enriching, detailing and relating data to other data, it is possible to largely increase accuracy,

appropriateness and portability of descriptions, yet this also depends on availability and quality of associated data.

Prime examples for semantic languages include vocabularies created upon RDF (Schreiber and Raimond, 2014). While originally intended for specifying metadata in the context of markup languages like XML, RDF-based vocabularies can also be employed independently as decentralized databases. RDF enables to make statements about data using triple-logic involving subject, predicate and object, where subjects can refer to multiple objects through predicates and objects can also act as valid subjects. That is, RDF implements a flexible, extensible and distributable data model through a graph, which uses URIs as identifiers and as links to related concepts. Schema definition languages such as RDF schema (RDFS) (Brickley et al., 2014) and Web Ontology Language (OWL) (Bechhofer et al., 2004) facilitate specifying RDF-based vocabularies, whereas query operations are supported through languages such as SPARQL (Harris and Seaborne, 2013). In addition to serialize RDF-based statements completely as markup or data interchange languages, like JSON for Linked Data (JSON-LD) (Sporny et al., 2014), other approaches allow for embedding RDF triples within these languages. These approaches include RDF in attributes (RDFa) to “[...] augment the visual information on the Web with machine-readable hints” (Herman et al., 2015), HTML Microdata (Hickson, 2013) and domain-specific, vocabulary-providing microformats like hCard (Çelik and Suda, 2013). Here, hCard fully represents vCard (Perreault, 2011a) in HTML with semantic annotations. To close the circle, technologies like Gleaning Resource Descriptions from Dialects of Languages (GRDDL) (Connolly, 2007) enable to recreate purified RDF graphs by extracting (relevant) RDF triples from documents they were embedded in.

For describing different types of entities, there are various domain-specific yet frequently interlinked semantic vocabularies available, which contribute

to increase the “collection of interrelated datasets on the Web”, known as Linked Data (W3C, 2015a). The Dublin Core (DC) (DCMI, 2012) ontology, the FOAF vocabulary (Brickley and Miller, 2014), the contact ontology (Berners-Lee, 2001) or the vCard ontology (Iannella and McKinney, 2014) are examples that enable to describe aspects of people and organizations in an open, standardized and machine-readable notation. Semantic vocabularies for web applications and web services, however, usually focus on one particular type of service aspect. While OWL-S, SAWSDL or WSMO are designed for describing web services that involve the Simple Object Access Protocol (SOAP), SA-REST and ROSM are tailored for services that implement the REpresentational State Transfer (REST) architectural style (Fielding, 2000; Lee and Kim, 2010). Although their service aspect specifications vary in precision and scope, they share a unified type of description and universal identification and, hence, provide advantages originating from their RDF heritage including flexibility and ease of interpretation (Wild and Gaedke, 2014). Supporting the specification of both RESTful and SOAP-based web services in terms of message types, interfaces, bindings and end points, the WSDL 2.0 RDF mapping is an example of a semantic description language offering sufficient precision and scope (Kopecký, 2007).

To sum it up, semantic languages do almost completely fulfill the collective requirements, as shown in Table 3.19. Assurances are supported by RDF-based vocabularies through permitting variable extension and distribution of related data via linking, which is an intrinsic part of the framework. Nevertheless, assurances have to be made and processed individually because of the domain-specific focus of RDF vocabularies.

Table 3.19: Analysis Results of Semantic Languages

Accuracy	Appropriateness	Assurance	Interpretability	Portability
●	●	◐	●	●

Summary of Identity Description Languages

When exchanging and processing identity descriptions among diverse providers is a matter of concern, proprietary languages are not preferable. Data interchange and markup languages partially address shortcomings of proprietary languages through offering improved portability, interpretability and appropriateness, but they lack in terms of linking and interpreting data without having prior knowledge or pre-defined routines available. Semantic languages cover these aspects well. Moreover, the analysis results for semantic languages indicate that they are not only well-suited for describing people and machines, but also have the potential to address the interoperability issues detected in web engineering models (cf. 3.3.1). Before concluding the chapter in the next section, Table 3.20 summarizes the analysis results using the criteria specified in Section 3.2.

Table 3.20: Analysis Results of Identity Description Languages in Terms of Collective Requirements

IdDL	Collective Requirements				
	Accuracy	Appropriate.	Assurance	Interpret.	Port.
Proprietary	◐	○	◐	○	◐
Data Interch.	◑	◑	◑	◐	◑
Markup	◑	◑	◑	◑	◑
Semantic	●	●	◑	●	●

3.4 Summary

By investigating approaches for web engineering, systems for IdM and languages for describing identities, we retrieved an overview of technologies related not only to management of personal data, but also to control and protection of such data.

Focusing on reusability of building blocks, CBWE approaches offer strong support for evolving web systems in relation to approaches orientating on other aspects. Here, especially WebComposition in association with WAM satisfies the collective requirements by considering the life cycle of web systems and by providing conciseness, review and simplicity when designing and making architectural changes to web systems. WAM enables to describe composition of federated web systems and involved components as well as their relations at a high level of abstraction. It is easy to apply during design using a graphical notation and allows for automatic model verification during system evolution (A. Heil, 2012). However, it lacks capabilities for consistently describing and detailing semantics of involved entities, which would assist in modeling web systems at different granularity levels and improve interoperability and utilization of models through machines.

With regard to IdM, the open silo model is superior to alternative models through fostering control, ownership, responsibility and self-determination while ensuring accessibility to personal data. WebID is a promising representative of a decentralized IdMS implementing this model. It 1) empowers individual entities rather than authorities, 2) supports domain-independent authentication involving certificates that offer high cryptographic strength, 3) utilizes flexible, extensible and machine-interpretable identity descriptions, and 4) provides application-independent identification and linkability of identities. Here, identities are not restricted to represent characteristics of human entities only. Moreover, WebID facilitates consolidating personal data and, thus, eases controlling access and privacy with the option of both local and global revocation of identities or identity proofs respectively.

Easing interpretation and inference by machines, semantic languages are well-suited for conceptual modeling and domain-specific descriptions of identities in an accurate, expressive and portable way. RDF-based vocabularies are application-neutral, standard-based and

web-compliant. They enable to create adaptable, distributed, extensible, interlinked and reusable descriptions of heterogeneous facts, which human and non-human entities can then integrate, process or utilize in a dynamic, scalable and seamless manner.

Now that technologies suitable for meeting the challenges outlined in Chapter 2 have been detected, the next chapter proceeds with describing how we make use of them as part of the proposed solution.

Enhanced Security in Managing Personal Data

To holistically address the challenges outlined in Chapter 2 with respect to the suitability of technologies analyzed in Chapter 3, this chapter describes an approach to enhance security in managing personal data by web systems. Section 4.1 outlines the design of the proposed solution. Using the design, Section 4.2 then specifies the solution architecture and process to enable reusability and show possible integration points. Section 4.3 introduces three key components that extend the fundamental solution architecture in order to meet the remaining challenges. Not only to manifest the proposed solution architecture and process, but also to establish a basis for integrating the key components, Section 4.4 describes the proof-of-concept platform. Summarizing our proposal for solving the central problem stated in Section 1.2, Section 4.5 concludes this chapter.

4.1 Design

For designing the solution, we involve three reusable artifacts that are building upon each other. Each artifact represents a certain state of the design through a particular model, i.e., conceptual model, logical model, and physical model. Taking into account the principles of web engineering and WebComposition in particular (cf. Subsection 3.3.1), we facilitate both adaptability and reusability through postponing the technical implementation until a sound security foundation has been established. As a starting point, Subsection 4.1.1 describes the conceptual model. To accomplish a common understanding of the matter, this model only denotes significant entities as well as the relationships between them, and creates a generalized formal structure. Subsection 4.1.2 then outlines the logical model, which details the conceptual model by putting the concepts into context, yet without considering the physical representation. Finally, Subsection 4.1.3 specifies the physical model, which defines the basis for the technical implementation of the logical model.

4.1.1 Conceptual Model

In order to model the concepts relevant for enhancing security in managing personal data by web systems, we have to properly take account of all entities involved, i.e., particularly persons, web applications and web services. For consistently defining these entities, we make use of semantic vocabularies as they enable to apply the same metamodel by RDF and, thus, offer advantages with regard to universal linkage, discovery, accessibility and arbitrary detailing data (cf. Subsection 3.3.3). Therefore, we distinguish between entity classes, entities and identities. While entity classes define the general concepts of entities, identities characterize specific aspects of entities within defined contexts (cf. Subsection 2.1.2 on “Terms and Def-

initions”). For instance, Alice is an entity of entity class person and has a co-worker identity representing her in a business context.

As an effort of making architectural descriptions of web systems machine-readable and linkable, we use the WAM ontology proposed and extended in (A. Heil, 2012; Wild and Gaedke, 2014). Modeled using OWL, the WAM ontology does not only define the classes of web entities, legacy entities, and security realms, but also their associations. To reflect the organizational boundaries of control, both web and legacy entities can be contained within security realms. The subclasses identity (service) provider, application, and service are inherited from the web entity class. They might invoke other web entities and maintain relationships to legacy entities, which are responsible for tasks like storage or processing. Figure 4.1 illustrates the WAM ontology.

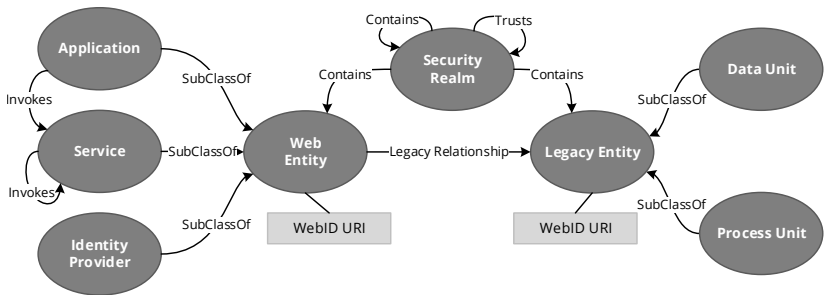


Figure 4.1: WAM Ontology

Having extended WAM (Meinecke and Gaedke, 2005) towards a semantically enriched architecture model for web systems, we also have to consider typical SOA implementations with SOAP and RESTful services. Similar to above model definition, we detail the concepts present in such architectures using domain-specific semantic vocabularies. A descriptive approach using Linked Data facilitates systematic use by authorized services in later phases and also assists in controlling the evolution of web systems as proposed in (Wild and Gaedke, 2009). In addition to describing web applications

and web services, this allows denoting the people involved as suggested in (Maamar et al., 2011). For modeling identities that denote entities of the class of web services, we rely on the vocabulary introduced with WSDL 2.0 RDF mapping (Kopecký, 2007). Furthermore, we make use of existing vocabularies such as FOAF or the contact ontology to characterize identities that refer to entities of the class of persons. Although other ontologies enable to model the identities of the remaining entity classes referenced in the WAM ontology, these classes are of less relevance for this dissertation and, therefore, left out of consideration.

To clearly identify relevant representations of entities in an architecture description of a web system, we need to detail essential concepts of identity management first. For IdM we rely on WebID (Sambra et al., 2014). Not only does WebID allow for identifying entities, but it also enables authentication and facilitates the creation of precise, extensible, interpretable, linkable and portable descriptions (cf. Subsection 3.3.2). WebID is a distributed identification approach that involves three underlying artifacts for various purposes, including recognition, characterization and legitimization. These artifacts are the WebID URI, the WebID profile and the WebID certificate. The following formalism of WebID extends the definitions of Subsection 2.1.2.

A **WebID URI** refers to an identity i that represents entity e . Like a username in other IdMS, a WebID URI $w \in W \subset U$ is a URI denoting an identity i , where W is the set of all WebID URIs and U is the set of all URIs. Dereferencing a WebID URI w returns a set of RDF triples $T \subset \mathfrak{T}$ that describes personal attributes of identity i using Linked Data. While \mathfrak{T} is the set of all RDF triples, each triple $t \in \mathfrak{T}$ consists of subject t_1 , predicate t_2 , and object t_3 . Equation (4.1) formalizes the dereferencing through function $\alpha(u)$, which yields T for URI u being a valid WebID URI.

$$W = \{u | \alpha(u) = T\}, u \in U \quad (4.1)$$

A **WebID profile** is a URI addressable resource. It is available at WebID URI w and contains a set of RDF triples T describing identity i . As RDF is used for specifying all personal data, an identity's attributes are expressive, extensible and machine-readable (Schreiber and Raimond, 2014). This is a major advantage to other IdMSs, which are restricted in assigning and exchanging user attributes. Such semantic descriptions facilitate controlled large scale exploitation of profile data to optimize customer services and improve the user experience (Wild, Chudnovskyy, et al., 2013a). As RDF triples T span graph $G = (V, L)$, $G \in \mathfrak{G}$, where \mathfrak{G} is the set of all graphs, and graph G refers to a set of triples describing identity i , we formalize this equivalence in Equation (4.2).

$$\begin{aligned} T \sim G &\Leftrightarrow \forall t = (t_1, t_2, t_3) \in T : \\ t_1, t_2, t_3 &\in V \wedge (t_1 t_2) \in L \wedge (t_2, t_3) \in L \end{aligned} \quad (4.2)$$

In addition to being a semantic repository for personal data of an identity i , a WebID profile also contains a set of public keys described by triples $T_K \subset T$. Each single public key $k' \in K$ is described by triples $T_{k'} \subseteq T_K$, where $K \subset \mathfrak{K}$ is the set of asymmetric keys owned by identity i and \mathfrak{K} the set of all asymmetric keys. $T_{k'}$ specifies diverse attributes of a public key k' , including type, modulus and exponent. A k' -corresponding private key $k'^{-1} \in K$ allows for proving that an identity i actually owns public key k' . Equation (4.3) defines the relation between k' and k'^{-1} by means of function $\beta(k, m)$, which maps messages M and the set of keys \mathfrak{K} on the set of messages, where key $k \in \mathfrak{K}$ and message $m \in M$.

$$\beta : \mathfrak{K} \times M \rightarrow M \quad \beta(k', \beta(k'^{-1}, m)) = m \quad \forall m \in M \quad (4.3)$$

A **WebID certificate** $X_{k'} \in X \subset \mathfrak{X}$ contains WebID URI w and public key k' of an identity i , where X is the set of WebID certificates associated with an identity i and \mathfrak{X} is the set of all WebID certificates. This is formalized

by Equation (4.4). WebID certificate $X_{k'}$ is signed with the corresponding private key k'^{-1} or the private key of a trusted party.

$$X_{k'} = (w, k') \quad (4.4)$$

An identity $i = (w, T)$ is described by WebID URI w and personal data T contained in the associated WebID profile. Unlike knowledge-based authentication approaches using username/password pairs as proof of identity, WebID is an ownership-based authentication approach. For authentication, it relies on public key data available in both WebID profile and certificate. An identity i is authenticatable when i has a WebID certificate $X_{k'}$ containing a public key k' for which i owns the corresponding private key k'^{-1} , as defined in Equation (4.5).

$$i = (w, T) \text{ is authenticatable} \Leftrightarrow \exists k' : T_{k'} \subset T, \exists X_{k'}, \exists k'^{-1} \quad (4.5)$$

The above stage of authentication is performed after the ownership of the private key k'^{-1} is proven during the TLS handshake (Dierks and Rescorla, 2008). Building on the conceptual model, the next subsection continues with discussing the authentication flow as part of the logical design.

4.1.2 Logical Model

Relating the concepts of the prior model, the logical model puts them into context by outlining valid sequences of their usage. The logical model also details the conceptual model, so that we can apply now established capabilities for identification not only to the entity class of person, but also to web and legacy entity classes defined by the WAM ontology. This enables to employ WebID URIs for identification (cf. Figure 4.1) and for referencing open silos of personal data that specify identity attributes. By laying the foundation for entities of arbitrary class to refer to each other

and access associated data, it becomes possible that persons can reference and retrieve data of machines and vice versa. Having set the identification, authentication follows as the next logical step.

In accordance with the WebID-TLS specification (Inkster et al., 2014), Figure 4.2 illustrates the logical authentication flow as a UML sequence diagram. Here, entity e intends to access a resource made available by a web entity. For the sake of simplicity and understandability, let e be a person called Alice that employs a user agent, like a web browser, to perform the request on her behalf. Let furthermore the resource-providing

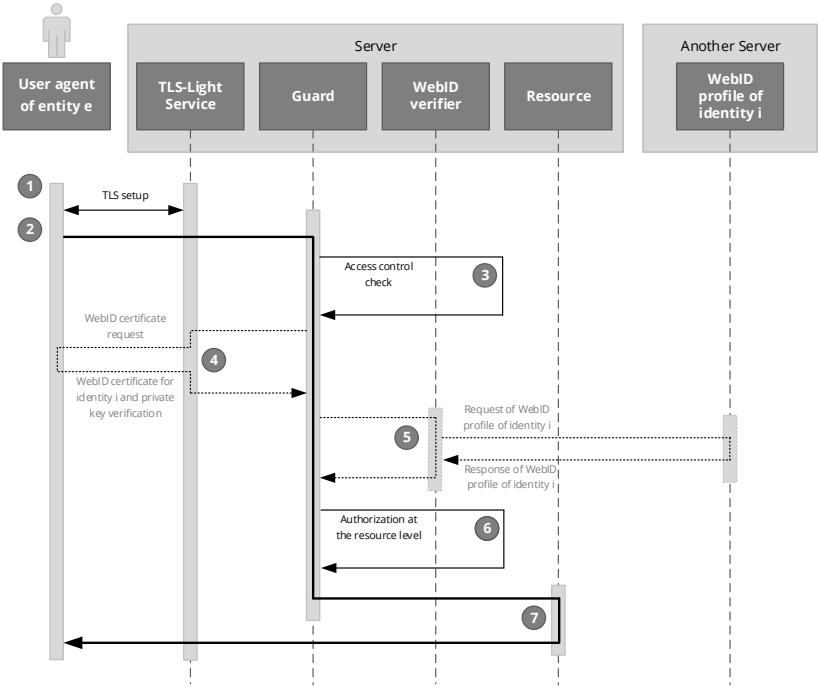


Figure 4.2: Default WebID Authentication Sequence

web entity be a storage server. Acting as a SP, the storage server allows for authenticating users via WebID. To request a particular resource stored on the server, Alice has to initiate a secure connection via her user agent. Having established a TLS-secured connection in ①, a guard shielding the storage server directly intercepts Alice's actual request sent in ②. The guard parses the request to match the detected target and existing access control settings in ③. If the requested resource is access controlled, the guard asks Alice for authenticating herself by providing a credential in form of a WebID certificate in ④. Given that principal Alice selected an identity i through WebID certificate $X_{k'}$ for which she owns the corresponding private key k'^{-1} , the public key k' of the certificate is compared to a valid one found in Alice's WebID profile (cf. Equation (4.5)). That is, this valid public key described by $T_{k'}$ has to be out of T , which in turn represents all personal data stored in the profile associated with her identity i . The WebID verifier, being responsible for this check, therefore automatically retrieves Alice's WebID profile. This is done by dereferencing WebID URI w stored in WebID certificate $X_{k'}$, which Alice has been provided. Here, not the SP, but another server hosts Alice's WebID profile. It is important to note that Alice either owns or sufficiently trusts this profile-hosting server in its role as IdP. Assuming that both public keys are identified as equal in ⑤, Alice is potentially granted permission to the requested resource in ⑥, which she eventually accesses in ⑦.

On closer examination of above default authentication flow, we can make a number of observations relevant for the further course. While a person represents the requesting entity in this case, e could also have been a non-human entity, like another web service, without changing anything on the sequence shown in Figure 4.2. With an IdP offering a WebID profile that includes both personal data and essential means for a successful authentication, it is necessary that SPs not only can access the WebID profile on such system, but also trust data described therein. On the side of the

requesting entity, a user agent has to assist in selecting a WebID certificate. During authentication, the principal does not have an opportunity to review and approve that statements inside the WebID profile are correct and as intended by the identity owner. Individual circumstances of the SP, the IdP and most importantly the requesting entity are not taken account of. Moreover, any SP retrieving the WebID profile obtains the identical view on personal data independently from its own identity. Even though identity owners could adjust access to personal data by creating multiple identities with varying data sets stored inside the corresponding WebID profiles, keeping personal data among different WebID profiles in sync would be an additional maintenance effort. By extending the IdP by means for authentication similar to those of the SP, it would be possible to distinguish between different sources that attempt to retrieve profiles.

Now that we have explained the logical modeling of the underlying concepts, the following subsection adds further details in order to create the physical model.

4.1.3 Physical Model

With the concepts and logic defined in the previous subsections, the physical model builds on them to facilitate the technical implementation as a subsequent step.

Figure 4.3 shows the physical model of the WebID artifacts. Here, WebID certificate $X_{k'}$ is a common X.509 client certificate as per (Cooper et al., 2008). It contains only non-essential personal data like the *subject* statement, which however can assist human entities in locating the right certificate. More importantly, $X_{k'}$ stores WebID URI w in its Subject Alternative Name (SAN) property in (cf. **A**) in Fig. 4.3) as well as public key k' by specifying algorithm, modulus and exponent (cf. **B**). Dereferencing w returns a specific

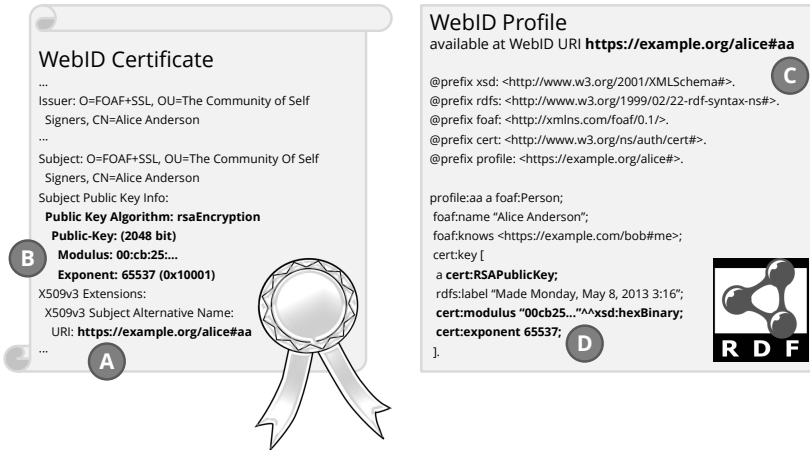


Figure 4.3: Artifacts in WebID: Certificate, URI and Profile

representation of a WebID profile made available at that location (cf. ©). Valid representations include JSON-LD, Notation3 (N3) (Berners-Lee and Connolly, 2011), RDF/XML (Gandon and Schreiber, 2014) or Turtle (Beckett et al., 2014), where compliant accessors must stringently support the latter. RDF triples T stored in the WebID profile are depicted in Turtle syntax. In addition to applying common semantic vocabularies or adequate extensions like RDF, RDFS and XSD, RDF triples $T_{k'}$ specify a k' -corresponding public key (cf. Ⓓ) using the cert ontology (Story, 2008). While RDF enables employing various ontologies, it is apparent that T describes personal data associated with an identity of a human entity via appropriate attributes using the FOAF vocabulary (Brickley and Miller, 2014).

In addition to identity descriptions of human entities (cf. Figure 4.3), WebID profiles allow for describing identities of arbitrary entities without altering process and technologies involved for identification, authentication,

attribute access or retrieval. As an example, Listing 4.1 shows the WebID profile associated with a WAM-compliant web entity.

The listing describes an accounting web service by semantically specifying diverse characteristics including the public key k' used for authen-

Listing 4.1: WebID Profile of a Web Service (non-essential RDF triples are skipped)

```
1 @prefix wsdl: <http://www.w3.org/ns/wsdl-rdf#> .
2 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
3 @prefix cert: <http://www.w3.org/ns/auth/cert#> .
4 @prefix owl: <http://www.w3.org/2002/07/owl#> .
5 @prefix aSrv: <https://example.org/AccountingService#>.
6
7 aSrv:me a foaf:Agent;
8     foaf:name "Accounting Service";
9     foaf:maker <https://example.org/BobBuilder#me>;
10    owl:sameAs aSrv:wsdl .
11    cert:key [ a cert:RSAPublicKey;
12              cert:modulus "a45dle..."; cert:exponent 65537; ];
13    foaf:knows <https://example.org/DatabaseService#me> ;
14    foaf:knows <https://example.org/ReportingService#me>;
15
16 aSrv:wsdl a wsdl:Description;
17     wsdl:interface aSrv:interface;
18     wsdl:binding aSrv:bindingHTTP, aSrv:bindingSOAP;
19     wsdl:service aSrv:service .
20
21 [...]
22
23 aSrv:service a wsdl:Service;
24     wsdl:endpoint aSrv:endpointHTTP, aSrv:endpointSOAP;
25     wsdl:implements aSrv:interface; [...] .
26
27 aSrv:endpointHTTP a wsdl:Endpoint;
28     wsdl:address <https://accounting.ex.org/rest/>;
29     wsdl:usesBinding aSrv:bindingHTTP; [...] .
30
31 aSrv:endpointSOAP a wsdl:Endpoint;
32     wsdl:address <https://accounting.ex.org/soap/>;
33     wsdl:usesBinding aSrv:bindingSOAP; [...] .
```

tication (cf. lines 11 and 12), associated services identified via WebID URIs (cf. lines 13 and 14), and endpoints to access resources provided by the web service (cf. lines 23 to 33).

With defining artifacts at this level of detail, we can determine the use of Turtle for representing semantic descriptions of personal data, SPARQL for queries and OWL for schema definitions. That is, to transform the physical model into executable source code, these technologies need to be supported by suitable programming languages and libraries. After completing the physical state, the next section reports on the integration of the design artifacts into the solution architecture and the associated process.

4.2 Architecture and Process

For assembling the design artifacts described in the previous section, we factor in the various objectives different stakeholders have in terms of a satisfying solution. The findings retrieved by investigating the challenges in Chapter 2 suggest that an enhanced security in managing personal data necessitates to take account of relevant entities present in web-based architectures. Relevant entities include both the components, like web applications or web services, and the users that interact with these entities, i.e., persons or other components, through certain interfaces either explicitly or implicitly.

By making use of light-weight models, we intend to deliver Result 1: “Improved Modeling of Security Aspects for Web Systems” by enabling an integral perspective on security for both human and non-human entities and improve diverse attributes at design time and runtime. Our proposition builds upon WebComposition as a CBWE approach and WAM as an architectural modeling method that puts importance on security. To assist in modeling web systems at different granularity levels, improve interoperability and in-

crease the utilization of models through machines, we implement a holistic approach to consistently identify and semantically describe relevant entities.

In line with the vision of utilizing semantics in a broader and integrative context (Papazoglou et al., 2007), the proposed solution involves identification and semantic description for persons, individual components and compositions resulting from CBWE efforts. Through providing also non-human entities with identities, we facilitate the application of eligible building blocks when engineering web systems. Here, WAM helps engineers to model architectures of web systems with respect to security matters, where the proposed solution allows for universally identifying and semantically describing included components, like web services, in an expressive manner (Wild and Gaedke, 2014). In addition to describe components, it also enables to specify their relationships through trust bonds and invocations as well as their affiliations through security realms within the architecture (cf. Activity 1.1: “Extend Means for Modeling Secure Web Systems”). To this end, we provide appropriate support by tools, as described in the remainder of this chapter. Resulting architecture models of web systems are made retrievable via URI-accessible, semantic descriptions as per WAM ontology (cf. Figure 4.1) through relying on RDF.

Rather than realizing a push model, where users entrust SPs with management and storage of their data, we propose an architecture that forces a pull model, where interested parties can actively obtain personal data from the users’ individual repositories according to their conditions. In concert with delivering Result 2: “Reduced Need for Accumulation of Personal Data by Third Parties”, we nurture a controlled user-centric integration of personal data by implementing the open silo model (cf. Subsection 3.3.2), where identity management and storage are decoupled from usage of identities and associated data. Our proposition hereby regards the objectives of individual persons, companies and governments. It also considers essential

elements of identity management, i.e., digital identities, storage and integration of personal data and related processes (Dinger and Hartenstein, 2008). Relying on the basis established through WebID, we enable arbitrary entities to holistically manage personal data associated to their identities at self-defined web-accessible locations, which are either under their exclusive control or in control of entities they sufficiently trust. Yet, this also implies that identity owners have to take over responsibility through governing access, quality and protection of (their) personal data.

The proposed architecture involves four inherent elements: an IdP, a personal data repository, a protection service and tool support for identity management. Regardless of whether a system implementing this architecture serves one or many entities, all four architectural elements have to be present in order to allow a self-deterministic identity management with focus on security of personal data. While we refer to these elements individually, they are combinable with each other in order to create consolidated service offerings. Figure 4.4 exemplarily illustrates the proposed architecture using WAM. Here, an IdP (cf. **A**) allows for creating an identity representing an entity in a certain context. Furthermore, a central repository (cf. **B**) semantically stores all data associated with this identity. Along with assisting in accomplishing diverse IdM-related tasks, an identity management application (cf. **C**) enables an entity to express the conditions to which third parties can access personal data owned by this particular entity. A service protecting personal data stored in the repository (cf. **D**) enforces the conditions defined by the entity owning the data. Depending on the trust relationships¹⁵ between an entity making personal data available and potential requesting entities (cf. **E**, **F** and **G**) as well as the legitimization provided by them, the protection service permits or

¹⁵For the sake of simplicity in illustrating the trust relationships in Figure 4.4, we assume that only one identity is employed per security realm.

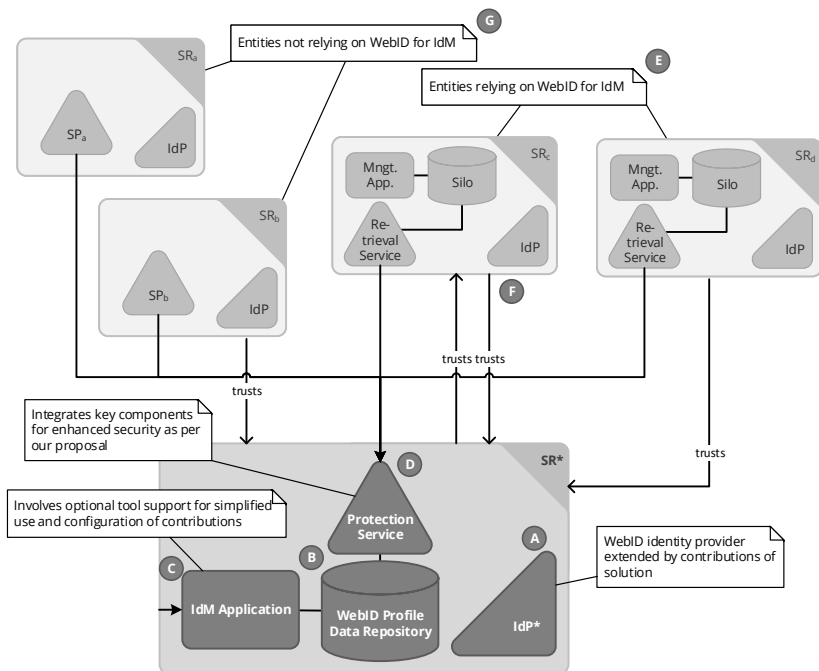


Figure 4.4: Architecture for Self-Deterministic Identity Management

denies access to certain personal data sets. Moreover, the protection service safeguards personal data from several threats, as outlined in Section 4.3.

For fostering self-deterministic IdM on the basis of the open silo model and means for an enhanced security, the proposed solution defines a five-stage life cycle valid for identities of both human and non-human entities. Figure 4.5 schematically illustrates this IdM life cycle, where identity management begins with the stage of initiation followed by provision, moves on with the potentially recurring stage of operation, and ends up with the stage of deprovision before final termination. While combining initiation with provision and deprovision with termination are valid options, treat-

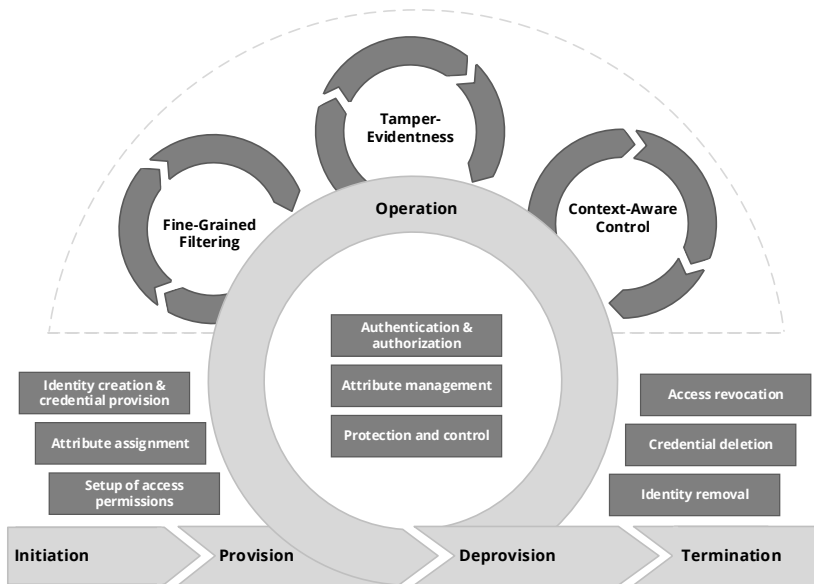


Figure 4.5: IdM Life Cycle as per Proposal

ing these stages separately allows for more flexibility in IdM, as further explained when characterizing each stage in the following:

Initiation Employing an own or a sufficiently trusted IdP, an entity triggers the creation of an identity, which consists of at least a universally unique identifier. The identity owner either manually specifies or automatically generates such identifier.

Provision According to the entity class an identity has been created for, an IdP declares a default set of vocabularies and optionally assigns common attributes to the identity, which have been provided by the initiating entity. That is, an identity of a human entity involves vocabularies and attributes different to those of identities of non-human entities, e.g., a

composition or a web service. Furthermore, an entity can decide upon applying precautions for tampering protection of data associated with an identity (cf. Section 4.3). In line with individual preferences and privacy needs, an entity can optionally determine the parameters of the credential to be issued by an IdP¹⁶ for a created identity, such as the type of signature, cryptographic strength and validity (cf. Section 4.3). Depending on the selected parameters, the proposed solution enables creating self- as well as third-party asserted identities. Finally, the IdP preconfigures the privileges for managing the identity by setting appropriate access control rules.

Operation Contrary to other stages in the IdM life cycle, the stage of operation particularly contains ongoing and recurring activities that necessarily imply provisioned identities. Representing a practicable choice for both human and non-human entities, the proposed solution specifies ownership-based authentication of users. When authenticating to an IdP with an appropriate credential, an identity owner can control various aspects concerning the identity in question. Among other things, these aspects include characterizing an identity semantically using an extensible set of class-related attributes, establishing links to other human or machine identities, and maintaining credentials. To execute operations on behalf of identity owners, IdPs need access to personal data. By setting permissions to personal data in a homogeneous way, identity owners are elevated in the position to control access for requesting entities of arbitrary class. For further enhancing the security in self-deterministic IdM, the proposed solution supports and protects identity owners through context-aware control, tamper-evidentness, fine-grained filtering of personal data, which represent integral yet discrete operations provided by individual components (cf. Section 4.3).

¹⁶The IdP used for initiation may be distinct from the IdP issuing the credential, yet not without causing some disadvantages in user experience and automatic data integration.

Deprovision On behalf of the identity owner, the IdP releases the deprovisioning by disabling the attributes associated with the identity and by revoking access through putting corresponding privileges on hold. Disabling the attributes prevents the identity owner from authentication and, thus, authorization on a global scale, i.e., independent of specific SPs.

Termination With finishing the stage of deprovision, the IdP finally removes the identity and associated attributes so that the identifier is not longer valid. Moreover, the IdP deletes all privileges on hold. In contrast to other IdMS, it is the entity's responsibility to erase the credential associated with the identity, i.e., the digital certificate.

The IdM life cycle applies to relevant entities present in web-based architectures. In comparison to persons and components, however, compositions take a special role. With reference to Figure 4.6, which illustrates the IdM life cycle for compositions by making use of the Business Process Model and Notation (BPMN), we detail this process for compositions in the next two paragraphs.

Even though the stages of initiation and provision for identities of compositions (cf. ① in Fig. 4.6) are quite similar to other classes of entities, the stage of operation involves two tiers of identity management: one for the composition and another one for involved components. When describing the architecture of a web systems using WAM in ②, engineers can not only denote components according to the WAM ontology, but also their connections and associations to security realms (cf. Activity 1.1). After modeling the web system at the composite level, web engineers can discover and integrate eligible components in ③. In case no suitable component is available, an identity specifying the missing component has to be created prior to the actual implementation in ④. Web engineers must therefore initiate the IdM life cycle for each non-existing component (cf. ⑤ and ⑧). Once an identity representing a component is created, engineers can

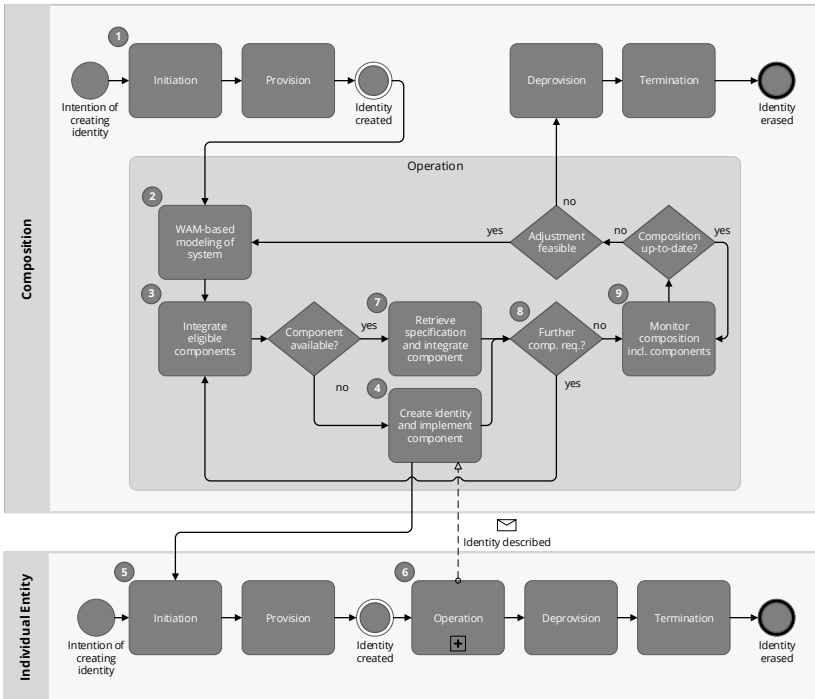


Figure 4.6: IdM Life Cycle for Compositions

describe specifics of the new identity in ⑥ by employing known semantic vocabularies such as the WSDL 2.0 RDF mapping, in case the component is a web service. The proposed solution also enables web engineers to refer to already existing components by their identifiers in order to acquire data associated with their identities, e.g., interface definitions or statistical data (cf. ⑦). It is worth noting that components represented through identities are not bound to particular compositions by design, but can be employed in multiple compositions in accordance with prevailing policies. That is, components, which were used in a composition that became outdated or unnecessary for some reason, might still be applicable in other compositions.

Beyond the stage of initially modeling web systems, web engineers also benefit from the proposed solution after a composition has been deployed. During runtime of a web system, service providers can reflect changes to components, like new endpoints extending an existing interface, in updating associated identity data. By monitoring, discovering and obtaining updated identity data via WebID URIs (⑨ in Fig. 4.6), web engineers and also capable components can learn from propagated changes and potentially adapt to them in a semi-automatic fashion through interpreting semantic component descriptions (Wild and Gaedke, 2014). If adjustments to a web system are feasible, web engineers can reenter the inner process loop with modifying the underlying model (cf. ②).

Moreover, the proposed solution allows entities for authenticating themselves and retrieving—if authorized—runtime parameters of other entities, like the current utilization of a web service or the number of issues emerged within a specific period (Wild and Gaedke, 2009). For authentication according to WebID, entities like web services can employ evidence for a claimed identity through an ownership-based credential issued during provision, i.e., a WebID certificate as explained in Subsection 4.1.1. Holistically enabling semantic description, universal identification and access control for compositions and components, the proposed solution supports web engineers in maintaining and evolving web systems using light-weight models (cf. Result 1: “Improved Modeling of Security Aspects for Web Systems” and Activity 1.1: “Extend Means for Modeling Secure Web Systems”). For example, engineers can detect a busy web service within a composition by means of propagated runtime parameters and find eligible alternatives that feature compatible interfaces while being more responsive. Providing an up-to-date big picture of the architecture, the proposed solution facilitates a systematic, well-directed and controlled maintenance and evolution of web systems (Wild and Gaedke, 2009; Wild and Gaedke, 2014).

While the IdM life cycle mostly relates to functions of IdPs, service providers have to be considered as well during the stage of operation, albeit less comprehensively. An IdP acts as a particular SP that is specialized in identity management. Similar to other IdMSs, SPs must integrate appropriate software modules to support authentication as per WebID. The proposed solution does explicitly not involve that SPs maintain own repositories for storing personal data, but they still can access data associated with an identity, like personal data, under the terms of the corresponding identity owner.

During operation, both IdPs and SPs must therefore ensure that security measures taken by identity owners are in force and obeyed. In order to so by delivering Result 3: “Extended Means for Control and Protection of Personal Data”, the following section introduces three key components for enhancing security.

4.3 Key Components for Enhanced Security

To enhance security in managing personal data by web systems, it is not sufficient to *only* secure web applications and web services that exploit data associated with identities. The systems that offer data have to be protected as well. It is evident from the IdM life cycle (cf. Figure 4.5) and the architecture (cf. Figure 4.4) that we have to particularly protect identity (service) providers, as they make personal data on identities available. The fact that our proposal towards self-deterministic identity management involves centralized data repository in control of identity owners furthermore underlines the absolute necessity to specially focus on protecting IdPs. Nevertheless, the cooperation by SPs is mandatory for implementing a holistic and comprehensive approach towards protection. When identities are

about to be employed by entities, SPs have to confirm that protective measures are in place and taken into account during authentication attempts. In line with delivering Result 3, we consequently dedicate key components to safeguard personal data managed by identity (service) providers from three main types of threats, i.e., improper use, tampering and identity theft, and unwanted retrieval (cf. Activities 3.1 to 3.3 on page 58). As the use of the components is part of the IdM life cycle as per our proposal, we symbolically illustrate the intrinsic processes related to them at the top of Figure 4.5, with detailed explanations to follow in Chapters 5 to 7.

For taking advantage of these key components, we integrate them into the WebID authentication sequence and, thus, extend the original process shown in Figure 4.2. There, entity e , exemplarily named Alice, wanted to retrieve an access-controlled resource and had to authenticate with her WebID certificate before. The extension to this process by our solution is labeled “Security enhancement by proposal” and highlighted bold in Figures 4.7 and 4.8. Here, ① to ⑤ are analogous to Figure 4.2. Our proposal adds ⑥ and ⑦, as shown in Figure 4.7. These two extra steps are responsible for coping with Problem Cause 3.2: “Risk of Identity Theft and Tampering of Personal Data” and Problem Cause 3.1: “Insufficient Control of Identity Based on Individual Context” (cf. pages 45 and 47) through verifying both the integrity of personal data and the delegation rights stored within an identity owner’s WebID profile. Both allow detecting tampering and improper use of personal data corresponding to WebID identity owners. As an example, this assists in discovering malicious requests originating by profile data compromised by aggressors or subjects seemingly acting on the identity owner’s behalf yet not within mutually agreed upon scopes.

On the side of the IdP hosting Alice’s WebID profile, our proposal adds a mechanism to address Problem Cause 3.3: “Incomplete Range and Granularity of Access Control” (cf. page 48) through creating customized views

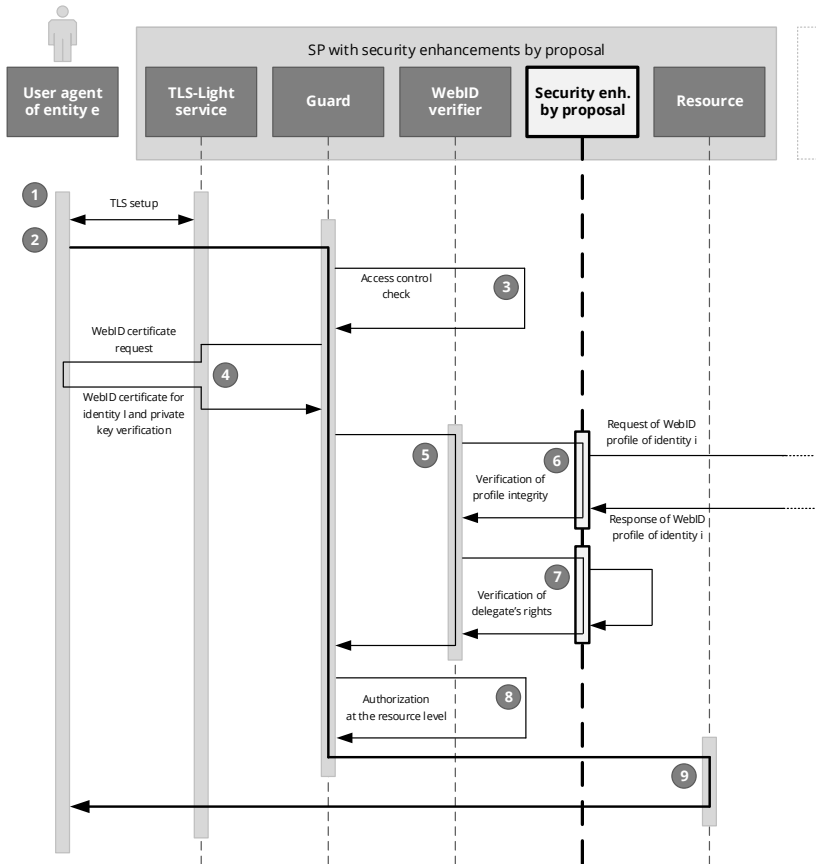


Figure 4.7: Security-Enhanced WebID Authentication Sequence for SPs

that are specific to requesting entities and, thus, avoid unwanted disclosure of Alice’s personal data, as depicted in Figure 4.8.

All three key components are complementary, rely on pre-existing security artifacts and help to increase the protection of personal data (Wild et al., 2015). They are intended to seamlessly fit into the RDF-based semantic

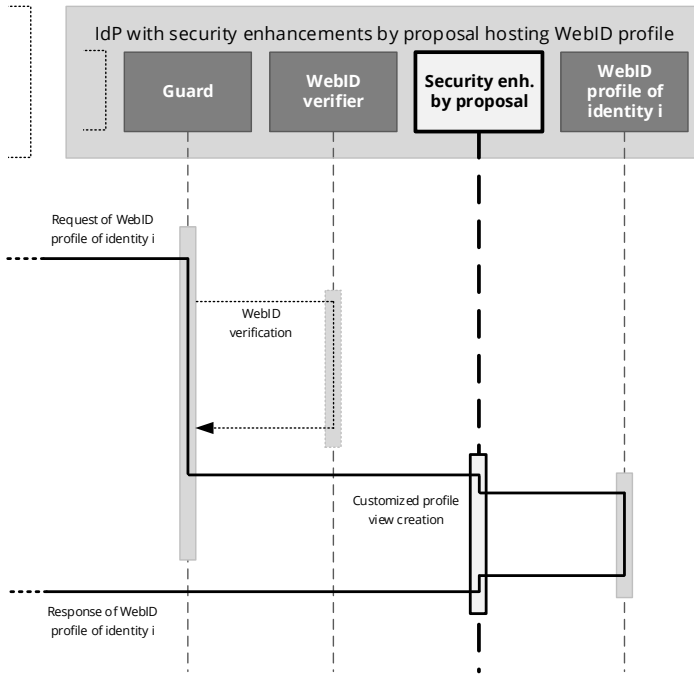


Figure 4.8: Security-Enhanced WebID Authentication Sequence for IdPs

landscape and contribute assuring quality and maintainability by reducing adjustments and extensions to the necessary minimum. Unlike both verification mechanisms that are integrated on the service provider's side, protection against unwanted disclosure is only available on a system hosting an identity owner's WebID profile, i.e., an IdP. Since an IdP hosting a WebID profile serves incoming profile requests as well, the verification mechanisms could be also integrated there in order to check requests initiated by subjects authenticated via WebID.

The next three subsections briefly describe the key components enriching our proposal by measures for increased protection and control. While

Subsection 4.3.1 outlines a way for avoiding improper use of personal data by subjects acting outside their scope yet on behalf of identity owners (cf. ⑦ in Figure 4.7), Subsection 4.3.2 explains how WebID profiles can be secured from malicious manipulation by offering an integrity protection and detection mechanism (cf. ⑥ in Figure 4.7). Subsection 4.3.3 presents the third key component. There, we show our contribution against unwanted retrieval attempts using fine-grained filters.

4.3.1 Context-Aware Control

In order to resolve Problem Cause 3.1: “Insufficient Control of Identity Based on Individual Context”, 3.1.1: “Inadequate Consideration of Individual User Conditions”, 3.1.2: “Risk of Improper Use of Identity Data in Delegation Scenarios” and 3.1.2.1: “Missing Control of Delegation Conditions by Delegates” by carrying out Activity 3.1: “Improve Control of Identity Based on Individual Context”, the component for context-aware control complements the proposed solution by measures for an improved consideration of individual conditions, stated preferences and current contexts of users by web systems, including web applications and web services (Wild, Ast, et al., 2013; Wild and Gaedke, 2014). Here, we focus on the provision in the IdM life cycle because decisions made by identity owners at this stage fundamentally affect the security in managing personal data also during future operations.

When recalling the bank analogy we first employed when describing the problem in Section 1.2, it is obvious that customers choose banks that fit their preferences and security needs by offering appropriate assurances, like a solid legitimization process. That is, customers can control the terms their monetary capital is managed and protected. Similar to limited authorities of bank accounts, which bank customers can issue to other entities to access certain information or perform particular functions (like transfer money,

but not delete the account), we enable web users to both define and control the scope of actions of the entities authority has been issued to.

As a consequence, the component furthermore allows for mitigating risks of improperly employing authority as well as personal data in delegation scenarios (Wild et al., 2015). By enabling delegators to clearly specify the scope in which a delegate is allowed to operate on the delegator's behalf, we provide more control about the conditions of delegations (Scholtz et al., 2015a).

Chapter 5 details this synopsis of the component for context-aware control through providing insights on analysis, development and evaluation.

4.3.2 Tamper-Evidentness

In order to resolve Problem Cause 3.2: “Risk of Identity Theft and Tampering of Personal Data” and 3.2.1: “Lack of Means to Detect Identity Theft and Manipulation” by carrying out Activity 3.2: “Mitigate Risk of Identity Theft and Tampering of Personal Data”, the component for tamper-evidentness complements the proposed solution by protective means to detect identity theft and malicious manipulation of personal data. With this component, we particularly aim for reducing the risk originating from potentially compromised IdPs, regardless whether aggressors are operating outside or within the premises of IdPs (Wild et al., 2015).

Coming back to the analogy, imagine all employees of a bank—from the director over management to regular staff members—would have the keys to the safety deposit lockers containing their customers' monetary capital. With the keys, they could open the safety deposit lockers, take a look inside as well as add, modify or remove content. If such behavior would be obvious to potential bank customers, they would probably not trust the

bank any longer. Applied to the web-based management of personal data, malicious IdP operators or aggressors might have or already have acquired such extended read/write access to sensitive data, which bears the danger of data tampering happens without the data owner's knowledge.

Encrypting personal data is not a practicable choice, as it would largely complicate matters through issues like affirming public accessibility or distributing and updating cryptographic means. Although there is no satisfying solution available to prevent personal data from various kinds of manipulation (like replacement, altering, removal, addition), this component enables sound proof of the identity owner's intent by verifying the integrity of personal data stored in WebID profiles and detecting possible anomalies (Wild et al., 2014). That is, identity owners are put in the position to assure SPs and other requesting entities that the personal data they obtained is as intended by these identity owners. Through ensuring that especially identity data was not altered by unauthorized entities, we increase the authenticity of personal data and, thus, the credibility of identity owners during diverse operations, like authentication.

Chapter 6 details this synopsis of the component for tamper-evidentness through providing insights on analysis, development and evaluation.

4.3.3 Fine-Grained Filtering

In order to resolve Problem Cause 3.3: "Incomplete Range and Granularity of Access Control" and 3.3.1: "Limitation of Access Control Facilities to Specific SPs" by carrying out Activity 3.3: "Increase Range and Granularity of Access Control", the component for fine-grained filtering (FGF) complements the proposed solution by measures for controlling access to sensitive data at the attribute level. By shifting the default location of typically storing personal data from various SPs to individual IdPs, the proposed

solution for self-deterministic IdM already facilitates implementing holistic access control by identity owners. Unlike other systems that involve protection of data only at the resource level, we increase the granularity on the one hand and, on the other hand, reduce the efforts identity owners would have when dividing and distributing their personal data manually among web-accessible resources (Wild et al., 2015).

Relying on the bank analogy once again, it is evident that employees do not share the same view on account data of customers. On the contrary, the privilege to access certain information is specific to particular persons, depending on their trust level and position in the company. Transferring these characteristics concerning the management of monetary capital to social capital would result in establishing requester-specific customized views on personal data. For protecting data stored *within* resources, i.e., in particular WebID profiles containing personal data, this component applies a facade design pattern to create customized views on data (Wild, Chudnovskyy, et al., 2013a). Depending on the identity of the requesting entity, we filter semantic data within URI-addressable RDF resources in a fine-grained manner (Wild, Chudnovskyy, et al., 2013b).

Chapter 7 details this synopsis of the component for FGF through providing insights on analysis, development and evaluation.

Following the introduction of the key components that extend the proposed solution architecture in diverse security respects on IdP and SP side, the next section proceeds with explaining how to substantiate and verify the underlying concepts in practice.

4.4 Proof-of-Concept Platform

As an effort to manifest the proposed solution including the key components, we created Sociddea in the context of the EC Seventh Framework Programme for Research and Technological Development (FP7) project OMELETTE (Chowdhury et al., 2013; Tschudnowsky et al., 2013). Established on the IdMS foundation of WebID, Sociddea is an open, extensible and component-based identity (service) provider and management platform developed using ASP.NET MVC4. It puts the constituents of our proposal into effect, including the IdM life cycle. Entities can deploy and maintain their own instance of the Sociddea platform or rely on an already existing one administrated by a third party they sufficiently trust. Employing Sociddea, users can automatically create a new WebID identity incl. a WebID URI, an underlying WebID profile and an associated WebID certificate. Although Sociddea allows users for hosting their WebID profiles in the ecosystem provided by this platform, there is no constraint to do this. That is, users are also empowered to create new WebID certificates for profiles hosted somewhere else. To improve user experience and adoption, Sociddea is capable of bridging between WebID and other identity concepts such as OpenID (cf. Subsection 3.3.2) (Rienäcker et al., 2014).

According to the entity class employing the platform, Sociddea provides several customized user interfaces. For supporting the composition of web systems using WAM, it offers a web-based diagram editor (Scholtz et al., 2015b), as illustrated in Figure 4.9. Furthermore, Sociddea implements online forms human entities can employ to edit personal data associated with their identities. Sociddea also assists web engineers in describing identities of SOAP-based and RESTful web services via appropriate input masks (Braune et al., 2014). In addition to these graphical interfaces, Sociddea features a RESTful interface for

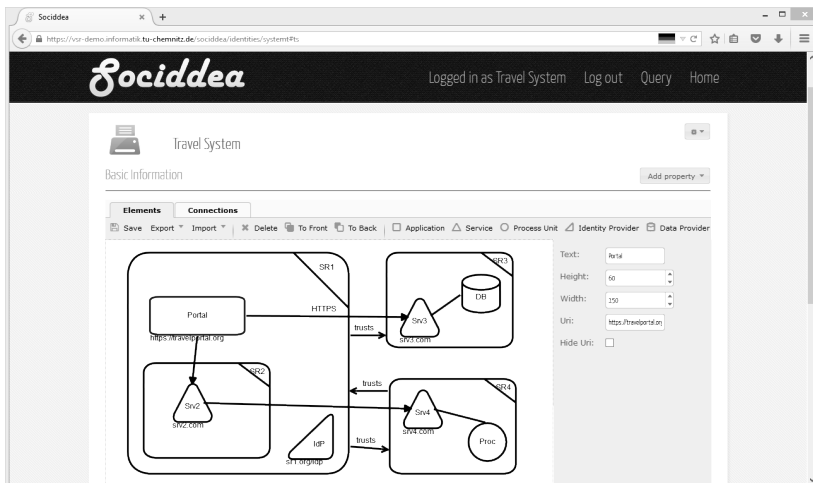


Figure 4.9: Tool Support for WAM-based Architecture Modeling

machines. Depending on the content type requesting entities declare when accessing WebID profiles managed using Sociddea, like HTML, JSON-LD or RDF/XML, they receive a representation compliant with their preferences, as exemplarily shown in Figure 4.10.

After presenting the proposal-compliant Sociddea platform that assists arbitrary entities in self-deterministic IdM, we conclude this chapter by summarizing our contributions in the next section.

4.5 Summary

In the conceptual, logical and physical designs, we formalized WebID as an approach we employ for IdM, specified the WAM ontology for describing architectures of web systems, and agreed upon several vocabularies. Based



Figure 4.10: Representations of a WebID Profile Managed Using Sociddea (Wild, Chudnovskyy, et al., 2013b)

on these designs, we proposed a solution that intends to meet two major challenges by delivering Result 1: “Improved Modeling of Security Aspects for Web Systems” and 2: “Reduced Need for Accumulation of Personal Data by Third Parties”. The proposal involves representing relevant classes of entities, i.e., persons, compositions and components, by identities and, thus, benefiting from universal identification, semantic description and authentication. For systematically managing such identities, we defined an IdM life cycle applicable to persons, compositions and components. As an effort to deliver Result 3: “Extended Means for Control and Protection of Personal Data”, we created three key components that enrich the solution architecture by complementary protective means with regard to context-aware control, tamper-evidentness and fine-grained filtering. To manifest the proposed solution, we built Sociddea as a proof-of-concept and integration platform, which we also utilize for similar purposes in the chapters to come. The platform provides basic tool support for specifying identities, including a web-accessible WAM diagramming editor to compose

web systems. Now that our proposal for enhanced security in managing personal data has been described, the following chapter proceeds with addressing the first key component in depth.

Context-Aware Control

5

To particularly address Problem Cause 3.1: “Insufficient Control of Identity Based on Individual Context”, this chapter starts with analyzing requirements and related work specific to the problem domain in Section 5.1. With the obtained analysis results, Section 5.2 describes the development of the context-aware control (CAC) component to completely meet the domain-specific requirements. Employing the success indicators and verification sources described as part of the strategy on page 61, Section 5.3 evaluates our approach to context-aware control to verify the compliance with the requirements. Finally, Section 5.4 concludes this chapter by summarizing the outcome of our contribution for context-aware control.

5.1 Analysis

With reference to Activity 3.1: “Improve Control of Identity Based on Individual Context”, Subsection 5.1.1 describes several scenarios to highlight the necessity for context-aware control as a recommended security enhancement of the proposed solution. Using these scenarios, Subsection 5.1.2 then derives a set of requirements that we employ to analyze related work, with results discussed in Subsection 5.1.3.

5.1.1 Scenarios

In addition to inherent features of the proposed solution, the capabilities of context-aware control play a decisive role in enhancing security, as shown in the next three scenarios:

Scenario 5.1: Task Delegation. From Scenario 2.3: “Holiday Replacement” we know that Alice needs to delegate tasks to others to act on her behalf. These persons should have access to certain yet not all personal data sets stored in her WebID profile in order to accomplish a task. She knows that her authorization would not be misused for other purposes, when she is delegating a task to a person she fully trusts. While this is the case with close friends, Alice is not sure about her new co-worker Casey in his role as delegate. When Casey uses Alice’s authorization intentionally, he usually acts on her behalf within her specified scope, but sometimes he also does other things in her name. This is an improper use of her authorization. Alice therefore wants to control the activities Casey does on her behalf.

Scenario 5.2: Scope of Delegation. Building upon Scenario 5.1, Alice does not want Casey to work on a particular task on her behalf beyond a fixed deadline or use other unspecified services in her name. She tries

to prevent Casey from misusing her authorization outside the scope they mutually agreed upon.

Scenario 5.3: Credential Generation. When Alice intends to create an identity as per WebID, she also has to generate a credential for this identity during provision in order to enable operations like authentication. Contrary to knowledge-based authentication systems, where she can control only a few parameters concerning a credential (like length and complexity of passwords), Alice has to take account of diverse factors that have a bearing on the WebID certificate generation. Being a not that technically experienced user, Alice does not want to employ command line tools, like *OpenSSL* (OpenSSL Software Foundation, 2016) or *keytool* (Oracle, 2016) but specialized web applications that promise an accelerated creation of her WebID certificate. However, variations in cryptographic capabilities, performance and supported devices, operating systems and web browsers complicate that matter for Alice. Depending on her individual conditions and trust needs, Alice has to carefully consider diverse factors on her own in order to find the most appropriate way for generating a WebID certificate. This involves the risk that inexperienced users, like Alice, make either no or wrong assessments, which may result in suboptimal choices that potentially impair their security and privacy.

The next subsection outlines the requirements inferred from the scenarios.

5.1.2 Requirements

By examining Scenarios 5.1 to 5.3, we derived compatibility, context-awareness, controllability, scope-compliance and secrecy as five essential requirements on approaches towards context-aware control, which are detailed in the following.

Compatibility In order to avoid degrading key features offered by the proposed solution involving WebID, like universal identification and authentication, we must rely only on already existing, standard-compliant WebID artifacts for accomplishing delegations (cf. Scenarios 5.1 and 5.2). As this involves either reusing existing or creating new WebID identities, we have to ensure the latter by a way for generating WebID certificates which is compatible with most users (cf. Scenario 5.3).

Context-Awareness As a user's individual conditions have significant impact on the process of finding a suitable way for generating a WebID certificate, we must provide means to acquire individual conditions of users before taking them into consideration (cf. Scenario 5.3). In delegation scenarios, it is essential to clearly identify delegator and delegate in order to prevent delegates from actions unwanted by the delegator, like impersonation or data disclosure (cf. Scenario 5.1). Therefore, we must make a distinction between delegators, delegates and entities that intentionally operate outside delegation contexts by employing their *common* identities.

Controllability While context-awareness facilitates taking account of individual conditions, users must retain control over the generation of their WebID certificates (cf. Scenario 5.3). That is, users must be enabled to express their preferences with regard to security, protection and trust, even though they might conflict with acquired conditions at an acceptable level. For example, users with strict requirements regarding privacy and secrecy of personal data might want to avoid creating WebID certificates – completely or in parts – by a third party they do not control (cf. secrecy). This also means that users have to be in control of the process how WebID certificates are created and made available to them. With reference to delegation, entities must be enabled to control at any time whether they act as delegator or as delegate (cf. Scenario 5.1).

Scope-Compliance Delegators must have the option to clearly specify the scope in which delegates are allowed to operate on the delegator's behalf.

To set the scope of a delegation, we must provide delegators with means to define constraints including service restrictions and validity of a delegation (cf. Scenario 5.2). Service providers must verify that delegates operate within the scopes that delegators have optionally defined. A mechanism to generate WebID certificates should ensure scope compliance by taking account of scope-related preferences like validity or certification by a third party (cf. Scenario 5.3).

Secrecy WebID is based on the premise that nobody except for the actual identity owner can modify the WebID profile owned by this entity. Otherwise, there is a severe risk of tampering and identity theft. Identity owners should therefore maintain their WebID profiles only on servers they either own or trust, and protect stored personal data against unauthorized read/write access (cf. Chapters 6 and 7). That is, delegates must not or only under certain circumstances defined by the delegator be enabled to make modifications on a delegator's WebID profile. Furthermore, the delegator must be in the position to ensure secrecy by only allowing particular requesting entities, including delegates, to retrieve specific personal data sets contained inside the delegator's profile (Scenarios 2.3 and 5.1). In line with ensuring secrecy of WebID profile data, we must prevent unwanted access from the very beginning by avoiding disclosure of private keys during WebID certificate creation (cf. Scenario 5.3). This is because an aggressor could use a disclosed private key together with the easily retrievable public key to construct an own WebID certificate, which then would permit access to personal data in a way indistinguishable from the actual data owner. As the term *private* in private key already implies, this type of key should never be accessible to an untrusted party.

Having specified the requirements, we discuss related work next.

5.1.3 Related Work

With WebID certificates being an essential artifact of the IdMS underlying the solution proposed in the previous chapter, we start the discussion of related work with their generation and, then, proceed with approaches to delegation.

Web applications are typically divided into two sides, i.e., server and client. This separation enables executing functionality on server side, on client side or partially on both sides (Ast et al., 2013). The same division applies when it comes to web-based generation of WebID certificates initiated from within web browsers.

In the traditional client-server model, tasks and workloads have been partitioned to utilize servers and relieve clients because servers typically offer more powerful computing capabilities and higher reliability. In the mode corresponding to this model, only the server side is involved in creating a WebID certificate and underlying asymmetric key pair, whereas the client side has just to provide the server side with a WebID URI. Almost all user agents are compatible with the server side generation mode in the sense that they allow submitting a request to the server for further processing. Moreover, an extensive support of programming languages, libraries and tools eases constructing WebID certificates on the server side. The server side generation mode allows creating both CA-signed and self-signed WebID certificates and, thus, third-party asserted identities as well as self-asserted identities. Generating a self-signed WebID certificate requires the private key of the entity this certificate is to be issued to, whereas a CA-signed certificate only claims that the entity is owning the private key. In any case, the server side signs the certificate with the corresponding private key from the key pair or a private key of a trusted party and, then, sends the WebID certificate back to the client side. Users need to add the WebID

certificate manually to their certificate/key store. Despite advantages in signing and compatibility, generating a WebID certificate completely on the server side requires creating or accessing an asymmetric key pair there, which implies disclosure of private key data to a third party.

In the past, the client side of a web application was solely responsible for providing a graphical user interface (GUI). Advances in JavaScript and related technologies, like Asynchronous JavaScript and XML (AJAX), enable to execute client-side code more dynamically and efficiently nowadays. Specialized frameworks enable to outsource more and more complex functional parts of a web application from server side to client side (Ast et al., 2014). Today's modern web browsers, like Mozilla Firefox, Google Chrome or Microsoft Internet Explorer, are optimized for executing JavaScript code. In line with these developments, also cryptographic routines have been adapted to run decoupled from the server side using JavaScript only, like Forge (Longley et al., 2014). That is, solely the client side is involved in generating a WebID certificate and asymmetric key pair. As a consequence, the client side generation mode allows for creating WebID certificates offline, yet users need to add produced certificates manually to their certificate/key store. While primarily designed for constructing self-signed certificates, the client side generation mode is also capable of generating a certificate signing request (Turner, 2010) and, thus, create a CA-signed certificate.

Rather than strictly separating server and client side, there is a so-called hybrid mode that divides the WebID certificate generation into two steps. In the first step, the client side creates an asymmetric key pair by making use of a particular mechanism, like the HTML5 `keygen` element (Berjon et al., 2014). The resulting private key is directly transferred to the local key store and, thus, never has to leave the client side. In the second step,

the client side submits the WebID URI, the public key and a challenge¹⁷ to the server side. When using the HTML5 keygen element, this submission is triggered via the HTML5 form element. On the server side, both the public key and the WebID URI are included during the construction of the WebID certificate. Since the private key is already available in the local key store, the WebID certificate is automatically associated to it when received by the client side. The hybrid generation mode only enables to create CA-signed certificates because the private key usually remains on a system different to the certificate-issuing one. However, the hybrid generation mode shows larger incompatibilities caused by missing native support through web browsers (SurveyMonkey Inc., 2015).

Above explanation revealed that WebID certificate generation modes not only vary in their characteristics but also depend on more dynamic factors including the user's individual conditions and trust needs. This matter has been further investigated in (Wild, Ast, et al., 2013). For example, the server-side generation mode might be unqualified due to private key disclosure, the client-side generation mode might be unqualified due to lower performance regarding key creation, and the hybrid generation mode might be unqualified due to incompatibility or missing self-signing support. Therefore, users have to be assisted in choosing an appropriate WebID certificate generation mode by taking account of their preferences, by clearly communicating potential risks associated with certain modes, and by indicating fallback measures. Having outlined the need for mechanisms that are aware of an entity's context already during credential creation, we expand the discussion to delegation approaches in the following.

SAML allows for detailed specification of authorization and delegation aspects (cf. Subsection 3.3.2), whereas eXtensible Access Control

¹⁷A challenge proves that a user possesses the private key matching to the submitted public key.

Markup Language (XACML) (Rissanen, 2013) enables describing and implementing policy-based authorization (Busch et al., 2012). Furthermore, XACML differentiates between access and delegation rights, and fosters a decentralized management of access policies. The syntax of XACML, however, complicates the task of specifying policies (Busch et al., 2012). As both languages are based on plain XML, they lack semantic features, including high expressiveness, self-descriptiveness and simplified machine interpretability, which makes them hardly compatible with WebID and, thus, with the proposed solution.

OAuth is a widely adopted open standard to authorization (Hardt, 2012). It has been designed to allow users to grant third parties access to their personal resources without disclosing their private credentials. The protocol flow in brief: An entity requests access to a protected resource. From the resource owner, the entity retrieves an authorization grant and presents it to the authorization server for validation. Once the entity received an access token, it can request the protected resource from the resource server. Evidently, this is a delegation of access rights from the resource owner to the entity. While OAuth facilitates restricting the entity's scope of action through setting constraints, it does not directly integrate with WebID.

An extension of WebID towards access delegation is discussed in (Tramp, Story, et al., 2012). The approach distinguishes involved entities by their roles as principal (delegator) and secretary (delegate). A WebID URI stored in the principal's WebID profile denotes the secretary. To act on the principal's behalf, the secretary has to add an `X-On-Behalf-Of` header to each HTTP request she issues. This HTTP header field contains the principal's WebID URI. When a capable service receives such request, it pre-authenticates the secretary using the default WebID authentication sequence (cf. Figure 4.2). To check the claimed on-behalf-of relationship to the principal, as specified in the HTTP header, the service then derefer-

ences the principal's WebID profile. If it contains an appropriate statement confirming the claim, the secretary is authorized to access the requested resource with the same access rights as the principal. Although this approach provides an almost practicable solution that enables compatibility by reusing existing WebID artifacts, it also depends on a header extension of each HTTP request. Not only does this increase complexity, but it also decreases controllability, interoperability and applicability due to relying on adequate support by user agents.

As related work does not sufficiently fulfill the requirements, the following section continues with presenting a distinct approach to context-aware control.

5.2 Development

For developing a component encapsulating our approach to context-aware control (CAC)¹⁸ on the basis of (Ast et al., 2013; Ast et al., 2014; Scholtz et al., 2015a; Wild, Ast, et al., 2013; Wild et al., 2015), we adopt the design procedure that has been applied to model the artifacts of the solution in Section 4.1. That is, we involve three models, each representing a different state of the design. Using the conceptual model formalized in Subsection 5.2.1, we specify the logical model in Subsection 5.2.2. By deriving the physical model from the logical model in Subsection 5.2.3, we establish the technical foundation for the implementation described in Subsection 5.2.4.

¹⁸For the sake of brevity, we refer to this “*component encapsulating our approach to context-aware control*” simply as CAC.

5.2.1 Conceptual Model

In order to assist users in generating WebID certificates in line with their individual conditions and preferences, as indicated in Section 5.1, CAC comprises hiding the complexity of choices by associating several interdependent concepts affecting compatibility, context-awareness, controllability and secrecy with each other under the umbrella of a so-called need for privacy. These interdependent concepts include generation mode, signing type, key strength and validity of a WebID certificate. As WebID certificates also have a central role in CAC with regard to scope-compliant delegations, we first introduce this contribution and postpone relating the concepts that affect the creation of WebID certificates to the logical design in the next subsection.

In a delegation according to CAC, a delegate acts on behalf of a delegator (cf. Subsection 2.1.2), where identity i_1 denotes the delegator, and identity i_2 denotes the delegate. This is formalized in Equation (5.1), where w_1 is the delegator's WebID URI, T_1 represents the delegator's WebID profile, w_2 is the delegate's WebID URI, and T_2 represents the delegate's profile.

$$i_1 = (w_1, T_1) \quad i_2 = (w_2, T_2) \quad i_1, i_2 \in \mathcal{I}; w_1, w_2 \in W; T_1, T_2 \in \mathcal{T} \quad (5.1)$$

While a delegator's WebID certificate is according to Equation (4.4), a delegate's WebID certificate $X_{2,k'_2} \in X_2$ is as formalized in Equation (5.2), where $X_2 \subset \mathcal{X}$ is the set of WebID certificates owned by delegate's identity i_2 . Not only does the delegate's WebID certificate contain the delegate's WebID URI w_2 and a public key $k'_2 \in K_2$ owned by the delegate, but also the WebID URI w_1 denoting the delegator's identity i_1 . Here, $K_2 \in \mathcal{K}$ describes the subset of asymmetric keys owned by delegate i_2 .

As usual, WebID certificate X_{2,k'_2} is signed with the corresponding private key $k_2'^{-1} \in K_2$ or the private key of a trusted party.

$$X_{2,k'_2} = (w_2, k'_2, w_1) \quad (5.2)$$

When the delegate authenticates to another entity, like an application or service, with such certificate X_{2,k'_2} , this entity can use the delegator's WebID URI w_1 stored within the certificate to dereference the delegator's WebID profile represented by T_1 . This profile contains a set of delegations $D_1 \subset D$ specified by delegator's identity i_1 , where D is the set of all delegations. While D_1 is described by triples $T_{1,D_1} \subset T_1$, each delegation $d \in D_1$ is described by triples $T_{1,d} \subseteq T_{1,D_1}$. Equation (5.3) defines delegation d that involves task $b \in B$ to be done by delegate i_2 taking a set of constraints Q into account. Here, WebID URI w_2 refers to delegate i_2 and B is the set of all delegatable tasks.

$$d = (b, w_2, Q) \quad (5.3)$$

With delegation d specified in delegator's i_1 WebID profile through $T_{1,d}$ and the *delegation-enabled WebID certificate* X_{2,k'_2} issued to delegate i_2 , this particular delegate referred to by WebID URI w_2 is enabled to act on behalf of the delegator referred to by WebID URI w_1 (cf. Equation (5.4)).

$$i_2 \text{ can act on behalf of } i_1 \Leftrightarrow \exists w_1, w_2 : \exists T_{1,d} \in T_1, \exists X_{2,k'_2} \quad (5.4)$$

Building on this conceptual model for CAC, the next subsection proceeds with specifying the logical flow.

5.2.2 Logical Model

To associate the concepts affecting the WebID certificate generation, CAC establishes a case differentiation that depends on the preferences stated by a user as well as obtained conditions. It allows for generating certificates on client side, on server side and in hybrid mode, yet in conformity with the analysis results outlined before and detailed in (Wild, Ast, et al., 2013). As a consequence, it utilizes server-side generation if and only if a user has declared that secrecy is not an issue. When making this choice, CAC enables constructing self-signed and CA-signed certificates, which can optionally involve a short key length and a long validity. Furthermore, it involves hybrid generation for creating CA-signed certificates if and only if a user has not a high need for privacy. In contrast, CAC employs client-side generation for severing all other preferences, but only if signing by CAs is not required. The client-side generation of WebID certificates is the only option provided to users that have a high need for privacy. Figure 5.1 depicts the described case differentiation, starting from the user input on the left side and summarizing the consequences on the right side. In line with detected conditions, like support of hybrid generation mode, CAC may block some cases, which then reduces the overall number of possible choices user can make.

Having described a way to generate the artifacts that are central to CAC with regard to delegation, we continue with specifying the corresponding process based on the conceptual model. The delegation process involves initializing a delegation, notifying potential delegates, accepting a delegation, performing the task by a delegate on behalf of the delegator, monitoring a delegation, and, finally, terminating a delegation. While Figure 5.2 illustrates the delegation process using BPMN, we describe it in the following.

Initializing a Delegation. The process of delegation is driven by a delegator represented by identity i_1 . This has various advantages regarding

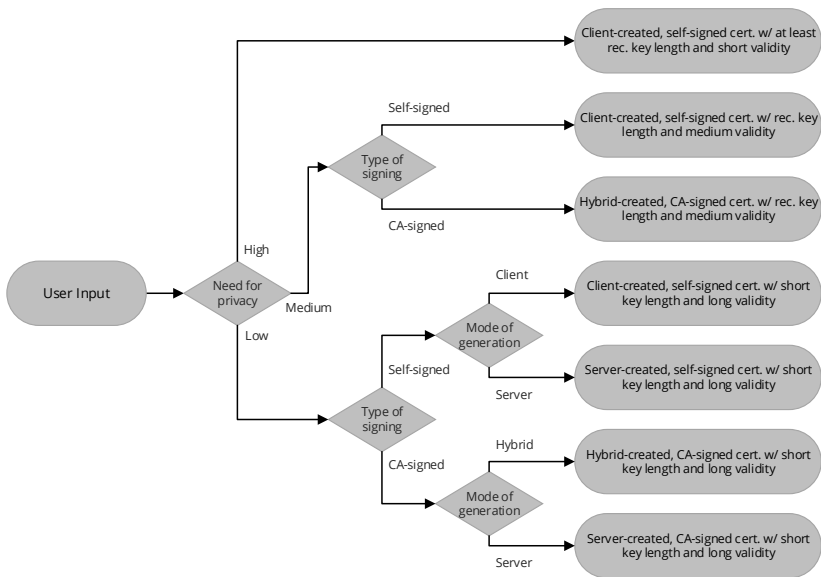


Figure 5.1: Case Differentiation for Certificate Generation According to Preferences

the purposefulness of the process and the protection of the delegator’s personal data represented by T_1 . The delegator has to formalize the intention that another entity should act as delegate in order to perform a certain task on the delegator’s behalf. Therefore, the delegator describes the delegation parameters d and optionally sets constraints Q , like service restrictions or validity (cf. ① in Fig. 5.2).

Notifying the Potential Delegate. After parameters d specifying a delegation are added to the delegator’s WebID profile (cf. ②), the delegator can inform the potential delegate. Using the delegate’s WebID URI w_2 from delegation parameters d , the delegator can obtain further information about the delegate on the basis of the thus referenced WebID profile, represented

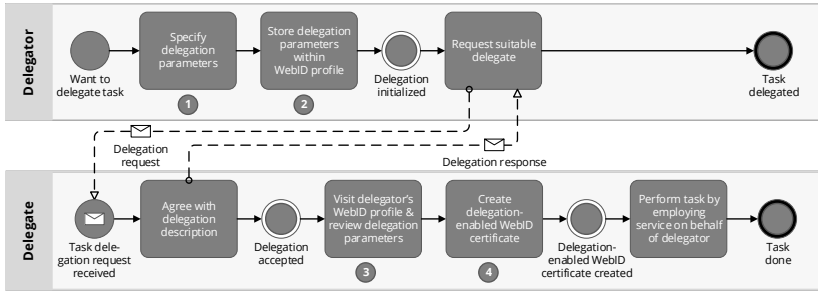


Figure 5.2: Delegation Process Model

by T_2 . That is, the delegator can inform the delegate about the delegation request by employing one of the delegate’s preferred communication methods that are outlined in the corresponding WebID profile.

Accepting a Delegation. When the delegate retrieves the delegator’s WebID profile, delegation parameters d are visible to him (cf. ③). Given that the delegate received the delegator’s notification, the delegate can read the description of task b and, consequently, either accept or reject the request the delegator intends to entrust him with. If the delegate decides to work on this task on behalf of the delegator, CAC enables to create a delegation-enabled WebID certificate X_{2,k_2} (cf. ④), as described above and in the conceptual model. Such WebID certificates contains two WebID URIs that denote the delegator’s identity i_1 and the delegate’s identity i_2 , i.e., w_1 and w_2 .

Executing a Delegation. Having received a delegation-enabled WebID certificate, the delegate can authenticate to a SP that supports WebID authentication and integrates CAC, which is represented by a dedicated component. To allow for a closer look, Figure 5.3 details CAC within the authentication sequence as per proposed solution, denoted by ⑦ in Figure 4.7. For reasons of clarity, the UML sequence diagram just depicts the WebID verifier,

the security enhancements by CAC and the WebID profiles of delegator's identity i_1 and delegate's identity i_2 , which are stored on different servers.

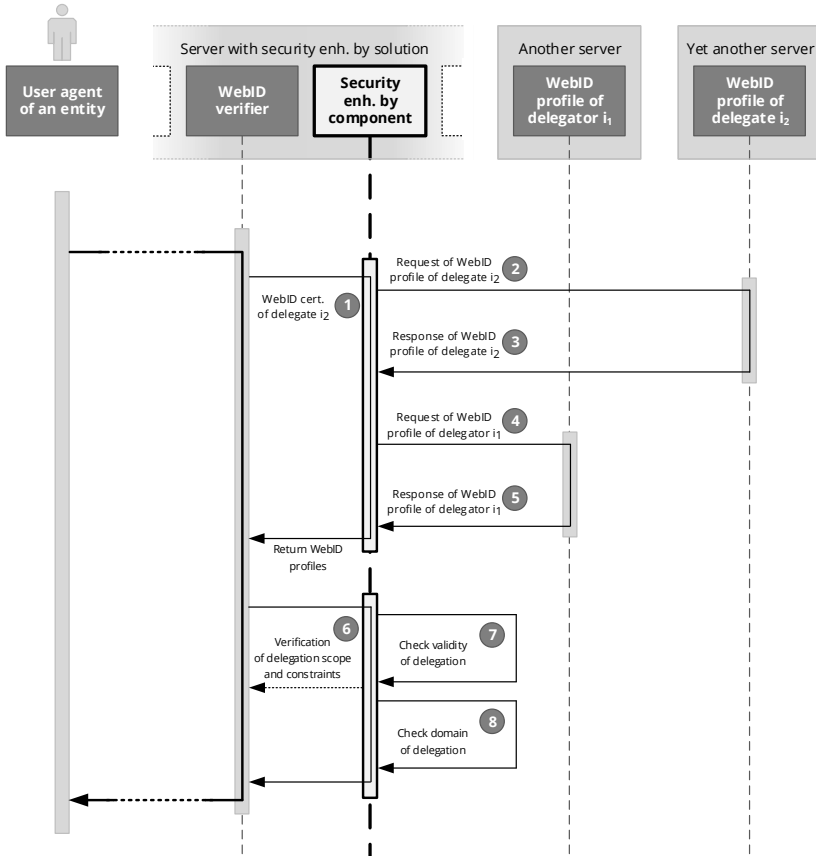


Figure 5.3: Sequence of Verification for Scope-Compliant Delegation

When authenticating to a system implementing the proposed component, the WebID verifier passes the delegate's valid delegation-enabled WebID certificate to the component (cf. ① in Fig. 5.3). It then parses the delegate's WebID certificate and extracts both WebID URIs w_1 and w_2 contained therein

(cf. Equation (5.2)). Afterwards, the component employs w_2 to request the delegate's WebID profile in ② and receives it in ③. Similar to the delegate, the component then employs w_1 to request the delegator's profile in ④ and receives it in ⑤. Both WebID profiles are passed back to the WebID verifier. The delegation parameters d are analyzed in ⑥ using the delegator's profile, which has been retrieved before. They must contain the delegate's WebID URI. Otherwise, the verification of the delegation rights fails, i.e., either the entire authentication fails or the delegate is allowed to use the SP on his own but *not* on the delegator's behalf. This decision depends on the implementation of the component by the SP. With regard to validity and service (domain) restrictions, the delegation constraints Q defined by the delegator are checked in ⑦ and ⑧. Once a constraint verification fails, the delegation fails as well and, similar to the failed delegation rights check discussed above, possibly the entire authentication. Provided a successful authentication and delegation, the delegate can perform the task using the SP on the delegator's behalf, as defined in Equation (5.4). Here, the SP and the delegate might have access to the delegator's personal data to be used to perform task b by the delegate.

Monitoring a Delegation. Similar to other digital or traditional access control procedures, it is possible that the delegate will face problems while working on a task entrusted to him by the delegator. The delegate could discuss these issues with the delegator to find an appropriate solution, e.g., by adjusting the deadline associated to the delegation accordingly. Under certain circumstances, however, it is important to know the progress made by the delegate independently from personal status reports (cf. Scenario 5.1). In addition to delegates, also delegators can authenticate to SPs employing their own WebID identities. For instance, when a SP offers status indicators or activity logs to customers, then a delegator could find out about a delegate's progress with respect to the task assigned to him. Offering such mechanisms, however, is the responsibility of SPs.

Terminating a Delegation. When completing the task on the delegator's behalf before the deadline, the delegate can optionally inform the delegator about this success using a suitable communication channel possibly outlined in the corresponding WebID profile. If the delegate could not finish the task within the given time frame, then the authorization to act in the delegator's name is no longer valid. As a consequence, CAC will not allow the delegate to work on the task after passing the deadline. Furthermore, CAC also enables the delegator to terminate the delegation at any time. This might be necessary when the task is expendable for some reason like priority shifts. For the current task, the delegator can do so by changing or removing the delegate's WebID URI from the delegation parameters. All these types of completing a delegation necessitate to be aware of updates affecting the delegation. It is therefore required that a SP implementing the CAC component either checks the authorization or automatically authenticates the delegate again on a regular basis.

Having created the logical model from the conceptual one, the next subsection continues with detailing technical aspects of CAC.

5.2.3 Physical Model

In order to establish the basis for the technical implementation of the design artifacts, this physical model further specifies the logical model.

For identifying the *real* entity which is using a service, CAC comprises adding identifiers for both the delegator denoted by w_1 and the delegate denoted by w_2 to a WebID certificate issued to the delegate. Figure 5.4 illustrates this WebID certificate (top, left). The original semantics of the WebID certificate remains unchanged, i.e., the SAN field of the certificate still contains the WebID URI referring to the entity that will primarily employ this certificate (cf. Ⓐ in Fig. 5.4). In the delegation context, this is the delegate. In

addition to the rather common data contained in the delegate's WebID certificate, CAC exploits the *Issuer* and the *Issuer Alternative Name (IAN)* certificate fields (cf. **Ⓑ**). These fields are used to denote the delegator represented by i_1 both by name and by WebID URI w_1 referring to the corresponding WebID profile T_1 (cf. Equation (5.1)).

The delegate's WebID profile (bottom, left), represented by T_2 , remains as it is, whereas WebID profile of the delegator (top, right), represented by T_1 , needs to be extended for storing the delegation parameters (cf. Equation (5.3)). This extension is necessary to prevent attackers to act in the delegator's name by creating such *delegation-enabled WebID certificate* (cf. Equation (5.2)) on their own. It is recommended to include either the entire set of delegation parameters d or a reference to it in the delegator's WebID profile (cf. **Ⓒ**).

To specify a delegation in terms of task, associated constraints and potential delegate, CAC makes use of the WebID Delegation Language (WDL) (Wild et al., 2015), which is a semantic vocabulary based on RDF. In WDL, the description of a task is a URI pointing to a resource containing further information about the work to be done. Such description is not directly included in WDL to separate concerns. Furthermore, the way tasks are actually described is outside the scope of CAC, yet we recommend the use of semantic vocabularies. WDL enables delegators to define constraints regarding validity and domain of a delegation. Here, the validity is represented by a time stamp indicating the end of a delegation and, thus, the deadline of the assigned task. The WDL domain constraint defines a restriction of services that a delegator authorizes a delegate to use. That is, by specifying the domain name of a service, the delegate is only allowed to perform the task within this particular domain. WDL involves another RDF triple to refer a delegate by means of a WebID URI. Listing 5.1 shows the structure of a WDL-based delegation specification in Turtle syntax:

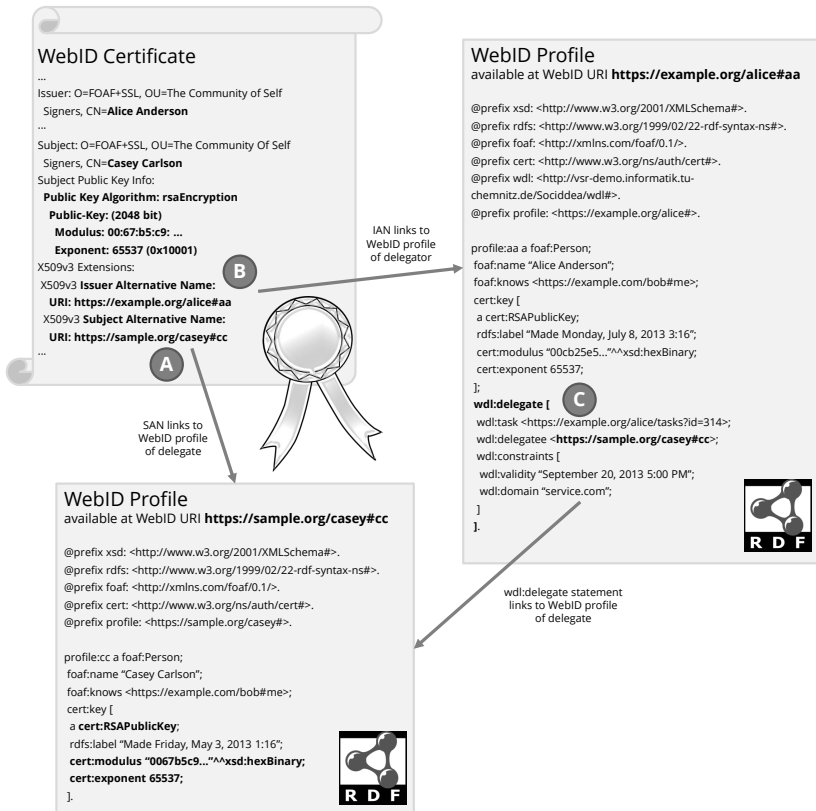


Figure 5.4: WebID Artifacts For Context-Aware Control in Delegations (Wild et al., 2015)

Now that all artifacts relating to CAC have been described, the following subsection outlines the implementation of the design.

Listing 5.1: Template of Delegation Specification as per WDL

```
1 <WEBID URI OF DELEGATOR> wdl:delegate [  
2   wdl:task <URI POINTING TO TASK DESCRIPTION>;  
3   wdl:constraints [  
4     wdl:validity DEADLINE;  
5     wdl:domain SERVICE  
6   ];  
7   wdl:delegate <WEBID URI OF DELEGATE> ].
```

5.2.4 Implementation

To put the conceptual, logical and physical model into practice, we consolidated all design artifacts in a self-contained component and exemplarily integrated this CAC component into the Sociddea platform, which has been initially introduced in Section 4.2.

With Sociddea integrating CAC, an entity can automatically create a new identity consisting of WebID URI, WebID profile and WebID certificate. Figure 5.5 depicts the GUI for identity creation presented to human entities. Here, an HTML form permits human entities to enter their basic set of personal data and their preferred WebID URI. In addition, human entities can also create an identity for a web system or a web service, as shown in this figure. Before initializing the provisioning of a new WebID certificate, users are put in the position to express their individual preferences by making certain selections with regard to privacy need, signing type, generation mode, key strength and validity (cf. Figure 5.1).

The component for generating WebID certificates according to user preferences and context builds upon JavaScript and involves the Forge library to execute cryptographic operations (Longley et al., 2014). Not only does this enable creating certificates directly from within web browsers but also on the server side. In compliance with (Barker et al., 2012), the com-

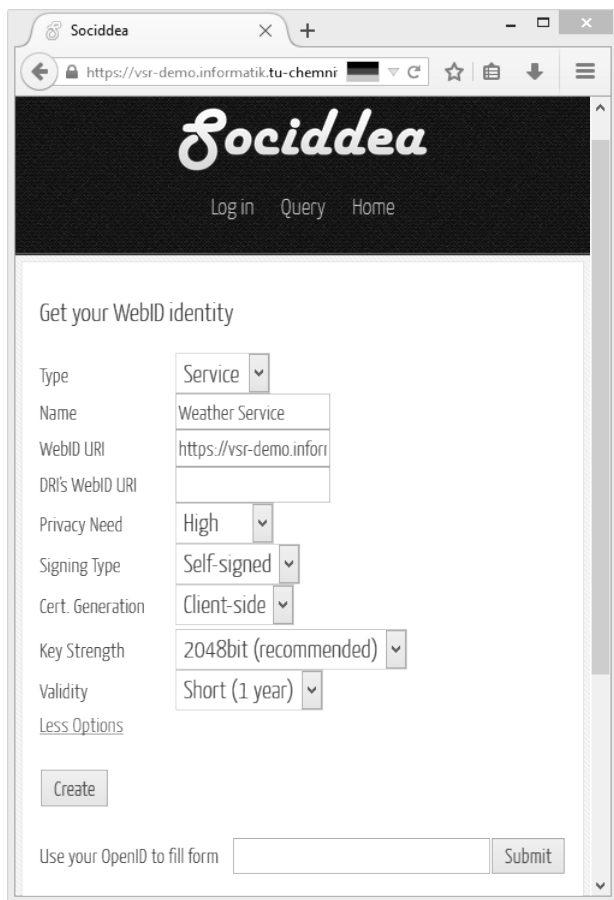


Figure 5.5: WebID Certificate Creation with Sociddea

ponent uses 2048 bits as recommended key length, while also allowing 1024 bits (short) and 4096 bits (long) depending on the declared privacy need. Furthermore, it defines a default validity of 1 year (short), but also accepts 3 years (medium) and 10 years (long), even though the latter options are not recommended for safe and productive use.



Figure 5.6: Delegation Creation with Sociddea (Scholtz et al., 2015a)

In much the same way as during common identity creation, users can benefit from this assistance in generating WebID certificates during dele-

gation scenarios, as exemplarily shown in Figure 5.6. There, a delegator initializes a delegation by specifying several parameters (cf. ①¹⁹), which are then stored in the corresponding WebID profile in ②. Having received the notification from the delegator, the delegate can address the delegator's WebID profile. When using a web browser to address this WebID profile, Socidea automatically informs the authenticated delegate about the offer to act on the delegator's behalf in ③. In the event of agreeing with the offer, the delegate can create a delegation-enabled WebID certificate with regard to known preferences in ④.

After completing development, the next section is about evaluation of CAC.

5.3 Evaluation

For evaluating CAC, Subsection 5.3.1 first determines the characteristics to be assessed. Subsection 5.3.2 then describes the evaluation procedure that takes account of these characteristics, and, finally, Subsection 5.3.3 discusses the evaluation results.

5.3.1 Characteristics

By defining five requirements on approaches for context-aware control in Subsection 5.1.2, we created the foundation for a systematic analysis of related work and a well-directed component development. Building on this foundation, we reapply the requirements—compatibility, context-awareness, controllability, scope-compliance and secrecy—as criteria for evaluating our work in the context of context-aware control.

¹⁹The numbering in Figure 5.6 matches the numbering in Figure 5.2.

5.3.2 Procedure

Relying on the success indicators and verification sources declared in Subsection 2.3.4, we employ the proof-of-concept implementation as well as unit and operational acceptance tests to evaluate the component with regard to the characteristics defined in Subsection 5.3.1. In order to assess the degree to which each evaluation criterion has been fulfilled by CAC, the four-level rating system, known from Subsection 3.2.3, suits our purpose.

5.3.3 Results

On the basis of the evaluation results obtained by applying the procedure specified in Subsection 5.3.2, we discuss how CAC addresses each criterion defined in Subsection 5.3.1. Building upon the discussion of each evaluation criterion, we draw a conclusion involving a rating.

Compatibility While delegation according to CAC only involves existing WebID artifacts except for a so-called delegation-enabled certificate, SPs need to support it. A delegation-enabled certificate is standard-compliant as per (Cooper et al., 2008). Compared to common WebID certificates, its creation effort is almost the same, i.e., it requires an additional entry for exactly referring to the delegator by a WebID URI. When unsupported by a SP, a delegate can still employ a delegation-enabled certificate for default authentication but not to obtain extended authority.

To conclude, CAC largely fulfills the compatibility criterion.

Context-Awareness Relying on already existing representations of delegator and delegate through two different WebID identities facilitates their distinguishability. Here, employing a delegation-enabled rather than a common WebID certificate does not change the underlying identity but enables assigning a role depending on the context provided through such

special certificate. With regard to acquiring the individual conditions of a user, CAC offers only limited capabilities. Although the component can detect some incompatibilities, e.g., missing support of `keygen` element, more advanced features, like estimated performance of client side, are missing yet important for an optimized context-aware generation of certificates.

To conclude, CAC partially fulfills the context-awareness criterion.

Controllability Besides context-awareness, CAC enables entities to make adjustments to recommended parameters for generating WebID certificates, unless they conflict with the underlying model (cf. Figure 5.1). Furthermore, using JavaScript allows for offline generation of certificates, which contributes to increase control. In terms of delegation, an entity can control whether to act as a delegate via a distinct WebID certificate.

To conclude, CAC completely fulfills the controllability criterion.

Scope-Compliance By setting constraints, delegators can specify the scope of a delegation with respect to validity and service domain. Yet, the completeness of this set of restrictions is to be questioned. Furthermore, the verification of compliance requires capable SPs that potentially have to perform such verification on a regular basis in case of changing conditions.

To conclude, CAC partially fulfills the scope-compliance criterion.

Secrecy With CAC, entities do not have to disclose private key data when generating WebID certificates on the client side only. This is the default setting offered by the component and the only valid options for entities with a high need for privacy. As delegates employ one of their own identities for delegation, they do not have to cope with personal data issues caused by redundant repository. In consequence, delegators can control access to personal data stored in their WebID profiles according to the identities which potential delegates will use.

To conclude, CAC completely fulfills the secrecy criterion.

By completing the component evaluation with results indicating an overall large fulfillment of the criteria, the following section sums up the outcome of CAC.

5.4 Summary

With delegations according to our approach to context-aware control, entities are enabled to act on other entities' behalves in scope-compliant ways. By providing a component to prevent unwanted exploitation of authority, we contributed to increase control for delegators in delegation scenarios. Through context-aware control, entities also benefit from a simplified and accelerated credential generation that takes account of the individual context and stated preferences towards security and privacy. Based on three scenarios, we derived five requirements to systematically analyze related work and align the development of our contribution. Taking compatibility, context-awareness, controllability, scope-compliance and secrecy into account, we conceptually, logically and physically designed the contribution for context-aware control. Through focusing on reuse of existing identity, the contribution allows for clearly distinguishing between delegator and delegate. To prove the concept for context-aware control, we transferred the design artifacts into a self-contained component, which we then exemplarily integrated into the Sociddea platform. From the results obtained by evaluating the component, we concluded an overall large fulfillment of the requirements, but with some justifiable drawbacks in terms of context recognition.

Now that the component for context-aware control has been described, the following chapter continues with detailing the second key component.

Tamper-Evidentness

6

To particularly address Problem Cause 3.2: “Risk of Identity Theft and Tampering of Personal Data”, this chapter starts with analyzing requirements and related work specific to the problem domain in Section 6.1. With the obtained analysis results, Section 6.2 describes the development of the tamper-evidentness (TE) component to completely meet the domain-specific requirements. Employing the success indicators and verification sources described as part of the strategy on page 61, Section 6.3 evaluates our approach to tamper-evidentness to verify the compliance with the requirements. Finally, Section 6.4 concludes this chapter by summarizing the outcome of our contribution for tamper-evidentness.

6.1 Analysis

With reference to Activity 3.2: “Mitigate Risk of Identity Theft and Tampering of Personal Data”, Subsection 6.1.1 describes several scenarios to highlight the necessity for tamper-evidentness as a recommended security enhancement of the proposed solution. Using these scenarios, Subsection 6.1.2 then derives a set of requirements that we employ to analyze related work, with results discussed in Subsection 6.1.3.

6.1.1 Scenarios

In addition to inherent features of the proposed solution, the capabilities of tamper-evidentness play a key role in enhancing security, as shown in the next three scenarios:

Scenario 6.1: Manipulation of Personal Data. WebID identity owner Alice relies on a third-party IdP for managing her identity. When hosting her profile on such IdP, Alice was aware of the risk that a third party could access and disclose personal data contained in her WebID profile²⁰. However, Alice is unaware of the fact that Mallory operates the IdP that also hosts Alice’s WebID profile. Mallory has malicious intentions, i.e., she wants to impair Alice by manipulating her identity and associated personal data. Even though Alice controls access to her WebID profile at the resource level to avoid personal data disclosure by unwanted requesting entities from outside, she cannot apply the same access control for the inside. Apart from accessing Alice’s personal data, Mallory can also manipulate it, e.g., she can change Alice’s email address to point to one of her own or add new social connections linking to strange guys or Alice’s enemies. That is, Mallory can tamper Alice’s user profile data without her intent, knowledge

²⁰This common risk affects all unencrypted files hosted on third-party operated servers.

and notice. While a human entity is the attack target in this case, this deficiency also applies to identities of other entity classes, like web services.

Scenario 6.2: Identity Theft. Based on Scenario 6.1, Mallory wants to take full control of Alice's WebID profile. For this purpose, Mallory adds her own public key to Alice's profile and creates a WebID certificate linking to Alice's profile. That is, Mallory's WebID certificate contains the public key that is also available in Alice's WebID profile. As a consequence, Mallory is enabled to authenticate herself using Alice's identity and, thus, impersonate Alice in her malicious actions. To make this even worse, Mallory may remove all other public keys from Alice's WebID profile, so that Alice cannot authenticate herself to other subjects any longer. Alice would need to inform all her social connections and services accessing her profile about the forgery. Finally, Alice would need to choose a new WebID URI and re-create both her certificates and her profile.

Scenario 6.3: Temporary Exploitation. Mallory permanently tampered Alice's personal data in Scenarios 6.1 and 6.2. Alice might find out about such malicious manipulations sooner or later. To cover her tracks, Mallory adjusted her approach and only tampers Alice's WebID profile data on special occasions, now. That is, her modifications to Alice's profile are temporary instead of permanently. As an example, before authenticating to a service as Alice, Mallory adds her public key to Alice's WebID profile (cf. Scenario 6.2). After making use of this service as Alice, Mallory reverses her malicious changes by removing her public key. In the time between these events, Mallory could send emails, book or order something in Alice's name. In this case, Alice would probably not discover that a malicious manipulation of her personal data was the reason for future events and issues.

The next subsection outlines the requirements inferred from the scenarios.

6.1.2 Requirements

By examining Scenarios 6.1 to 6.3, we derived accessibility, applicability, integrability, non-impairment and public verifiability as five essential requirements on approaches towards tamper-evidentness, which are detailed in the following:

Accessibility Regardless of whether identity owners control access to personal data, we have to ensure the possibility to obtain certain data sets without prior legitimization. By implementing the open silo model through relying on WebID, the proposed solution enables arbitrary entities to make data associated with their identities available. Only in this way, we can benefit from discovery and description of identities, like detecting suitable components, and contribute to the growth of the Linked Data cloud.

Applicability Protection against tampering and identity theft should be universally and easily applicable to non-existing and existing identities, i.e., independent from a specific representation of personal data contained in a WebID profile.

Integrability The fact that malicious manipulations of personal data in WebID profiles can happen on a permanent as well as on a temporary basis (cf. Scenario 6.3) requires a data integrity check on every access attempt to a WebID profile. That is, an integrity verification must be integrated into the WebID authentication process to allow identity owners for detecting tampering of personal data stored in their WebID profiles. In order to simplify integrating such protective measure, the process of authentication and retrieval of WebID profiles should be modified as little as possible, with modifications implying none or only few additional dependencies.

Non-Impairment Making use of tamper protection must not impair human and non-human entities that rely on WebID-based identities to accomplish their tasks. Such protective mechanism is a recommended yet optional

enhancement. It must therefore also not impair entities that are not employing it, e.g., SPs or IdPs during authentication, attribute retrieval and management.

Public Verifiability While identity owners cannot completely prevent manipulation of WebID profile data by malicious server operators or external aggressors, we have to reduce the impact of attacks by making affected entities aware of unintended changes to personal data. To detect tampering and identity theft, it must therefore be possible to publicly verify the integrity of personal data contained in WebID profiles.

Having specified the requirements, we discuss related work next.

6.1.3 Related Work

Literature about file systems and database systems broadly deals with the topic of ensuring data integrity, yet this discussion of related work focuses on the suitability of approaches to detect tampering attacks on web systems only.

Even though encryption can protect personal data in general, it is inappropriate for WebID. Profiles have to be at least partially accessible for authentication of identity owners and for queries of requesting entities. Furthermore, identity owners would need to distribute keys for decryption to an unknown number of potential requesting entities, which also impairs public verifiability. Alternatively, central authorities could consolidate the key management. However, this does not match our proposal towards self-deterministic IdM.

When signing personal data by utilizing the public keys that are already contained in WebID profiles, they would need to be protected from manipulation as well. This could be accomplished by a PKI involving CAs or a

WoT. By investigating the common domain model as part of the state of the art analysis in Subsection 3.3.2, we discovered several general drawbacks of PKIs as prime examples for IdMSs of this type, especially in terms of openness and security. PKIs utilize so-called CAs to issue certificates after a strong review process, i.e., the owner of a CA-signed certificate has to prove intensely to be the identity claimed. Due to this validation process, a PKI integration into WebID would not only increase the effort of creating new WebID certificates by users, but also negatively affect applicability, integrability and non-impairment. While a PKI associates certificates with real world identities, a WebID identity is allowed to be more anonymous to provide privacy. While WebID allows for adapting this model, e.g., similar to signing WebID certificates by a trusted third party rather than by an identity owner, it interferes with the decentralized approach of WebID that intends to involve and empower individuals instead of authorities and, thus, also contradicts our self-deterministic IdM vision.

By contrast to PKI, the WoT concept represents a flat hierarchy only relying on individuals (Caronni, 2000). The WoT concept is more compatible to the open silo model and, by implication, also WebID (cf. Subsection 3.3.2), but it needs member discovery and makes updating public keys and signatures difficult due to their necessary distribution and inclusion in other data stores, e.g., user profiles. In consequence, the WoT concept would cause additional efforts for identity owners, also by complicating integration of signatures and keys.

As related work does not sufficiently fulfill the requirements, the following section continues with presenting a distinct approach to tamper-evidentness.

6.2 Development

For developing a component encapsulating our approach to tamper-evidentness (TE)²¹ on the basis of (Wild et al., 2014; Wild et al., 2015), we adopt the design procedure that has been applied to model the artifacts of the solution in Section 4.1. That is, we involve three models, each representing a different state of the design. Using the conceptual model formalized in Subsection 6.2.1, we specify the logical model in Subsection 6.2.2. By deriving the physical model from the logical model in Subsection 6.2.3, we establish the technical foundation for the implementation described in Subsection 6.2.4.

6.2.1 Conceptual Model

To improve the protection of personal data in WebID profiles against malicious manipulation, TE has to enable a profound data integrity verification. Being aware that data integrity of potentially third-party hosted profiles cannot be protected, TE aims at making malicious manipulations recognizable to identity owners and requesting entities. For guaranteeing authenticity and integrity of personal data in WebID profiles, TE needs to involve digital signatures. Signing the underlying RDF attributes ensures that personal data provided in the profile originates from the *real* identity owner²². Furthermore, to prevent malicious change of public keys, TE has to accomplish a binding key(s) and WebID profile in an unchangeable manner.

As serializations of RDF triples can express personal data within WebID profiles in various ways using different syntaxes, TE relies on RDF graphs

²¹For the sake of brevity, we refer to this “*component encapsulating our approach to tamper-evidentness*” simply as TE.

²²Here, we intentionally use the term *identity owner* and not the entity described by personal data.

that represent WebID profiles, with this equivalence formalized in Equation (4.2). To address different orders of RDF triples and blank nodes²³, TE performs a canonicalization by utilizing the one-step deterministic labeling method proposed in (Carroll, 2003) and the methodology described in (Tummarello et al., 2005). Without changing semantics, it transforms G representing the WebID profile data into its canonical representation $\tilde{G} \in \mathfrak{G}$ using function η , as formalized in Equation (6.1).

$$\eta(G) = \tilde{G} \sim \tilde{T} \quad \tilde{T} \in \mathfrak{T} \quad (6.1)$$

To sign a WebID profile in canonical representation \tilde{G} , TE combines the hashes of each statement $\theta(t)$ into a single value. The hash of a statement is computed by concatenating the hashes of each subject, predicate and object, which then will be hashed again, as formalized in Equation (6.2). While (Kasten and Scherp, 2013) creates an overall hash of the sorted hashes, TE deterministically canonicalizes WebID profile data. So, the hashes of all statements can be concatenated again and a new hash can be calculated from them.

$$m_{\text{dig}} = \theta(\tilde{T}) = \theta\left(\sum \theta(t)\right) \forall t \in \tilde{T} \quad (6.2)$$

For the purpose of signing the hash obtained above, identity owner i has to choose a *main* asymmetric key pair (k'_*, k'^{-1}_*) , where both keys $k'_*, k'^{-1}_* \in K$ are in possession of this particular identity owner. While the main key pair is specially used for signing and verifying the WebID profile represented by T , the identity owner can employ it as a WebID certificate for authenticating as usual. Equation (6.3) formalizes the creation of signature $m_{\text{sig}} \in M$ using function β with main private key k'^{-1}_* and hash m_{dig} as parameters.

$$m_{\text{sig}} = \beta(k'^{-1}_*, m_{\text{dig}}) \quad (6.3)$$

²³Blank nodes in RDF are nodes without a URI reference and they aggregate concepts like a person's address (Schreiber and Raimond, 2014).

Signature m_{sig} is attached to the WebID profile of identity owner i so that requesting entities can retrieve certain personal data sets as well as corresponding signature data. In addition to signing WebID profiles per se, the means for verifying signatures have to be accessible without requiring unreasonable efforts by requesting entities. As signing a WebID profile is insufficient to protect it against identity theft (cf. Scenario 6.2), signature verification of the WebID profile would fail as well. An attacker could sign the WebID profile once again with an own private key that is associated to the public key that has been added recently to the attacked WebID profile.

To address this matter, TE binds an identity owner's main public key $k'_* \in K$ to the corresponding WebID URI $w' \in W$. Using WebID URI w' , function ζ yields k'_* , as formalized by Equation (6.4). That is, changing the main public key would invalidate the owner's WebID identity $i = (w', T)$ (cf. Equation (4.5)).

$$\zeta(w') = k'_* \tag{6.4}$$

Building on this conceptual model for TE, the next subsection proceeds with specifying the logical flow.

6.2.2 Logical Model

For ensuring requesting entities that personal data within a WebID profile data is as intended by the corresponding identity owner, we derive three main activities from the conceptual design: signing personal data, storing/retrieving a signature, and verifying data integrity of profiles. Figure 6.1 illustrates these activities using BPMN, while we describe them in the following.

Signing WebID Profile Data. By signing a WebID profile, an identity owner (cf. top of Figure 6.1) proves that personal data stored in the profile is valid and was not changed by another party. In order to avoid signing tampered data, the data integrity of the WebID profile needs to be checked (cf. ① in Figure 6.1) prior to updating relevant data (cf. ②) and creating a signature (cf. ③). Algorithm 6.1 specifies in pseudo code notation the signing of WebID profile data. As per conceptual design, TE employs a canonicalized RDF graph representation of WebID profile data for computing hash values independent from specific data serializations, e.g., RDF/XML or Turtle.

To avoid disclosing an identity owner’s private key to a third party, the signing process is divided into server side and client side. The server side

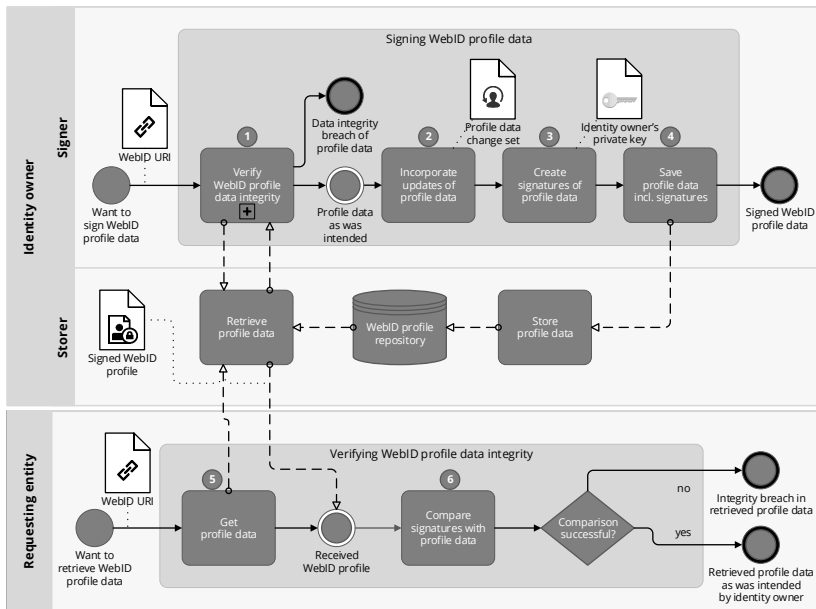


Figure 6.1: Tamper-Evidentness Process Model

computes hash values of each *minimum self-contained graph (MSG)* (Tumarello et al., 2005) found in the graph representation of the WebID profile. It combines all hash values to a signing request afterwards (cf. lines 3 to 7). The client side analyzes this request and signs the content. It creates the signatures through encrypting each hash value with a private key (cf. lines 8 to 11). The identity owner has to select the corresponding private key beforehand.

Once received by the server side (cf. lines 12 to 15), TE verifies the signed response containing the signatures. Provided that the verification was successful, it then stores the signatures in the identity owner's WebID profile in ④. When storing (cf. middle of Figure 6.1), TE applies the method proposed in (Sayers and Eshghi, 2002). This method closely links a public key stored in the profile with the WebID URI and, thus, assists in detecting attacks that aim at removing profile data and signatures.

Discovering WebID Identity Theft. Following the principle of empowering individuals instead of authorities, we could not solely rely on attaching signatures to personal data²⁴. By creating a binding between public key and WebID URI, TE therefore ensures that this key cannot be changed without losing personal relationship data such as incoming social connections expressed via `foaf:knows` WebID URIs. Having the public key stored in the WebID URI allows detecting the same key inside the profile. This facilitates not only discovering identity theft done by detecting malicious key manipulation, but also using the public key for verifying signatures.

Verifying WebID Profile Data Integrity. To make signed WebID profiles easily verifiable for requesting entities, we integrate the verification process into the WebID authentication routine. It is triggered when the WebID

²⁴By gaining access to the system storing the WebID profile, an attacker could tamper identity data and manipulate signatures stored in the profile. Due to this vulnerability to attacks, an external authority would be required to provide proof of correctness.

Algorithm 6.1: Digitally Signing Personal Data Stored in WebID Profiles

Input: WebID URI w' , Private Key $k_*'^{-1}$
// on server side
1 get WebID profile G from w' ;
2 generate canonicalized graph \tilde{G} from G as per (Carroll, 2003; Tummarello et al., 2005);
3 **repeat**
4 | delete MSG \tilde{G}_* from \tilde{G} ;
5 | create hash value m_{dig} of \tilde{G}_* ;
6 | add m_{dig} to server inquiry m_{inq} ;
7 **until** \tilde{G} is empty;
// on client side to avoid private key disclosure
8 **foreach** hash value m_{dig} in server inquiry m_{inq} **do**
9 | create signature m_{sig} by encrypting m_{dig} with $k_*'^{-1}$;
10 | add m_{sig} to client response m_{res} ;
11 **end**
// on server side
12 **foreach** signature m_{sig} in client response m_{res} **do**
13 | **if** signature m_{sig} is invalid **then** stop;
14 **end**
15 add all signatures in client response m_{res} to graph G ;

profile has been loaded. For verifying signed profile data (cf. bottom of Figure 6.1), TE involves receiving a WebID profile via a WebID URI in (5). It tries to detect a plausible public key²⁵ inside the profile. Such public key has to correspond to the key representation stored in the WebID URI. As soon as a valid public key has been found, TE computes hash values of WebID profile data as mentioned in the signing process. It then compares the hash values with the hash values retrieved by decrypting the signatures using the public key, as indicated by (6). The data integrity of WebID profiles cannot be guaranteed in case a detection or verification step has failed. Handling

²⁵A public key with a common length, e.g., 2048 bit.

failed verifications depends on the scenario and authentication target. It is therefore not part of this contribution and needs to be addressed separately.

To allow a closer look, Figure 6.2 details TE within the authentication sequence as per proposed solution, denoted by ⑥ in Figure 4.7. For reasons of clarity, the UML sequence diagram just depicts the WebID verifier, the security enhancements by TE and the WebID profile of identity *i* stored on another server.

Verification begins with invoking the WebID verifier to check the credential supplied by the owner of identity *i*. As usual, the identity owner uses a WebID certificate as a potential proof of identity *i*. The WebID verifier passes the supplied WebID certificate to TE, represented by a dedicated component, for validating the integrity of the WebID profile associated with *i* (cf. ① in Figure 6.2). TE obtains the WebID URI from the SAN property of the WebID certificate and tries to extract the key representation from inside the obtained WebID URI in ②. If there is no such key representation available inside the WebID URI, the WebID profile is not integrity-protected using TE. Consequently, TE falls back to the WebID authentication sequence as per our proposed solution in ③ and either abort or proceed with the common verification of identity *i* in ⑦²⁶. Otherwise, the component requests the WebID profile referred to by the WebID URI in ④. By transmitting personal data including possibly available signatures, the server hosting the WebID profile responds to the request.

TE then transforms each public key listed in the retrieved WebID profile into a key representation according to an agreed method and compares it with the key representation from the WebID URI in ⑤. If a public key matches the key represented in the URI, this is the main public key,

²⁶While depending on the actual implementation, this allows for making tamper-evident WebID profiles a mandatory requirement for authentication.

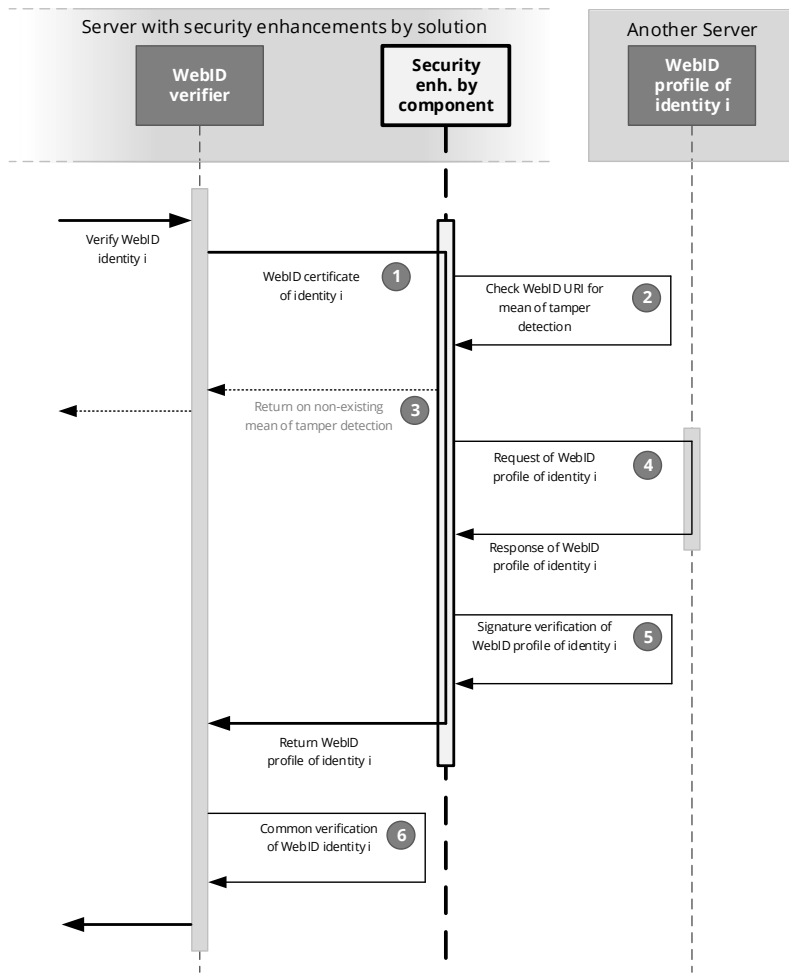


Figure 6.2: Sequence of Detecting Tampering and Identity Theft

which the component uses to verify the signatures in the identity owner’s WebID profile. No match would be an indication for manipulation of personal data and cause a failed integrity verification. In this case, the

component sends a failure notification to the WebID verifier, which, in turn, declares the WebID verification as failed in ⑥.

Using the main public key, TE verifies the signatures contained in the retrieved WebID profile by means of the hashes created from personal data in the way described before (cf. ⑤). If the signature verification fails, it sends a failure notification to the WebID verifier. Depending on the implementation and other protective measures, the component passes either a success notification or the integrity-verified WebID profile data to the WebID verifier.

Having created the logical model from the conceptual one, the next subsection continues with detailing technical aspects of TE.

6.2.3 Physical Model

In order to establish the basis for the technical implementation of the design artifacts, this physical model further specifies the logical model.

TE involves binding a representation of a main public key k'_* to a WebID URI w' that denotes an identity i (cf. Equation (6.4)). As the length of such public key, e.g., 2048-bit or 4096-bit, makes it inconvenient to store it directly inside a WebID URI, TE utilizes general-purpose hash algorithms for generating a representation of the main public key. In comparison to public keys per se, hash values allow for much shorter variants, e.g., 160-bit or 256-bit, without sacrificing security. For attaching key representations to WebID URIs, the approach converts the hash value of a main public key via Base64 encoding with Uniform Resource Locator (URL) and filename safe alphabet (Josefsson, 2006). Relying on the Secure Hash Algorithm (256 bits) (SHA-256) extends a WebID URI *only* by 44 characters.

To transform WebID profile data represented by RDF graph G into a canonical form, as defined in Equation (6.1), TE first converts it into N-Triples notation. Since transforming an RDF graph into N-Triples notation does not imply any fixed sequence, TE then has to perform a sort in lexicographic order to enable computing the same hash values for the same graph. As one and the same blank node might have different identifiers²⁷ without changing semantics, TE applies the one-step deterministic labeling method that names blank nodes in a deterministic fashion (Carroll, 2003).

Now that all artifacts relating to TE have been described, the following subsection outlines the implementation of the design.

6.2.4 Implementation

To put the conceptual, logical and physical model into practice, we consolidated all design artifacts in a self-contained component and exemplarily integrated it into the Sociddea platform, which has been initially introduced in Section 4.2.

When creating an identity according to the IdM life cycle, we allow users for choosing protection against tampering and identity theft through tamper-evident WebID profiles. In case, a user selects this option, Sociddea issues an identity $i = (w', T)$ consisting of a WebID URI, which accommodates the encoded hash value of the main public key, an unsigned WebID profile, which contains almost no personal data at this point in time, and a WebID certificate, which includes the WebID URI and the main public key.

Producing a tamper-evident WebID profile necessitates signing personal data worth protecting. In line with the process description for the client

²⁷Different identifiers would infer calculating different hash values from the same personal data stored in a WebID profile.

side in Subsection 6.2.1, we rely on a client-side signing tool at the moment. It assists human entities in transforming a signing request into a signed response, which is then sent back to the server side. Figure 6.3 depicts the GUI of this tool. Here, an identity owner first selects a WebID certificate. The tool verifies that the selected certificate contains a WebID URI that refers to a WebID profile and is prepared for tamper-evidentness, i.e., the URI involves a representation of a public key that is also available in the referred WebID profile. If this is the case and the selected WebID certificate is valid, the thus authenticated identity owner can trigger the automatic signing of personal data stored in the associated WebID profile. For signature creation, the tool employs the private key that corresponds to the public key stored in the selected WebID certificate. After verifying generated signatures found inside the signed response of the client, the



Figure 6.3: Client-Side Tool Support for Creating Tamper-Evident WebID Profiles

server side of the component automatically inserts all signatures into the WebID profile of the identity owner that triggered the signing process.

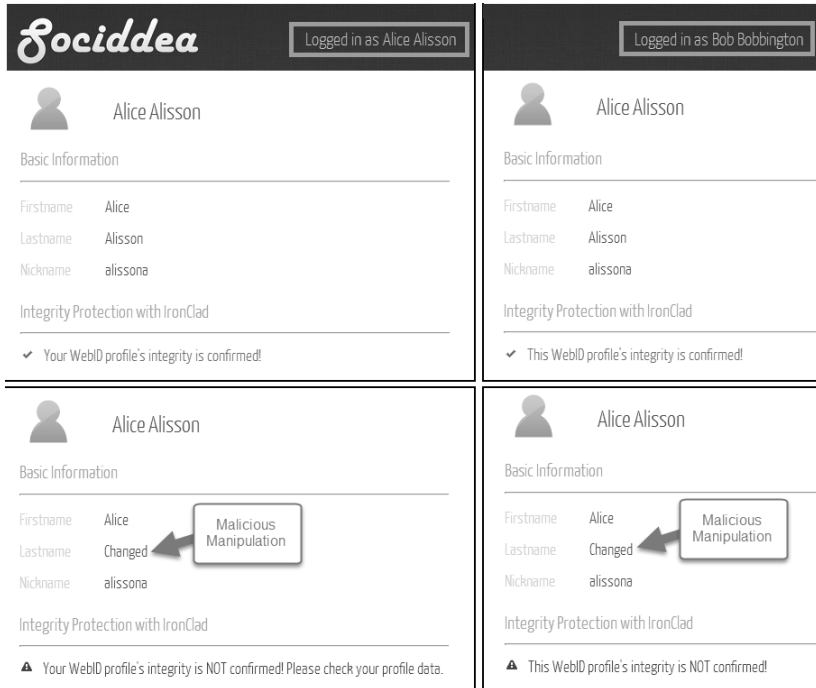


Figure 6.4: Results of Tamper-Evidentness Component in Sociddea (Top/Left: Successful Verification Shown to Identity Owner; Top/Right: Successful Verification Shown to Requester; Bottom/Left: Possible Tampering Indicated to Identity Owner; Bottom/Right: Possible Tampering Indicated to Requester) (Wild et al., 2014)

For verifying tamper-evident WebID profiles, we do not only implement tamper detection for checks during the authentication sequence that apply to diverse classes of entities, but also support human requesting entities in discovering anomalies in visual representation of personal data. Even though verification is primarily intended for SPs, the Sociddea platform as

an IdP also utilizes the component for tamper-evidentness. When human requesting entities take a look at tamper-evident WebID profiles managed using the Sociddea platform, they receive information on the integrity of personal data contained in such WebID profiles. Figure 6.4 illustrates four different scenarios: identity owner Alice’s view on her valid profile data (top/left), ‘Lastname’ changed - tampering detected (bottom/left), requesting entity Bob’s view on Alice’s valid profile data (top/right), Bob’s view on Alice’s tampered profile (bottom/right). While the figure visually highlights data manipulations caused by changing a personal data attribute of an identity owner, the component for tamper-evidentness also detects data integrity breaches caused by adding or removing RDF triples.

After completing development, the next section is about evaluation of TE.

6.3 Evaluation

For evaluating the component for tamper-evidentness, Subsection 6.3.1 first determines the characteristics to be assessed. Subsection 6.3.2 then describes the evaluation procedure that takes account of these characteristics, and, finally, Subsection 6.3.3 discusses the evaluation results.

6.3.1 Characteristics

By defining five requirements on approaches for tamper-evidentness in Subsection 6.1.2, we created the foundation for a systematic analysis of related work and a well-directed component development. Building on this foundation, we reapply the requirements—accessibility, applicability, integrability, non-impairment and public verifiability—as criteria for evaluating our work in the context of tamper-evidentness.

6.3.2 Procedure

Relying on the success indicators and verification sources declared in Subsection 2.3.4, we employ the proof-of-concept implementation (Wild et al., 2014; Wild et al., 2015) as well as unit and operational acceptance tests to evaluate the component with regard to the characteristics defined in Subsection 6.3.1. In order to assess the degree to which each evaluation criterion has been fulfilled by our contribution, the four-level rating system, known from Subsection 3.2.3, suits our purpose.

6.3.3 Results

On the basis of the evaluation results obtained by applying the procedure specified in Subsection 6.3.2, we discuss how the component for tamper-evidentness addresses each criterion defined in Subsection 6.3.1. Building upon the discussion of each evaluation criterion, we draw a conclusion involving a rating.

Accessibility Both common and tamper-evident WebID profiles share similar qualities, with attributes inside tamper-evident WebID profile constituting a superset of the attribute set contained in the same common WebID profile featuring no measures for tamper-detection. While tamper-evident WebID profiles consist of additional RDF-based signatures, existing personal data remains untouched. Here, signatures are loosely coupled statements about personal data. Removing all RDF triples relating to signatures would consequently reveal the original payload of a WebID profile. By refraining from encrypting personal data in WebID profiles, the component does not impair the accessibility of tamper-evident WebID profiles. Even though signatures and personal data contained in tamper-evident WebID profiles are accessible per se, view filters can assist in concealing particular RDF

triples, e.g., excluding all signatures might be beneficial for presenting personal data only. Such filtering, however, requires some modifications of the signing process, as detailed in Chapter 7.

To conclude, TE completely fulfills the accessibility criterion.

Applicability WebID profiles involve RDF for representing personal data in a machine-readable way, with appropriate RDF-based vocabularies enabling to describe and interlink new contents. This facilitates extending WebID profiles by additional RDF triples. It is consequently well applicable to represent and associate signatures to personal data attributes. Through directly operating on RDF graphs that represent WebID profiles, the component cannot only manage different orders and structures of RDF triples, but also diverse types of serialization. This allows for dealing with a high heterogeneity of vocabularies and attribute sets, and, thus, for covering identities from different types of entities. However, TE is only applicable to new identities because it requires appending cryptographic means to WebID URIs.

To conclude, TE partially fulfills the applicability criterion.

Integrability For accomplishing an extensive protection against malicious manipulation and identity theft through early detection, the component needs to be integrated into both IdPs and SPs. This is necessary for signature creation via IdP and for signature verification via SP. Yet, we simplify integrating the contribution for tamper-evidentness by a self-contained component hiding its complexity through offering an API. The component does not involve exotic dependencies for operation, but it requires access to an identity owner's WebID URI, WebID profile, and WebID certificate including the private key for signing hashes generated by general-purpose algorithms. In order to avoid disclosing the private key to a third party, we depend on a client-side signing tool, which entails requirements for system, platform and device support.

The backward-compatibility of the component is an advantage in terms of integrability. Despite comprising a key representation, a WebID URI referencing a tamper-evident WebID profile, as per our contribution, is still a valid WebID URI. SPs, which do not integrate the component, are therefore still enabled to perform the default WebID authentication of users, as specified in Subsection 4.1.2. Furthermore, the WebID verifier can request a tamper-evident WebID profile as usual and ignore included signatures without losing necessary personal data. In the opposite case, i.e., when a common (non-tamper-evident) WebID profile needs to be verified, the component detects this and falls back to the WebID authentication sequence as per proposed solution (cf. Figure 4.7).

To conclude, TE partially fulfills the integrability criterion.

Non-Impairment Combining the key representation—an encoded SHA-256-based hash value of the main public key—with the WebID URI creates a universal mean to detect tampering in WebID profiles. While other hash algorithms generate much shorter outputs, e.g., message-digest algorithm 5 (MD5) (22 characters) or SHA-1 (27 characters), they are either classified as insecure today or considered to be unsafe in the next years. In contrast, (Barker et al., 2012) estimate SHA-256 as secure until 2030.

Attaching cryptographic means to WebID URI comes at the cost of some implications. Losing or compromising the main public/private key pair requires creating a new WebID identity. That is, an identity owner would need to create a new main public/private key pair for signing the WebID profile. Due to the fact that the main public key is encoded inside the WebID URI, also the *new* main public key needs to be stored there. While changing a tamper-evident WebID URI results in creating a different and therefore new identity, we consider this as a common shortcoming of WebID. Not only is it impossible to transform common WebID URIs to tamper-evident ones, but also to change tamper-evident WebID URIs without invalidating all links that point to the underlying personal data, e.g., incoming connections

from social contacts expressing their relationships. Moreover, storing a hash value inside a WebID URI makes it difficult for human entities to memorize such identifiers (Halpin, 2014).

Unlike automatic data integrity verification, modifying a WebID profile demands that the identity owner recreates signatures for updated personal data using the main private. While not an issue for non-human entity, human entities need to be supported in this task. To ease signing for human entities, the component involves a client-side tool, which is, however, an additional mean users have to employ to enhance security.

To conclude, TE partially fulfills the non-impairment criterion.

Public Verifiability Verifying a tamper-evident WebID profile does not require human interaction, i.e., the component automatically verifies integrity of personal data contained in a WebID profile by checking the WebID URI and by validating stored signature. Although interpreting the results of the data integrity check rests on requesting entities as well as service providers, especially human entities can benefit from visual presentation of results, as exemplarily done by Sociddea (cf. Figure 6.4). Consider an attacker would remove signatures stored in a tamper-evident WebID profile, there is still the hash-included WebID URI publicly indicating that personal data inside the corresponding tamper-evident profile has to be integrity-protected. While there are other ways of using hash values in URIs (Sauermaun et al., 2007), the component just appends an encoded hash value to a common WebID URI, for the sake of simplicity and conformity.

Signing WebID profiles facilitates detecting malicious manipulation of contained personal data. When an aggressor changes personal data in another user's WebID profile, verifying the signature of this profile fails because the aggressor cannot sign the manipulated WebID profile with the main private key safeguarded by the actual identity owner. In case of implanting an own public key in another user's WebID profile, an aggressor could sign the WebID profile with the corresponding private key, but not change the

hash value of the public key in the WebID URI. Since we use algorithms considered as safe, it is unlikely to find a collision to the hash value encoded in a WebID URI or to create a private key from a given public key. Consequently, changing the hash value would change the identifier of an attacked user's WebID identity and, thus, create a new WebID identity rather than hijack the user's one. This protection also secures a WebID profile against temporary manipulation and temporary identity theft (cf. Scenarios 6.2 and 6.3).

To conclude, TE largely fulfills the public verifiability criterion.

By completing the component evaluation with results indicating an overall large fulfillment of the criteria, the following section sums up the outcome of TE.

6.4 Summary

With tamper-evident WebID profiles, requesting entities and identity owners are enabled to verify that stored personal data was not tampered, neither on systems that make profiles available nor during transmission. By providing a component to detect temporary and permanent integrity breaches in personal data caused by malicious manipulation and identity theft, we contributed to increase trustworthiness in self-deterministic IdM by human and non-human entities. Through tamper-evidentness, identity owners can verify that they are in control of the resources representing their WebID profiles and even manage their personal data on potentially untrusted or insecure systems, in case data disclosure would not be an issue. Based on three scenarios, we derived five requirements to systematically analyze related work and align the development of our contribution. Taking accessibility, applicability, integrability, non-impairment and public veri-

fiability into account, we conceptually, logically and physically designed the contribution for tamper-evidentness. Through focusing on empowering individuals instead of authorities, we enabled the verification without requiring prior knowledge, except for already known WebID identifiers. To prove the concept for tamper-evidentness, we transferred the design artifacts into a self-contained component, which we then exemplarily integrated into the Sociddea platform. From the results obtained by evaluating the component, we concluded an overall sufficient fulfillment of the requirements, but with some justifiable drawbacks. Security comes at a cost. Even though we are aware that tamper-evident WebID URIs and profiles complicate the management especially for human entities, we claim that most issues can be successfully addressed through utilizing techniques such as URI drag and drop, quick response (QR) codes or WebID URI embedded into other objects like WebID certificates.

Now that the component for tamper-evidentness has been described, the following chapter continues with detailing the third key component.

Fine-Grained Filtering

To particularly address Problem Cause 3.3: “Incomplete Range and Granularity of Access Control”, this chapter starts with analyzing requirements and related work specific to the problem domain in Section 7.1. With the obtained analysis results, Section 7.2 describes the development of the fine-grained filtering (FGF) component to completely meet the domain-specific requirements. Employing the success indicators and verification sources described as part of the strategy on page 61, Section 7.3 evaluates our approach to fine-grained filtering to verify the compliance with the requirements. Finally, Section 7.4 concludes this chapter by summarizing the outcome of our contribution for fine-grained filtering.

7.1 Analysis

With reference to Activity 3.3: “Increase Range and Granularity of Access Control”, Subsection 7.1.1 describes several scenarios to highlight the necessity for FGF as a recommended security enhancement of the proposed solution. Using these scenarios, Subsection 7.1.2 then derives a set of requirements that we employ to analyze related work, with results discussed in Subsection 7.1.3.

7.1.1 Scenarios

In addition to inherent features of the proposed solution, access control capabilities play a decisive role in enhancing security, as shown in the next three scenarios:

Scenario 7.1: Protection of Privacy. WebID identity owner Alice intends to restrict access to her personal data. She wants to do this because all data available inside her WebID profile could be easily retrieved, if not properly addressed by appropriate access control mechanisms. Sensitive personal data could be used for purposes she does not agree with, e.g., social network analysis, personalized advertisements or product marketing. Although restricting access to her entire profile would be an option, Alice is not interested in losing advantages like authenticating to new yet unknown services. To keep associated services up-to-date, Alice therefore wants to permit third-party entities, like other persons or web services, to monitor specific parts of her personal data for changes. Furthermore, Alice wants to allow anyone to access personal data she marked as public, even if Alice is currently unavailable or unauthenticated.

Scenario 7.2: Trust Relationships. Bob, who also maintains a WebID identity, wants to retrieve Alice’s current address data. Compared to the

anonymous subjects in Scenario 7.1, Alice knows and trusts Bob. She therefore granted him extended privileges some time ago. While Bob is allowed to see Alice's private address data as part of the personal data contained in her WebID profile, Alice does not want to share this kind of data with her co-worker Casey. Instead of private address data, only Alice's office address data should be visible to Casey.

Scenario 7.3: System Migration. Alice's WebID profile is hosted on a third-party server she trusted in the past. For justifiable reasons, she does not trust the server operator any longer and, thus, plans to switch the server hosting her WebID profile. Alice has distributed her WebID profile data to separate resources in order to apply access control at the resource level. For migrating to a new hosting server, Alice has to find, consolidate and transfer all her personal data being scattered among various resources. Additionally, she has to adjust access control lists (ACLs) for these resources due to issues like different hosting locations or naming conventions/restrictions. As an identity owner, Alice must be aware of all resources relevant to the migration. Depending on Alice's setup used for securing her personal data, this migration might be a complex undertaking.

The next subsection outlines the requirements inferred from the scenarios.

7.1.2 Requirements

By examining Scenarios 7.1 to 7.3, we derived accessibility, maintainability, portability, range and specificity as five essential requirements on approaches for FGF incl. underlying specifications, which we detail in the following:

Accessibility The requirement of accessibility, which we inferred for approaches to tamper-evidentness in Subsection 6.1.2, also applies for FGF.

WebID-based authentication requires an at least partially accessible public WebID profile because it contains an identity owner's public keys. Not only to ensure authenticatability of identity owners, but also to facilitate monitoring and retrieval of personal data by allowed SPs and by entities identity owners maintain relationships with (cf. Scenario 7.1), WebID profile data must remain accessible according to specified preferences. This furthermore implies refraining from encryption, as explained in Subsection 6.1.3.

Maintainability The preferences identity owners specify to control access to their personal data must be maintainable by human and machine (cf. Scenario 7.3). Relying on standard-compliant specifications and tools eases maintenance, enables reliability and avoids introducing too much overhead through adjustments, additional support or training efforts.

Portability Without making major adjustments, access control preferences must be portable to other systems by identity owners (cf. Scenario 7.3). Processors to interpret preferences and editors to declare them have to be either available or easy to implement. In line with *maintainability*, standard-compliant specifications and tools foster portability and are therefore a part of this requirement.

Range Identity owners must be enabled to express whom exactly they want to make certain personal data sets available to. Conforming with the preferences they have specified, access control of personal data contained in their WebID profiles must universally apply for anonymous requesting entities, groups of identities and specific identities of different entity classes, covering both human and non-human requesting entities (cf. Scenarios 7.1 and 7.2). That is, it has to be possible to treat each requesting entity authenticated via WebID differently, when attempting to access personal data inside a WebID profile.

Specificity In order to prevent unwanted retrieval, identity owners have to restrict access to certain (sensitive) personal data stored within their profiles (Scenario 7.2). They must therefore be enabled to precisely define

which parts of their WebID profile data has to be filtered out. Taking RDF graph representations of WebID profiles for granted (cf. Equation (4.2)), specifications must be expressive enough to enable fine-grained access control at the level of individual attributes, i.e., per RDF triple.

Having specified the requirements, we discuss related work next.

7.1.3 Related Work

While many proprietary implementations feature discretionary, role-based or mandatory access control, their underlying ACLs are neither semantically interpretable by machines nor universal in the sense of identifying the entities that request access as well as the resources to be protected (Wild and Gaedke, 2014). For this reason, we focus the discussion of related work on approaches involving RDF-based ACL specifications and means for identifying entities via URI.

As unprotected WebID profiles are potential information sources for known and wanted but also for unknown and unwanted requesting entities, identity owners can protect their personal data by defining access control rights at the level of (URI-addressable) resources (Hackett and Hawkey, 2012). While resource-based access control mechanisms allow for shielding from unwanted access, retrievals or tracking attempts (Bonneau et al., 2009), they typically offer only coarse-grained protection, without further intervention by identity owners.

With focus on resources rather than underlying data, identity owners that employ such mechanisms would need to outsource their personal data to separate resources and set proper permissions in order to enable a less coarse-grained protection. The number of resources required for a less coarse-grained protection increases with the extent

and heterogeneity of data to apply access control to, which, in turn, would cause a rise in complexity, and complicate modifications and transfers to other systems (Heitmann et al., 2010).

Being a vocabulary to define access rights, Web Access Control (WAC) enables discretionary access control and facilitates protecting URI-addressable resources against unauthorized access by anonymous requesters as well as subjects or groups, which are also identifiable by URIs (Hollenbach et al., 2009). ACLs specified by WAC are machine-readable through RDF and can be stored independently from the resources they protect. As described in (Chudnovskyy, Wild, et al., 2012), WAC is well-suited for scenarios involving many resources to control access to. Even though WAC does not support directly controlling access to specific data within resources, it is compatible with outsourcing particular personal data sets as self-contained resources. Such data distribution and related definition of corresponding ACLs comes along with declining maintainability and portability. For instance, a fine-grained control at its best would result in outsourcing each RDF triple within a WebID profile to a separate resource. When applying changes, this approach is impracticable.

The Access Control Ontology (ACO) and the User Access Ontology (UAO) are similar to WAC, but add support for roles and enable directly mapping permissions to HTTP verbs (Tomaszuk et al., 2011; Tomaszuk and Rybiński, 2011). Like WAC, ACO and UAO can only control access to resources. To protect data within resources, both approaches require outsourcing of relevant data to separate resources. ACO, UAO and WAC share the same maintainability and portability issues.

The *data perspective* approach customizes WebID profile data for specific identities of requesting entities by introducing sets of triples as alternative data sources (Tramp, Story, et al., 2012). It thus allows for manipulating

data represented by resources. For each combination of requested data, identifier and public key, a view is defined in terms of the set of triples to be returned. These view definitions increase flexibility by providing improved filter expressiveness, e.g., new triples can be directly added to a profile view. While the *data perspective* approach represents a promising work towards fine-grained filtering, it lacks maintainability. The approach distributes relevant data across view definitions and the actual WebID profile, which decreases maintainability as updates necessitate adjustments in several places. View definitions offer alternative information sources relative to existing WebID profile data, but require further processing to prioritize, merge or replace specific triples. If view definitions are used as an additional layer of information, the approach would store personal data in two different places, which causes a decrease in maintainability through redundancy and bears the risk of creating conflicts. Moreover, the approach does not support group-wise views and involves a custom vocabulary, which limits expressiveness and portability.

As related work does not sufficiently fulfill the requirements, the following section continues with presenting a distinct approach to fine-grained filtering.

7.2 Development

For developing the component for fine-grained filtering on the basis of (Wild, Chudnovskyy, et al., 2013a; Wild, Chudnovskyy, et al., 2013b; Wild et al., 2015), we adopt the design procedure that has been applied to model the artifacts of the solution in Section 4.1. That is, we involve three models, each representing a different state of the design. Using the conceptual model formalized in Subsection 7.2.1, we specify the logical model in Subsection 7.2.2. By deriving the physical model from the logical

model in Subsection 7.2.3, we establish the technical foundation for the implementation described in Subsection 7.2.4.

7.2.1 Conceptual Model

To avoid unwanted retrieval of personal data stored within WebID profiles, we enable identity owners to apply fine-grained filtering of sensitive personal data. Here, a WebID profile acts as filter input. Graph $G = (V, L) \sim T$ represents such profile, as formalized in (4.2). The graph-to-graph transformation function γ maps graph G onto graph $G' \in \mathfrak{G}$ depending on identity $i_r \in \mathfrak{I}$, as defined by Equation (7.1).

$$\gamma : \mathfrak{G} \times \mathfrak{I} \rightarrow \mathfrak{G} \quad (7.1)$$

Graph G' represents the WebID profile of an identity owner i filtered by sensitive data a requester denoted by identity $i_r \in \mathfrak{I}$ is *not* allowed for retrieving. Relying on Equation (7.1), Equation (7.2) formalizes such filtering.

$$\gamma(G, i_r) = G' = (V' \subseteq V, L' \subseteq L) \quad (7.2)$$

While all personal data is available in graph G , requester i_r is only granted the privilege to see a particular subset of the identity owner's data. Graph G' represents this particular subset. Filter function δ defines a mapping of triples on $\{0, 1\}$ depending on the identity. While “0” means sensitive data and, therefore, that the corresponding RDF triple is *not* present in graph G' , “1” means the opposite. Consequently, we can achieve blacklisting or whitelisting of sensitive WebID profile data for particular requesters using filter function δ , as defined by (7.3).

$$\delta : \mathfrak{I} \times \{t\} \rightarrow \{0, 1\}, \forall t \in T \quad (7.3)$$

The graph-to-graph transformation $\gamma(G, i_r)$ uses δ_{i_r} to create a filtered graph G' based on G for a requester denoted by identity i_r . For an identity owner i acting as a requester i_r , function δ_i yields “1” for each RDF triple in graph G , i.e., graph $G' = G$. RDF triples $T' \subseteq T$ span graph $G' = (V', L')$, $T' \sim G'$, as defined in Equation (7.4).

$$T' = \{t | \delta_{i_r}(t) = 1, t \in T\} \quad (7.4)$$

To relieve identity owner i from the need to define filter function δ_{i_r} for each potential requester, we introduce fallback function $\epsilon(i_r)$ that yields the best possible fallback identity $i_f \in \mathcal{I}$ for a given requester denoted by identity i_r . In order to facilitate grouping possible requesters, we introduce the following four identity sets:

- *requesters* authenticated using WebID: $I_z \subseteq \mathcal{I}$,
- *specific requesters* defined by the identity owner: $I_s \subseteq I_z$,
- requesters who are *friends* of the identity owner: $I_p \subseteq I_z$, and
- *anonymous requesters*: $I_o \subseteq \mathcal{I}, I_o \cap I_z = \emptyset$.

A fallback identity yielded by $\epsilon(i_r)$ is equatable to diverse identities as a function of both group membership and filters created by identity owner i . That is, if identity owner i created a filter for a requester denoted by identity i_r , then the fallback identity matches the identity of this particular requester. In case there is no such filter defined, but the identity owner is connected with the requester ($i_r \in I_p$) and has created a filter for friends, then the fallback identity matches the friend identity $i_p \in I_p$. When a requester is authenticated yet not known to the identity owner who has created a filter for authenticated users, then the fallback identity matches the identity of an authenticated user $i_z \in I_z$. With the identity owner having created a filter for anonymous users, unauthenticated requesters are assigned with a fallback identity matching the identity of an anonymous user $i_o \in I_o$. Finally, for dealing with the case of the identity owner not having any filter defined, the fallback identity corresponds

to a theoretical *null* identity $i_{\text{null}} \in \mathcal{I}$. As a consequence, we can also refine the set of possible fallback identities to $i_f \in (\{i_p, i_z, i_o, i_{\text{null}}\} \cup I_s)$. Equation (7.5) formalizes fallback function $\epsilon(i_r)$.

$$\epsilon(i_r) = i_f = \begin{cases} i_r & \text{if } \exists \delta_{i_r} \\ i_p & \text{if } \exists \delta_{i_p} \wedge i_r \in I_p \wedge i_r \notin I_s \\ i_z & \text{if } \exists \delta_{i_z} \wedge i_r \in I_z \wedge i_r \notin I_s \wedge i_r \notin I_p \\ i_o & \text{if } \exists \delta_{i_o} \wedge i_r \in I_o \\ i_{\text{null}} & \text{if } \nexists \delta_{i_r} \wedge \nexists \delta_{i_p} \wedge \nexists \delta_{i_z} \wedge \nexists \delta_{i_o} \end{cases} \quad (7.5)$$

Here, filter function $\delta_{i_{\text{null}}}$ implements a behavior as if no filtering is active and, thus, enables accessing profiles not having predefined filters (cf. Equation (7.6)). To make creating filters a requirement for identity owners, we could revise $\delta_{i_{\text{null}}}$ to result in $0 \forall t \in T$.

$$\delta_{i_{\text{null}}}(t) = 1 \forall t \in T \quad (7.6)$$

To use $\epsilon(i_r)$ as part of $\gamma(G, i_r)$, we adjust Equation (7.4) as shown in Equation (7.7).

$$\gamma(G, \epsilon(i_r)) = G' \sim T' = \{t | \delta_{\epsilon(i_r)}(t) = 1, t \in T\} \quad (7.7)$$

Representing the certain views an identity owner i can create for a WebID profile T depending on a requester denoted by identity i_r , $\mathfrak{G}_i \in \mathfrak{G}$ is the set of all filtered graphs for this particular profile, as formalized in Equation (7.8).

$$\mathfrak{G}_i = \{G' | G' = \gamma(G, i_f), i_f \in (\{i_p, i_z, i_o, i_{\text{null}}\} \cup I_s)\} \quad (7.8)$$

Adjustments to tamper-evidentness. Filtered graphs entail adjustments to the tamper-evidentness model presented in Subsection 6.2.1. Since this

conceptual model for tamper-evidentness relies on certain operations on RDF graph representations of WebID profiles, each *filtered* representation creates a new graph to be signed. The procedure we applied for signing so-called MSGs, as per (Tummarello et al., 2005), does no longer fit to our approach because the filters could exclude some blank nodes of a MSG. So, an agent verifying the data integrity of a WebID profile could receive a filtered representation of that profile, where some blank nodes of a MSG are missing. Verifying the signature of the filtered profile would fail because it was generated with blank nodes inside the RDF graph (Tummarello et al., 2005). Nevertheless, contained personal data still originates from the actual identity owner and not an aggressor.

As a consequence, the actual signing has to happen for each filtered graph out of \mathfrak{O}_i , which has been specified by an identity owner. When an identity owner modifies personal data, each existing filter needs to be applied and each resulting filtered graph $G' \in \mathfrak{O}_i$ has to be signed once again, as formalized in Equation (7.8). Adapting Equation (6.1), we therefore transform filtered graph G' into its canonical representation \tilde{G}' . Adapting Equation (6.2), we use the equivalent canonical representation of the filtered RDF triple set $\tilde{T}' \sim \tilde{G}'$ instead of \tilde{T} . According to Equation (6.3), the hash value of an entire filtered representation of a WebID profile and the main private key k'^{-1} of the identity owner are used to create signature m_{sig} of a filtered WebID profile. Finally, we attach each signature m_{sig} to its corresponding filter so that the original WebID profile remains unchanged and the existing signatures of other filters are still valid.

Building on this conceptual model for fine-grained filtering, the next subsection proceeds with specifying the logical flow.

7.2.2 Logical Model

To orchestrate the elements of the conceptual model in their primary usage sequence, we derive a logical model, as illustrated in Figure 7.1. When a requesting entity identifiable through i_r tries to retrieve personal data from the WebID profile of identity owner i , an appropriate filter is searched for using $\epsilon(i_r)$. To protect sensitive data, identity owner i has to specify eligible filters prior to this step. Filters are stored as filter specifications in the identity owner's WebID profile. Filter specifications are hidden from anyone but identity owner i . Otherwise, this information is a potential subject to social engineering, e.g., profile analyzers could conclude group affiliations utilizing knowledge about δ_{i_r} or $\epsilon(i_r)$. Each filter specification consists of fallback identity i_f and filter γ . Having detected a specification for i_f using $\epsilon(i_r)$, the filter $\gamma(G, i_f)$ converts graph G into graph G' that represents a WebID profile filtered by data marked as sensitive by i . That is, the profile retrieved by requester i_r contains only data which satisfies the constraints defined by filter function $\delta_{\epsilon(i_r)}$.

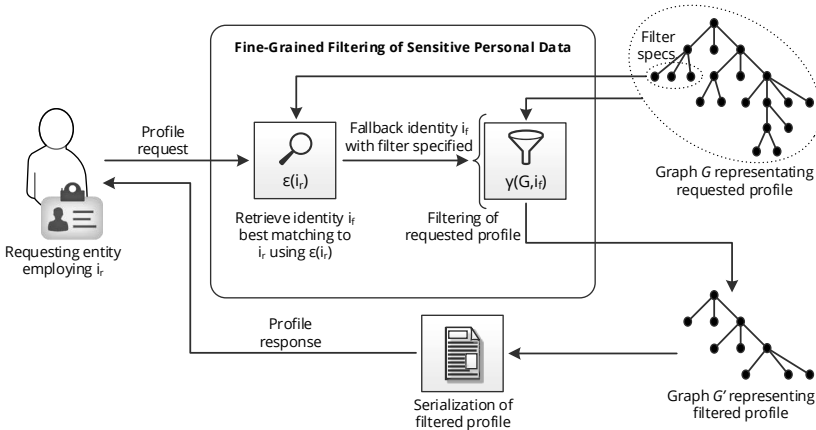


Figure 7.1: Fine-Grained Filtering Process Model

Having created the logical model from the conceptual one, the next subsection continues with detailing technical aspects of FGF.

7.2.3 Physical Model

In order to establish the basis for the technical implementation of the design artifacts, this physical model further specifies the logical model.

The approach for fine-grained filtering makes use of the semantic WebID Profile Filter Language (WPFL) (Wild, Chudnovskyy, et al., 2013a) and the SPARQL CONSTRUCT query form (Harris and Seaborne, 2013) for technically representing the transformation and filter function. WPFL allows for defining filter specifications that involve three basic elements: *entity name* for i_f , *filter command* for $\gamma(G, i_f)$ and a *specification element* to bind them together and connect the filter to the WebID profile. The specification element allows storing filter specifications either in an identity owner's WebID profile, represented by graph G , or separately as linked resources. Only three RDF triples describe the basic elements of WPFL, as exemplarily shown in Turtle syntax in Listing 7.1.

Listing 7.1: Template of Filter Specification as per WPFL

```
1 <WEBID URI> filter:specification [  
2   filter:entity ENTITY;  
3   filter:command COMMAND  
4 ] .
```

The SPARQL CONSTRUCT query form facilitates constructing a new graph G' based on an existing graph G , as required by Equation (7.2). According to Equation (7.4), it can include or exclude personal data during construction of $G' \sim T'$. A whitelisting²⁸, as defined by this equation, men-

²⁸Blacklisting data is also supported by SPARQL CONSTRUCT queries via MINUS statements.

tioning all personal data to be available in graph G' is described by the generic filter command shown in Listing 7.2.

Listing 7.2: Generic Filter Command Specification Using SPARQL CONSTRUCT Query Form

```
1 CONSTRUCT { ?s ?p ?o } FROM <WEBID URI> WHERE { ?s ?p ?o .  
2   FILTER(?s in (Subject1, Subject2, [...])) .  
3   FILTER(?p in (Predicate1, Predicate2, [...])) .  
4   FILTER(?o in (Object1, Object2, [...]))  
5 }
```

As an example, all contact references would be copied from G to G' , if solely the `foaf:knows` predicate is mentioned (cf. line 3). To increase filtering granularity, it is beneficial to also refer to subjects or objects of RDF triples (cf. lines 2 and 4), e.g., in order to include/exclude specific contacts. While this all together defines exactly one *filter directive*, the UNION keyword in SPARQL enables to employ several filter directives in one filter command.

To cover filtering of context-dependent personal data, we utilize SPARQL Property Path, as specified in (Harris and Seaborne, 2013). For instance, a street attribute could be context-dependent as it is element of an address, which in turn could be element of either private or business contact data. Property paths facilitate to address relevant elements in graph G by specifying the routes between them. For example, a filter command to construct a new graph by including name and image of identity owner i as well as city and country of the owner's home—but not street, postal code etc.—is described in Listing 7.3. Here, lines 3 and 4 create the context needed to include city and country (cf. lines 5 and 6) as part of the address data described using the contact ontology (Berners-Lee, 2001).

To select the best-matching available filter specification based on the retrieved filter entity, as formalized in Equation (7.5), the ap-

Listing 7.3: Exemplary SPARQL CONSTRUCT Query with Property Paths

```
1 CONSTRUCT { ?s ?p ?o } FROM <WEBID URI> WHERE {  
2   {?s ?p ?o . FILTER(?p in (foaf:name, foaf:img))} UNION  
3   {?s ?p ?o . ?t con:home ?o} UNION  
4   {?s ?p ?o . ?t con:home/con:address ?o} UNION  
5   {?s ?p ?o . ?t con:home/con:address/con:city ?o} UNION  
6   {?s ?p ?o . ?t con:home/con:address/con:country ?o}  
7 }
```

proach relies on a dedicated SPARQL query that uses the identifier possibly provided by the requesting entity.

Now that all artifacts relating to FGF have been described, the following subsection outlines the implementation of the design.

7.2.4 Implementation

To put the conceptual, logical and physical model into practice, we consolidated all design artifacts in a self-contained component and exemplarily integrated it into the Socidea platform, which has been initially introduced in Section 4.2.

The contribution for fine-grained filtering enables identity owners to create filters on their WebID profile data not only for anonymous and specific requesting entities, but also for groups of requesting entities, with group affiliations predefined through roles or declared manually. For assisting human entities, like identity owners or web service administrators, in creating and configuring filters for profile data, Socidea provides them with an appropriate GUI accessible via the common profile authoring mode (cf. Figure 4.10). Here, they can use all identity attributes presented in the profile authoring mode to specify filters, i.e., each personal data at-

tribute can be marked as either visible or hidden. Using the GUI shown in Figure 7.2 on the left side, Sociddea allows for predicate-based filtering, e.g., by first name or by service endpoint. By selecting a known requesting entity, represented by a particular identity, role or group affiliation, Sociddea can visualize the preferences that have been previously defined in an already existing filter specification.

```
<rdf:RDF [...]>
  <foaf:Person rdf:about="https://vsr-
demo.informatik.tu-
chemnitz.de/sociddea/profiles/alice#aa">
  <filter:specification>
  <filter:entity>anonym</filter:entity>
  <filter:command>CONSTRUCT [...>
  </filter:command>
  </filter:specification>
  [...]
</foaf:Person>
</rdf:RDF>
```

Filter Specification

**Creation of
Filter Specification**

```
CONSTRUCT { ?s ?p ?o }
FROM <https://vsr-demo.informatik.tu-
chemnitz.de/sociddea/profiles/alice#aa>
WHERE {
  ?s ?p ?o
  FILTER(?p in (
    foaf:name,
    foaf:img,
    foaf:homepage,
    [...]
  ))
}
```

Detailed View on Value of filter:command

Figure 7.2: Creation of Filter Specification Based on User Selection (Wild et al., 2015)

To enable machines to semantically process this yet informal filter configuration, the component for fine-grained filtering automatically creates a SPARQL CONSTRUCT statement corresponding to specified preferences. As the component implements a whitelisting of attributes, the resulting SPARQL statement contains references to all personal data attributes declared as visible for a particular requesting entity. All three RDF triples,

which denote filter specifics, are directly stored within the WebID profile they are intended to be applied to. Figure 7.2 schematically depicts the process of creating such personal data filter.

Beyond supporting rather unskilled users by a GUI, more experienced users can employ an advanced profile editor offered by Sociddea in order to express more complex customized views on personal data through creating SPARQL CONSTRUCT statements manually. Not only does this allow for filtering even personal data attributes unsupported by the GUI, but also for maxing out the full potential of SPARQL, like dealing with special cases including conditional filtering.

Once a filter specification has been created, our implemented contribution considers it automatically during all future attempts to access the particular profile containing the data to be filtered. When a requesting entities tries to retrieve WebID profile data, the component searches for an appropriate filter specification using the possibly provided identity and the `filter:entity` triples in the WebID profile. Having found a matching filter entity, the component extracts the `filter:command` triple belonging to the same `filter:specification` and directly passes it to a SPARQL processor, i.e., no modification is made to the command. While Sociddea renders results produced by the SPARQL processor as defined in the request of a requesting entity (cf. Section 4.2), rendering per se is not part of the component for fine-grained filtering. Figure 7.3 exemplifies the filtering of a WebID profile in Sociddea for an anonymous requesting entity. Here, the component detects and employs a previously defined filter specification to create a customized view, which is then rendered according to the requesting entity's preferences as an HTML representation.

After completing development, the next section is about evaluation of FGF.

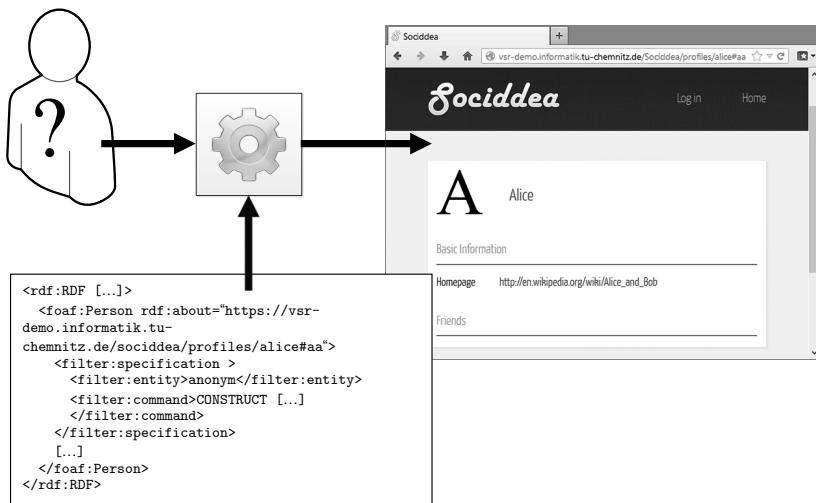


Figure 7.3: WebID Profile Data Filtered for Anonymous Requesting Entities (Wild et al., 2015)

7.3 Evaluation

For evaluating the component for fine-grained filtering, Subsection 7.3.1 first determines the characteristics to be assessed. Subsection 7.3.2 then describes the evaluation procedure that takes account of these characteristics, and, finally, Subsection 7.3.3 discusses the evaluation results.

7.3.1 Characteristics

By defining five requirements on approaches for fine-grained filtering in Subsection 7.1.2, we created the foundation for a systematic analysis of related work and a well-directed component development. Building on

this foundation, we reapply the requirements—accessibility, maintainability, portability, range and specificity—as criteria for evaluating our work in the context of fine-grained filtering.

7.3.2 Procedure

Relying on the success indicators and verification sources declared in Subsection 2.3.4, we employ the proof-of-concept implementation as well as unit and operational acceptance tests to evaluate the component with regard to the characteristics defined in Subsection 7.3.1. In order to assess the degree to which each evaluation criterion has been fulfilled by our contribution, the four-level rating system, known from Subsection 3.2.3, suits our purpose.

7.3.3 Results

On the basis of the evaluation results obtained by applying the procedure specified in Subsection 7.3.2, we discuss how the component for fine-grained filtering addresses each criterion defined in Subsection 7.3.1. Building upon the discussion of each evaluation criterion, we draw a conclusion involving a rating.

Accessibility Personal data stored in WebID profiles remains fully accessible for authorized requesting entities. The component for fine-grained filtering does not involve encryption, yet it facilitates hiding particular personal data sets, which the actual data owner considered as too sensitive for public disclosure. As a consequence, the component can also conceal filter specifications from requesting entities, so that these entities remain unaware of having possibly received just a constrained view on the identity owner’s complete personal data set.

To conclude, FGF completely fulfills the accessibility criterion.

Maintainability Contrary to related work (cf. Subsection 7.1.3), our contribution does not require outsourcing personal data to separate resources for implementing a fine-grained access control. Although the component allows for separating personal data and filter specifications, we recommend storing the latter in the WebID profile to be filtered. Thus, all necessary information can remain at one place, which simplifies updating, replacing or removing already existing filter specifications. With a minimal overhead of three additional RDF triples, we enable to define a specification for fine-grained filtering with regard to a certain requesting entity. Nevertheless, introducing a new vocabulary, like WPFL, reduces maintainability as it requires interpretation support. Depending on extent and specificity of fine-grained filters, the underlying commands may get complex and hard to maintain by human entities. Assistance by a GUI can mitigate this issue, as shown in Sociddea.

For the sake of homogeneity to the underlying profile data representation as RDF triples, the component demands using SPARQL (Subsection 3.3.3). This allows for directly passing a WPFL-based filter command to a SPARQL processor to create a new filtered graph. SPARQL is a standardized and widely adopted language with extensive tool support by optimized processors (Pérez et al., 2009). For operation, the component depends on only two input parameters: the graph representing an unfiltered WebID profile incl. a filter specification and the identity of the requesting entity. Using SPARQL rather than an own proprietary language ensures reliable processing and maintainability.

To conclude, FGF largely fulfills the maintainability criterion.

Portability In many cases filtering mechanisms depend on proprietary interpreters to apply access control (cf. Subsection 7.1.3). In contrast, our component features SPARQL with adequate processors existing for many platforms and architectures. This contributes to interoperability of the component and, thus, to a reasonable portability of filter specifications.

The effort to transfer specified filters to new systems is furthermore reduced by enabling identity owners to store necessary filter specifications within their WebID profiles. Despite these advantages, systems that utilize our contribution have to be capable of interpreting WPFL-based specifications and processing SPARQL queries.

To conclude, FGF largely fulfills the portability criterion.

Range Related work tries to reduce complexity by defining restricted vocabularies, whereas SPARQL allows for expressing complex queries. Restricted vocabularies offer advantages in terms of usability, but they also limit the possibilities of filtering and cause workarounds, like the necessity of outsourcing sensitive user profile data. To create a customized view on a WebID profile, the component automatically selects a filter specified for a requesting entity. If no filter specification is available, a fallback mechanism selects the most appropriate filter based on availability and provided identity. Not only is the component capable of handling anonymous and specific requesting entities that employ identities but also groups and simple roles like known contacts.

To conclude, FGF completely fulfills the range criterion.

Specificity While the component for fine-grained filtering focuses on selecting the most specific filter according to the identity supplied by a requesting entity, it also manages situations where a specific filter is unavailable. There, the procedure falls back to an available, more unspecific filter which matches at least some characteristics of the identity of the requesting entity. In contrast to other mechanisms, our contribution enables *real* fine-grained filtering at the level of attributes and beyond. Not only do we facilitate filtering of specific RDF triples representing personal data attributes, but also individual elements of RDF triples, i.e., subject, predicate and object. In addition, our contribution allows for modeling specific contextual dependencies among RDF triples to be filtered, like address data, through SPARQL queries involving property paths. The component for fine-grained

filtering can handle both whitelisting and blacklisting of RDF triples, but we recommend whitelisting because exposed filters do not contain any information on hidden personal data. Moreover, whitelisting eases constructing an empty or sparse graph representation of a profile, which might be relevant for identity owners having stringent requirements for privacy and, thus, want to forbid anonymous profile requests. Utilizing whitelisting, identity owners must actively unlock personal data attributes to be accessible for requesting entities, which also contributes to protection.

To conclude, FGF completely fulfills the specificity criterion.

By finishing the component evaluation with results indicating an overall complete fulfillment of the criteria, the following section sums up the outcome of FGF.

7.4 Summary

With customized views on WebID profiles, identity owners are enabled to keep control about amount and nature of personal data being presented to requesting entities. By providing a component to filter sensitive data in a fine-grained manner, we contributed to ensure privacy of identity owners. Through fine-grained filtering, we can mitigate data disclosure attempts of unwanted parties and facilitate migration to potentially more trustworthy systems. Based on three scenarios, we derived five requirements to systematically analyze related work and align the development of our contribution. Taking accessibility, maintainability, portability, range and specificity into account, we conceptually, logically and physically designed the contribution for fine-grained filtering. To cover almost all scenarios of hiding and showing specifics within WebID profiles, we utilized SPARQL and defined a small semantic filter vocabulary. Through whitelisting WebID

profile data per requester or group, we can exclude all filter specifications and make only particular non-sensitive personal data sets available. A fallback mechanism integrated into the component allows for automatically selecting the best-matching filter depending on the identity of a requesting entity. To prove the concept for fine-grained filtering, we transferred the design artifacts into a self-contained component, which we then exemplarily integrated into the Sociddea platform. From the results obtained by evaluating the component, we concluded an overall complete fulfillment of the requirements, only with minor drawbacks in terms of portability and maintainability of complex filter setups.

Now that the component for fine-grained filtering has been described, the following chapter continues with evaluating the proposed solution as a whole.

Overall Evaluation

8

To holistically evaluate the work done in Chapters 4 to 7 towards enhancing the security in managing personal data by web systems, this chapter first describes the characteristics to be validated in Section 8.1. On the basis of the characteristics, Section 8.2 then shows the consideration of an eligible procedure in order to specify relevant evaluation prerequisites including environment, method and setup. Having applied this procedure on the proposed solution including the components for context-aware control, tamper-evidentness and fine-grained filtering, Section 8.3 discusses obtained results. Finally, Section 8.4 briefly summarizes the outcome of the overall evaluation.

8.1 Characteristics

By considering this dissertation as a project directed toward the purpose defined in Section 1.4, we could utilize LFA for a systematic investigation of the challenges in Chapter 2. There, we derived a hierarchy of objectives from the results obtained by analyzing the prevalent problems in greater detail. Beyond identification and formulation of objectives, LFA also assists in assessing the degree of objective achievement during evaluation (cf. Subsection 2.1.1, Figure 2.1). For this reason, we employ the objectives described in Section 2.3 as criteria for the overall evaluation. As Research Questions 1 to 5 operationalize the purpose, we associate them with this particular objective.

Having specified the characteristics to be assessed, the next section determines a suitable evaluation procedure that takes them into account.

8.2 Procedure

With the objectives of Section 2.3 providing “a summary record of what was planned” (Miklić, 2008) and, therefore, the set of evaluation criteria, it is the next logical step to assess how well this plan has been put into practice by employing an eligible procedure on that basis. While such assessment is typically conducted as an independent examination by external collaborators (NORAD, 1999), this overall evaluation is solely to be treated as a self check made available as an essential part of the dissertation project. This procedure then allows for rendering external expert opinions that potentially review the findings acquired by the self check. “To provide decision makers with sufficient information [for making] an informed judgment about the past performance of the project, to

document lessons learned and to provide practical recommendations for follow-up action” (EC, 2004), we conduct an objectives-based study in consideration of the characteristics defined in Section 8.1, where inferred recommendations are outlined in the following chapter.

An objectives-based study is not only the most prevalent approach in program evaluation, but it is also suitable to be performed internally by engineers or program leads (Stufflebeam, 2001). Furthermore, such approach to evaluation is “especially applicable in assessing tightly focused projects that have clear, supportable objectives” and can be “strengthened by judging project objectives against the intended beneficiaries’ assessed needs, searching for side effects, and studying the process as well as the outcomes” (Stufflebeam, 2001). By setting the usual purpose toward verifying that the objectives of a project have been reached (Stufflebeam, 2001), the objectives-based study consequently fits well to this dissertation project.

As the “methods used in objectives-based studies essentially involve specifying operational objectives and collecting and analyzing pertinent information to determine how well each objective was achieved” (Stufflebeam, 2001), it is adequate to rely on already available objective specifications in terms of activities, results, purpose and overall objective. With regard to analyzing collected information, it is furthermore appropriate to utilize key assumptions, success indicators and verification sources as significant elements of the evaluation framework (Miklič, 2008). Reapplying the bottom-up strategy of Subsection 2.3.4 then enables to systematically conduct the evaluation. In line with Figure 2.12, it verifies that 1) all activities have been carried out, 2) all results have been delivered, 3) the purpose of this work has been achieved, and 4) the work contributed to attain the overall objective. Although the latter²⁹ is not profoundly verifiable within the

²⁹The achievement of the overall objective represents a joint long-run effort outside the direct control of an individual project.

scope of this dissertation, we provide indications for estimating the impact of our contributions. For each objective shown in Figure 2.12, we verify that assumptions hold true and success indicators are positive in addition to summarizing the actual contribution for achieving this particular objective.

In order to perform the verification as specified, the evaluation makes use of the sources declared in Section 2.3. On the basis of the Sociddea platform and the three components for enhanced security in managing personal data, we are enabled to incorporate the findings obtained from a variety of tests including integration tests, operational acceptance tests and proof-of-concept implementations. With this evaluation environment in place, it requires only a small setup consisting of Sociddea being equipped with some managed identities and integrating the components for context-aware control, tamper-evidentness and fine-grained filtering.

After outlining the evaluation procedure, the next section discusses obtained results.

8.3 Results

According to the procedure outlined before, we report on the findings in a bottom-up way, beginning with the execution of activities in Subsection 8.3.1, over the delivery of results in Subsection 8.3.2, the achievement of the purpose in Subsection 8.3.3 and the contribution to the overall objective in Subsection 8.3.4. Using these reports of objective attainment, Subsection 8.3.5 briefly puts our contributions in relation to the state-of-the-art technologies analyzed in Chapter 3.

8.3.1 Execution of Activities

The following evaluation results outline how well this work addressed the secondary causes of the central problem by executing five consolidated activities:

Activity 1.1: Extend Means for Modeling Secure Web Systems

To carry out this objective, our contribution built upon the solid foundation established through CBWE. Focusing on fast composition rather than complex a priori modeling, CBWE is not only in line with agile and lean development methodology, but facilitates the systematic development, maintenance and evolution through involving sets of reusable building blocks. For addressing deficits of CBWE approaches in terms of interpretability and interoperability at the individual and composite level (cf. Subsection 3.3.1), this work contributed a dedicated process model that specifies an IdM life cycle for compositions of web systems and underlying components, like web applications and web services. While it is a common practice to support identification and description of users through identities, our proposal towards self-deterministic IdM extended this practice by forcing the use of identities for adequately representing *all relevant* entities existing in web systems as an essential element to increase the overall quality and security. By employing a semantically-enriched variant of WAM and WebID, this work enabled semantic description, universal identification and linkage for compositions of web systems and individual components in a holistic way (Wild and Gaedke, 2014). It utilizes semantic vocabularies, like WSDL 2.0 RDF mapping, for modeling underlying components at design time and modify models during runtime. For carrying out this activity, we furthermore contributed a web-accessible WAM diagramming tool (Scholtz et al., 2015a), which supports web engineers in describing compositions of web systems, and dedicated input masks, which ease specifying components, like web services (Braune et al., 2014; Wild

and Gaedke, 2014). On the basis of universal identification and semantic description of identities from entities of a web system, our contribution allows for authentication via WebID-TLS and for both resource-based and fine-grained access control with WAC and FGF (Wild, Chudnovskyy, et al., 2013b). Moreover, we outlined in (Ast et al., 2013; Ast et al., 2014) how to consolidate also security-related parts of web systems to relieve web engineers from modeling and implementing them redundantly.

By assuming that companies are willing to use such extended means for modeling web systems, this contribution possibly interferes with existing technologies and processes they already employ. While relying on standard-compliant technologies and enabling a homogeneous way of modeling heterogeneous web systems and involved entities, it also requires additional efforts for making use of the means provided.

Despite the overall sufficient contribution to address Problem Cause 1.1: “Lack of Means for Modeling Secure Web Systems”, there is further work required, especially in terms of discovery and selection of eligible components, and the automatic code generation.

Activity 2.1: Offer Alternative to Customer Lock-in

To carry out this objective, our contribution reinforced a self-deterministic IdM that empowers individual identity owners and not authorities. Compliant with user-centric IdM, identity owners are put in control of their personal data directly or through sufficiently trusted entities. To support entities in managing their identities and associated personal data, this work contributed an IdM life cycle that is applicable to entities of arbitrary type. Identity owners keep *real* ownership of their personal data and can grant access to interested parties, like service providers or individual entities, without distributing their data among a variety of SPs. That is, interested parties may still access personal data of users, yet to the conditions of the

corresponding real data owners and not to those of SPs. With identity owners having to maintain a largely reduced set of identities, it is easier to keep associated personal data up-to-date for their own benefit as well as the benefit of companies, which built their business model on exploitation of social capital. Through reclaiming ownership and consolidating personal data, identity owners are no longer dependent on varying SP-centric IdM facilities and potentially offered migration options.

Recalling Scenario 2.1: “Online Shopping” and 2.2: “Password Trouble”, entities like Alice can easily make use of different service providers without having to enter personal data over and over again. By relying on one or a small set of identities representing her in several contexts, Alice is not only relieved from maintaining multiple copies of her data with the management facilities provided by different SPs, but also from creating and remembering complex passwords. So, changing her data or updating her credential at a single place would have an immediate effect for all parties involved.

On the basis of WebID, our proposition not only includes an increased and semantically enriched descriptiveness of identity attributes but also universal identification, recognition, discovery and linkage of identities representing entities of arbitrary class. This paved the way for a unified management, use and access of identities and associated personal data by both human and machine across web systems and independent from application-, domain- or usage-specific limitations. Along with facilitating reusability and exchangeability of UGCs through allowing for consolidation and ownership, this also contributed to an open social web including open social networking.

By assuming that persons are willing to regain ownership and self-manage their personal data, this contribution might interfere with old habits and inadequate technical skills of potential identity owners, yet this work proposes a set of tools (cf. Sociddea) for supporting them in respect of IdM.

Consolidation of personal data bears the risk of creating a single point of failure. Identity owners would therefore have to ensure the reliability and protection of their data and underlying system. Depending on their needs, however, they have to do this for only one or few systems and no longer need to worry about reliability, protection and use of their personal data by different third-party SPs, which also contributes to more transparency for identity owners. Building upon WebID might furthermore hinder adaption by persons because they would need to turn away from today's widely adopted knowledge-based authentication in favor of ownership-based authentication. As especially knowledge-based authentication is, however, increasingly challenged today (cf. Section 1.2) and will be further challenged in the future due to utilization of advanced computational capabilities for malicious purposes, ownership-based authentication represents a capable alternative. Not only does it enable enhanced cryptographic strength owing to more entropy compared to usual passwords, but it is also applicable for non-human entities and relieves human entities from creating and remembering complex passwords (cf. Subsection 3.3.2). These developments towards more self-determined control about personal data for individual persons are also in line with the endeavors of governments to support established legal frameworks and legally enforceable rights of citizens.

By furthermore assuming that companies are willing to open up for universal identification, linkage and discovery across web systems and to settle with just obtaining access to personal data and user-generated contents, this contribution possibly interferes with current business models that involve SP-centric management and storage of UGC and personal data. Yet, companies may still maintain access to consolidated, domain-independent, extensive and probably more up-to-date personal data of users, in addition to having the opportunity of analyzing the behavior of users that employ provided services. Without the need for providing facilities for managing,

storing, migrating and protecting identities and related data, companies can focus on offering rich and inviting feature sets to attract more users.

Despite the overall sufficient contribution to address Problem Cause 2.1: “Customer and Data Lock-in”, including 2.1.1: “Restricted Scope of Identity Data”, 2.1.2: “No Universal Identification, Linkage and Discovery of Identities” and 2.1.3: “Lack of Reusability, Openness and Exchangeability of UGCs”, there is further work required, especially in terms of governments urging companies, which are not yet convinced by the benefits described above, to open up for a more self-deterministic IdM of individual persons by providing means for ownership-based authentication and for making use of personal data stored in distributed user profiles.

Activity 3.1: Improve Control of Identity Based on Individual Context

To carry out this objective, our contribution enabled both scope-aware delegations and context-aware creation of credentials on the basis of self-deterministic IdM. Rather than relying on third-party asserted identities, where SPs predetermine associated contexts and restricted attribute sets, self-deterministic IdM allows identity owners for employing an open, extensible and more descriptive set of attributes across different application fields. Without requiring multiple identities to cover different scenarios and application fields, this work put entities into the position to use few consolidated identities combined with customized views on personal data in order to respect their specific privacy needs (Wild, Chudnovskyy, et al., 2013a). To furthermore assist in creating identities according to privacy needs, this work simplified the credential generation through taking account of the individual contexts defined by user preferences and conditions, and through providing an interface for human entities (Wild, Ast, et al., 2013). For mitigating risks towards exploitation of authority and personal data beyond original intentions in delegation scenarios, this contribution also enabled delegators to control the scope of delegates through a set of constraints. By

contributing a dedicated delegation approach (Scholtz et al., 2015a; Wild et al., 2015), all entities present in a delegation can maintain their already existing identities and, thus, reuse their personal data. Complementary to the contributions for enhanced security through fine-grained filtering and tamper-evidentness, this work allowed delegators to protect their personal data against data disclosure and unnoticed manipulation.

With regard to Scenario 2.3: “Holiday Replacement”, the proposed solution enables delegators like Alice to precisely specify the scope a delegate like Casey is permitted to operate in. Employing the CAC key component as part of the solution, it is no longer necessary that Alice shares her credential with her co-worker Casey in order to enable him to act on her behalf or to access her data. Through relying on two distinct identities for delegator Alice and delegate Casey, she can protect her personal data from unwanted disclosure and modification by Casey using the FGF and TE components. Moreover, all activities Casey performs on Alice’s behalf are accountable.

By assuming that persons permit obtaining their individual contexts, this contribution possibly interferes with identity owners which have stringent requirements towards privacy, though it is an optional feature solely responsible for properly factor in user conditions and preferences in order to enhance security. Here, identity owners must accept to make an additional effort for safeguarding their personal data through employing measures for context-aware control. By furthermore assuming that companies are willing to settle with just obtaining access to personal data and to integrate means into SPs for delegated access whilst obeying the scope defined by delegators, this contribution possibly interferes with current business models (cf. findings for Activity 2.1). However, a single component implementing the security-enhanced WebID authentication sequence (cf. Figure 4.7) encapsulates the protective means that SPs must integrate to ensure basic features of CAC.

Despite the overall sufficient contribution to address Problem Cause 3.1: “Insufficient Control of Identity Based on Individual Context”, including 3.1.1: “Inadequate Consideration of Individual User Conditions”, 3.1.2: “Risk of Improper Use of Identity Data in Delegation Scenarios” and 3.1.2.1: “Missing Control of Delegation Conditions by Delegators”, there is further work required, not only in terms of advanced detection of individual conditions and improved consideration of preferences of identity owners, but also with regard to specification of constraints to determine a delegate’s scope in delegation scenarios.

Activity 3.2: Mitigate Risk of Identity Theft and Tampering of Personal Data

To carry out this objective, our contribution facilitated to detect malicious manipulation of personal data and even identity theft through tamper-evidentness. In addition to identity owners, it also enabled requesting entities of arbitrary type, like persons or SPs, to discover anomalies in personal data as an indication of tampering by aggressors. It therefore contributed to enhance security during WebID profile access and authentication. Without relying on prior knowledge through centralized authorities, this work created a foundation for public verification via URI with signature data being attached to personal data of individual identity owners. Even though the proposed tamper-evidentness mechanism cannot prevent malicious manipulations, it implements the preliminary stage to protection, i.e., the discovery of such attempts regardless of whether tampering happened temporarily or permanently (Wild et al., 2014)

Recalling Scenario 2.3: “Holiday Replacement”, entities like Alice do not have to fear malicious manipulations of her personal data when employing

the proposed solution for tamper-evidentness, even though she explicitly allowed other entities, like delegate Casey, to act on her behalf³⁰.

By assuming that persons are willing to make an additional effort for protecting their personal data through tamper-evidentness, this contribution possibly interferes with existing conducts of persons to mainly entrust third parties with safeguarding their personal data. Similar to monetary capital, management and protection of personal data—as part of the social capital—should be chiefly the responsibility of the entities the personal data actually describes and belongs to. By furthermore assuming that companies are willing to settle with just obtaining access to personal data and to utilize means for integrity protection, this contribution may also interfere with current business models (cf. findings for Activities 2.1 and 3.1). Yet, companies are relieved from integrating extensive protective measures to ensure integrity of user data on their own.

Despite the overall sufficient contribution to address Problem Cause 3.2: “Risk of Identity Theft and Tampering of Personal Data” and 3.2.1: “Lack of Means to Detect Identity Theft and Manipulation”, there is further work required, especially with regard to simplification of applying means for tamper-evidentness on personal data and of memorizing the identifiers which hold the key representation.

Activity 3.3: Increase Range and Granularity of Access Control

To carry out this objective, our contribution reinforced the responsibility of identity owners for their personal data and, thus, also for controlling access of requesting entities. Instead of utilizing heterogeneous access control measures offered by individual SPs, this work enabled a holistic user-centric authorization throughout different web systems, web applications and web

³⁰This does obviously not include the case, in which a delegator, like Alice, permits a delegate, like Casey, to make adjustments to her personal data on her behalf.

services. According to self-deterministic IdM, identity owners control access to personal data stored on their own or on sufficiently trusted systems by employing resource-based authorization as well as customized views (Wild, Chudnovskyy, et al., 2013a). Through fine-grained filtering on the basis of RDF triples and individual elements (Wild, Chudnovskyy, et al., 2013b), customized views protect against unwanted data disclosure through specific requesting entities or groups. While semantic resource-based access control is applicable to contents of arbitrary types, fine-grained filtering focuses on white-/blacklisting semantic data described using RDF. When combined, both measures allow for largely reducing the diversity of access control facilities, for increasing scope and granularity of access control, and for creating more standard-compliance. Identity owners can store all filter specifications at a single place, which eases migration of access control settings.

With reference to Scenario 2.1: “Online Shopping” and 2.3: “Holiday Replacement”, the proposed solution enables entities like Alice to create fine-grained filters on her personal data in order to provide requesting entities, like SPs for online shopping or delegate Casey, only with the data they are allowed to retrieve. That is, Alice can reduce maintenance efforts while protecting her data by managing only one repository containing all relevant personal data, but with multiple views customized to the particular trust relationships she has with different requesters.

By assuming that persons are willing to employ measures for extended access control, this contribution possibly interferes with the initial effort that is necessary to transfer and consolidate the distributed access control preferences that identity owners set up on diverse SPs. Yet, making this sacrifice would reduce the dependency of individual entities on different access control mechanisms offered by SPs. Moreover, identity owners would benefit from a holistic protection of their data, which they

can control from a single place yet with an effect on multiple requesting entities, like service providers.

By furthermore assuming that companies are willing to settle with just obtaining access to personal data, this contribution may also interfere with current business models (cf. findings for Activities 2.1 to 3.2). As identity owners can directly control access to their data from a single place, companies do not have to integrate such means into their products any longer and, thus, are—at least partially—relieved from attacks that normally would have caused data disclosure. Turning away from *own* application-specific facilities for controlling access to personal data of users would also shift more responsibility from companies towards individual persons.

Despite the overall sufficient contribution to address Problem Cause 3.3: “Incomplete Range and Granularity of Access Control” and 3.3.1: “Limitation of Access Control Facilities to Specific SPs”, there is further work required, especially in terms of reducing the maintenance efforts of customized views when identity owners make adjustments to their personal data.

Having carried out all required activities, the next subsection continues with discussing the evaluation findings in terms of result delivery.

8.3.2 Delivery of Results

The following evaluation results outline how well this work addressed the primary causes of the central problem by delivering three results:

Result 1: Improved Modeling of Security Aspects for Web Systems

To deliver this objective, our contribution built upon the successful execution of Activity 1.1 to extend the means for modeling secure web systems (Ast et al., 2014; Braune et al., 2014; Scholtz et al., 2015b; Wild and

Gaedke, 2009; Wild and Gaedke, 2014). With the thus provided means, this contribution enabled companies to treat security in web systems more as a first thought or at least with the same care as other parts of their business. That is, it puts companies into the position for taking greater account of security in the engineering of web systems. Here, employing the IdM life cycle and WAM allows for conciseness, review and simplicity when engineering and making architectural adjustments to web-based solutions. It furthermore assists web engineers in creating machine-readable architecture descriptions of SOA-based web systems. The proposed modeling facilities established a basis for simplifying activities such as search, recollection and security-compliant integration of components into web system architectures, regardless of initiators being human or machine.

By assuming that companies are willing to use extended means for modeling web systems with special focus on security-related aspects, this contribution possibly interferes with the development methodologies companies are currently employing (cf. findings of Activity 1.1). While this contribution relies on CBWE for rapid composition of web systems from reusable building blocks, our proposal is combinable with other approaches through making use of standard-compliant technologies and enabling a homogeneous way of modeling heterogeneous web systems.

With overall largely addressing Problem Cause 1: “Security of Web Systems Treated as Afterthought”, we successfully delivered the result.

Result 2: Reduced Need for Accumulation of Personal Data by Third Parties

To deliver this objective, our contribution built upon the successful execution of Activity 2.1 to offer an alternative to customer lock-in (Chowdhury et al., 2013; Chudnovskyy, Wild, et al., 2012; Satzger et al., 2014; Tschudnovsky et al., 2013). Self-deterministic IdM empowers individual entities

to manage their personal data independently from SPs on their own or trusted premises. Through fostering distributed personal data repositories under control of the corresponding identity owners rather than some centralized authorities, this contribution helps to reduce the accumulation of personal data and, thus, to decrease the attractiveness for attacks. Requesting entities, such as service providers, might maintain access to personal data as usual³¹ but to the conditions specified by individual identity owners. Moreover, companies can independently obtain metadata when users employ their SPs. This furthermore reduces the necessitation for companies to aggregate larger amounts of individual persons' data as part of their business model and to spare with export functions for preventing drain of social capital to competing product offerings. Due to the fact that identity owners manage their personal data themselves, the latter functions are even no longer required.

By assuming that companies are willing to employ enhancements towards reducing necessitation of personal data accumulation, this contribution possibly interferes with old and strict ways of thinking how IdM should be implemented and creates fears concerning the very existence of some businesses. Nevertheless, we believe that the benefits of self-deterministic IdM outweigh these issues and bear hard to foreseen value for companies, like potential access to extensive, cross-domain personal data of users.

With overall largely addressing Problem Cause 2: “Accumulation of Personal Data by Third Parties”, we delivered the result successfully.

Result 3: Extended Means for Control and Protection of Personal Data

To deliver this objective, our contribution built upon the successful execution of Activities 3.1 to 3.3 to a) improve control of identity based on

³¹This is a controversial issue, yet if identity owners entrusted parts of their personal data to some SPs in the first place, they would probably grant them also access to data that is self-managed by them.

individual context (Scholtz et al., 2015a; Wild, Ast, et al., 2013; Wild et al., 2015), b) mitigate risk of identity theft and tampering of personal data (Wild et al., 2014; Wild et al., 2015), and c) increase the range and granularity of access control (Wild, Chudnovskyy, et al., 2013a; Wild, Chudnovskyy, et al., 2013b; Wild et al., 2015). With self-deterministic IdM in place, identity owners do not need to rely on SP-specific safeguard measures anymore. The contribution enabled them to holistically apply protection preferences beyond specific service providers, applications or domains. User-centric protection of personal data increases both control and transparency of safeguard measures, but also transfers responsibilities from companies to the entities the personal data actually belongs to.

By assuming that persons are willing to apply security enhancements for personal data management, this contribution possibly interferes with their existing conducts (cf. findings for Activity 3.2), yet when treating social capital similar to monetary capital, management and protection clearly becomes a responsibility of individual persons. By furthermore assuming that companies are willing to employ enhancements towards protecting personal data, this contribution may also interfere with existing business models and safeguard measures (cf. findings for Activities 2.1 to 3.2). However, self-deterministic IdM also largely relieves companies from adopting complex measures in their service providers.

With overall completely addressing Problem Cause 3: “Incomplete Control and Protection of Personal Data”, we successfully delivered the result.

Having delivered all required results, the next subsection proceeds with discussing the evaluation results in terms of purpose achievement.

8.3.3 Achievement of Purpose

To enhance security in managing personal data by web systems, this dissertation successfully produced three results for a) an improved modeling of security aspects for web systems, b) a reduced need for accumulation of personal data by third parties, and c) extended means for control and protection of personal data. Through prototypically integrating these results into the Sociddea proof-of-concept platform, we could demonstrate and verify the operational acceptance as well as the integral character of the solution.

Rather than burden web system providers with securing accumulated personal data of their users, the proposed solution built upon an open, system-independent approach to empower individual human and non-human entities to make their consolidated personal data publicly accessible through decentralized, self-maintained repositories. The proposed solution furthermore supported individual human and non-human entities with means to manage and protect their data from 1) unauthorized use, 2) unnoticed tampering, and 3) unwanted access. While Research Contributions 1 to 4 taken by themselves do not represent entirely novel approaches or means, their particular field of application, their reliance on existing security artifacts as well as a proven technological foundation, and their synergistic nature drives the innovation beyond the current state of the art, as described in the next four paragraphs.

Research Contribution 1 for “enhanced security in managing personal data” relies on the open silo approach, which fosters self-deterministic IdM. By enabling identity owners to self-manage their personal data in individual, decentralized repositories, attackers would need to deal with multiple distributed targets. According to our solution, identity owners are only responsible for their very own personal data, which lowers the attractiveness and lucrativeness for attackers because they would need to

adjust to the specific preferences and weaknesses of separate systems that host the data. Avoiding accumulation of data from many users in only few silos distributes the central problem and, thus, reduces its severity. The solution architecture and process showed that advantages of the open silo approach in general and WebID in particular do also apply to identities of non-human entities through sharing a common IdM life cycle. This allows for universal identification, semantic description, discovery, linkage and authentication of identities from arbitrary entity classes, including compositions and components of web systems, in a holistic manner and eases utilizing the *same* protective measures to enhance security in managing associated data (cf. Research Contributions 2 to 4). Not only does the solution facilitate modeling web systems and components at design time using WAM, but it also considers personal data management of user during runtime by taking advantage of the same technologies. This fosters high maintainability and, therefore, assuring quality of our solution.

Research Contribution 2 for “context-aware control” introduced an innovative delegation approach that allows a delegate to perform an assigned task using a web application on behalf of a delegator within a defined scope. All involved entities keep their common identities and can access personal data associated with the other identities, i.e., delegator, delegate and web application, as proposed by Research Contribution 1. There is no need for sharing credentials among involved entities. To separate concerns and make all actions accountable, the web application clearly distinguishes between delegator, delegate and delegate acting on behalf of delegator. The solution implements the actual delegation using the *same* artifacts as in the original authentication, i.e., WebID certificates. Furthermore, the delegator may grant controlled access to additional personal data to be used by the delegate in order to work on the task within the scope defined by the delegator (cf. Research Contribution 4). Making use of constraints, the delegator can restrict the scope of a delegate’s actions, where the web

application enforces this scope. A delegator does not need to fear unnoticed personal data tampering, even though the web application to be used is an IdM system and a delegate has read/write access to the delegator's profile data (cf. Research Contribution 3). The collaboration of the CAC component with both the TE and the FGF component illustrates the synergies among our research contributions. By removing references to a delegate from the semantic identity description, a delegator can revoke a delegation at any time, yet the corresponding delegate may use the web application as usual, i.e., not on behalf of the delegator.

Research Contribution 3 for “tamper-evidentness” represents a novel approach to make manipulations of personal data publicly recognizable without requiring a certifying third party like a certificate authority. In the open silo model to IdM, entities store consolidated personal data associated to their identities in distributed, self-managed and web-accessible repositories. These repositories also include public key data for authentication purposes, when using WebID as a representative of the open silo model. By including an inherent source of verification in the public identifier, identity owners can self-certify the authenticity of personal data associated to their identities and store the proof together with the proven data. Although aggressors might tamper personal data, public key data or proof data, they cannot alter the inherent source of verification without changing the identifier. Modifying a WebID identifier would not only cause loss of identification or creating a new identity respectively, but also impair discovery and render existing incoming links useless. As aggressors cannot certify once tampered data without having access to the *real* identity owner's private key, the research contribution for “tamper-evidentness” also assists requesting entities and identity owners in detecting identity theft.

Research Contribution 4 for “fine-grained filtering” introduced an innovative mechanism for creating highly customizable, requester-specific views

on personal data. It enables identity owners to keep control about amount and nature of personal data being presented to requesting entities. Supported human and non-human requesting entities are anonymous users, individual identified entities and groups. By relying on SPARQL as an open standard, view specifications are system-independent, portable, widely supported and directly executable. The granularity of view specifications is only limited by the expressiveness of SPARQL, i.e., fine-grained views can include or exclude distinct elements of RDF triples. The mechanism is not restricted to filter personal data only, but applicable to all RDF-based data sources.

The proposed security enhancements, however, necessitate that individual persons make additional efforts to protect their personal data against unwanted data disclosure, unnoticed tampering, identity theft and prohibited use in delegation scenarios. Enhanced security comes at a price individual entities must be willing to pay or, otherwise, live with the negative consequences. Furthermore, it is required that companies rethink the way of handling and protecting personal data they actually do not own. While our contributions enable a simplified and semantically enriched modeling of web systems with focus on security aspects, an accumulation of social capital outside the direct control of identity owners bears avoidable risks. Distributing and associating personal data with the actual owners, as specified in our proposal for self-deterministic IdM, helps to reduce the target characteristic of web systems for attacks. That is, aggressors probably will not jeopardize criminal prosecution, when the potential reward of a successful attack is obviously low. Finally, governments have to continue supporting individual persons to exercise more sovereignty and control about their personal data, if necessary, with measures for undiscerning companies.

In Section 1.4 we operationalized the purpose by narrowing it into Research Questions 1 to 5, with answers given on the basis of Research Contributions 1 to 4 in the following:

Research Question 1 dealing with the topic of “how to model web systems at different abstraction levels in an appropriate, interoperable, interpretable and supportive way that also puts a strong emphasis on security involving trust and invocation relationships” has been addressed by delivering Result 1. The result includes a dedicated process model to specify an IdM life cycle for compositions of web systems and associated components. While the contribution supports modeling web systems including trust and invocation relationships at the composition level using WAM, it also enables describing individual entities at the component level via specialized RDF-based vocabularies like WSDL 2.0 RDF mapping. Furthermore, the proof-of-concept platform *Sociddea* shows how to assist web engineers in their modeling tasks by offering user interfaces like the web-based diagramming tool to describe compositions with the graphical WAM notation and components by means of a graphical editing tool. No matter the tool employed for creation, all resulting models are interoperable and machine-interpretable due to their semantic heritage.

Research Question 2 dedicated to the topic of “how to describe and identify web systems, applications and services, as a basis for applying protective measures, by means adequate to describe and identify other entities like individual persons” has been addressed by our contribution that focuses on self-deterministic IdM. Here, identities and related management features are not limited to particular systems or domains. Building upon open standards, Research Contribution 1 and Result 2 enabled universal identification and semantic description of identities from entities of arbitrary class. The contribution allows for identification and description of heterogeneous entities related to web systems, including web services and individual persons, in a homogeneous way. The openness, integrability and extensibility of chosen technologies also established a foundation for providing protective measures.

Research Question 3 linked to the topic of “how to enable individuals to maintain ownership about their personal data, especially personally identifiable information, and allow controlled access for third parties incl. other users, web applications and web services” has been addressed by delivering Research Contribution 1 in association with Results 2 and 3. They foster self-deterministic identity management to empower individual entities rather than authorities and provides means for an enhanced security. The contribution enables individual entities to take responsibility of their personal data and store it at self-defined, sufficiently trusted locations. The underlying user-centric approach puts identity owners in control of their personal data, so that they can define the conditions to which requesting entities can access the data.

Research Question 4 approaching the topic of “how to authorize entities, including individuals as well as web applications and web services, to use web applications and web services within a defined scope on behalf of other entities in highly distributed environments” has been addressed by delivering Research Contribution 2 in association with Result 3. They introduce a dedicated delegation approach, which clearly distinguishes between the roles of delegator, delegate and web application, while permitting all involved entities to keep their identities and reuse related personal data. Not only does the delegation approach support entities of arbitrary class to fulfill the role of the delegator or the delegate, but also to restrict a delegate’s scope of action using constraints.

Research Question 5 covering the topic of “how to protect personally identifiable information against data disclosure without the corresponding individual’s knowledge or intent, against tampering and (identity) theft, and against misuse by third parties including users, web applications and web services” has been addressed by delivering Research Contributions 2 to 4 in association with Result 3. They provide three key components for

fine-grained filtering, tamper-evidentness and context-aware control that extend the proposed solution architecture. Fine-grained filtering allows for creating customized views on an identity owner's personal data for different requesting entities and, consequently, for restraining unwanted parties from accessing sensitive data. Tamper-evidentness enables both identity owners and requesting entities to detect manipulations of personal data and, thus, also indications of identity theft. Finally, context-aware control assists in preventing unauthorized exploitation of identities and associated personal data in delegation contexts.

With answering all research questions successfully, we implicitly prove the underlying Hypotheses 1 to 5, as outlined on page 15. Having achieved the purpose, the next subsection continues with discussing the evaluation results in terms of our contribution to the overall objective.

8.3.4 Contribution to Overall Objective

To contribute to increase the protection of privacy and reduce the risk of personal data disclosure through successful attacks on web systems, Research Contributions 1 to 4 partially addressed also several effects of the central problem, as described next:

By 1) implementing a user-centric, self-deterministic approach to IdM, 2) putting identity owners in charge of managing personal data on their own, and 3) providing means for data protection, we made a contribution to improve the overall control of personal data by individual persons throughout web systems (countermeasure for Problem Effect 2). Rather than relying on central authorities or specific SP-centric facilities for identity data management, identity owners only have to maintain and control access to one or few individual repositories containing their consolidated personal data. This does not only lower efforts in terms of handling cre-

dentials (countermeasure for Problem Effect 1.3.1), but also supports to accomplish the *privacy by default* initiative and helps to realize the *privacy by design* vision because achieving the purpose put identity owners in the position to set safe defaults regarding access to their consolidated personal data (countermeasures for Problem Effects 3 and 4). Here, the FGF and TE components allow for hiding sensitive data from unauthorized requesting entities in a fine-grained manner and for certifying the authenticity of data requested by authorized entities (countermeasure for Problem Effect 1.3).

Through separating service offerings from IdM and distributing personal data repositories according to the preferences of the *real* data owners, we also contributed to relieve companies from making investments for securing and storing personal data they do not own. Making identity owners responsible for managing their personal data enabled companies to focus on their core businesses. By minimizing the necessity to store and, thus, to accumulate large amounts of user data, we made a contribution to reduce the attractiveness for aggressors that target data disclosures and, therefore, partially mitigated risks for companies originating from data breaches made public (countermeasure for Problem Effect 1.1). Without potential access to large sets of accumulated personal data, aggressors cannot reach high returns on investment, which potentially has a positive bearing on cutting down today's high number of attacks on web systems and reducing risks associated with monetization of personal data (countermeasures for Problem Effects 1 and 1.2).

8.3.5 Contribution Beyond the State of the Art

Benefiting from the openness of WebID, we contributed to increase control and protection of this lightweight user-centric approach to IdM by three dedicated yet complementary components. Utilizing existing security artifacts

and semantic facilities, we non-invasively extended WebID by machine-readable means for control (component for context-aware control) and protection (components for fine-grained filtering and tamper-evidentness). All extensions are optional, i.e., their addition or removal does not interfere with regular IdM procedures. Furthermore, the use of semantic languages for specifying identities of human and machine entities supported addressing the interoperability issues of the chosen web engineering approach. Enabling homogeneous identification, description and linkage of heterogeneous entities contributed to holistically model web systems at both the composite level and the level of individual building blocks. Not only does this include web applications and web services, but also involved human and non-human users. Besides bearing advantages in assuring quality and maintainability through limiting the set of necessary methods, languages and tools, it also allowed for opening an integral perspective on security for both human and non-human entities. This is because all descriptive and protective measures are applicable to all WebID-compliant identities, which can represent arbitrary characteristics of arbitrary entities.

With finishing the discussion of the evaluation results, the next section sums up the outcome of this chapter.

8.4 Summary

In accordance with LFA, an objectives-based study matched the needs for an evaluation procedure that facilitates to “determine how well each objective was achieved” (Stufflebeam, 2001). For specifying the evaluation characteristics, it was therefore reasonable to reuse the objectives of Chapter 2 also as assessment criteria. By defining a bottom-up strategy to evaluation, the discussion of retrieved findings started with verifying the execution of five consolidated activities to tackle secondary causes of the central

problem, before continuing with assessing the production of three results to approach primary problem causes. With the successful result delivery, we evaluated the achievement of the purpose of this work to address the central problem. It has been apparent that this work included several contributions to improve the protection in handling personal data. Applying these contributions, however, requires that a) individual persons are willing to invest additional efforts, b) companies rethink their business models of locking customers in and accumulating their personal data, and c) governments continue with their support towards strengthen the rights of persons for more control about their data. By overall successfully enhancing security in managing personal data by web systems, the discussion of findings completed with evaluating to which extent this dissertation contributed to achieve the overall objective and to advance beyond the state of the art.

Based on the obtained evaluation results, the next chapter concludes this dissertation.

Conclusion

9

To conclude this dissertation, the final chapter begins with summarizing the major research contributions in Section 9.1. By reviewing what has been accomplished, Section 9.2 then critically reflects this work also by taking the results of the overall evaluation into account. Finally, Section 9.3 provides an outlook towards future work to highlight links for a systematic evolution from the current state of achievement.

9.1 Contributions

Leveraging the logical framework approach, this dissertation systematically investigated the challenges that emerged from an insufficient consideration of security in today's web systems for the amount of accumulated personal data (cf. Chapter 2). By analyzing the central problem stated in Section 1.2 and breaking it down into more manageable causes and effects, we first created a problem hierarchy from which we then derived a consolidated set of objectives representing the means necessary for achieving the purpose declared in Section 1.4. With the results retrieved by reviewing state-of-the-art technologies related to web engineering and identity management (cf. Chapter 3), our proposal for enhancing security in managing personal data by web systems built upon the foundation of self-deterministic identity management for both human and non-human entities.

Rather than entities having to distribute their personal data among various SP-centric repositories in order to make use of provided services, the solution proposed in Chapter 4 involves consolidated personal data repositories in control of individual identity owners or entities they sufficiently trust. In consequence, service providers and other interested parties, like persons and web services, may still maintain access to personal data yet under the terms of the *real* data owners, and not those of the service providers. This empowers the majority of individual identity owners instead of the minority of authorities. To support owners in managing their identities, the proposal includes an IdM life cycle that is applicable to persons, compositions and components of web systems. Through relying on WebID, the proposed solution features semantic RDF-based description of personal data as well as ownership-based authentication, which is compatible with arbitrary entity classes. According to our proposal, both human and machine entities can therefore access and interpret personal

data associated with identities – also with the chance to understand underlying concepts that are part of the Linked Data cloud. Each identity that represents an entity within a particular context is furthermore universally identifiable via a URI that refers to personal data.

Not only does universal identification allow for linking and connecting among human entities, but also for qualifying the relationships between non-human entities that constitute web systems. Here, WAM facilitates modeling web-based solutions by employing introduced concepts at the level of both web systems and underlying entities, like web services. Holistically enabling semantic description, identification and linkage for architecture models of web systems and involved web entities paved the way for a simpler and well-directed management and evolution. The proposed solution hereby assists web engineers in creating machine-readable big pictures of SOA-based web systems with a web-accessible WAM diagramming tool. In order to substantiate architectural blueprints, the solution permits utilizing semantic vocabularies, like the WSDL 2.0 RDF mapping, to specify underlying entities, like web services, at design time and refining them at runtime, e.g., updates to interface definitions. Through authentication via WebID-TLS and resource-based access control via WAC, our proposal supports taking account of elementary security aspects.

To enhance security beyond elementary measures for authentication and authorization, three contributions complement the fundamental solution in respect of fine-grained filtering, tamper-evidentness and context-aware control (cf. Chapters 5 to 7).

A dedicated component against unwanted retrieval enables identity owners to create customized views on their personal data, with the capability to apply a detailed filtering of personal data at the level of RDF triples and individual elements for specific requesting entities or groups. In addition

to controlling the way personally identifiable information is exposed to third parties by white- and blacklisting of data subsets, the FGF component also enables to harness the full expressiveness of SPARQL, e.g., to hide filter specifications from requesting entities via filters. For ensuring accessibility, portability and maintainability of filter preferences by human and non-human entities, associated specifications remain unencrypted, are self-contained and fit into the existing semantic ecosystem by relying on SPARQL and adequate vocabularies.

A second component against malicious manipulation allows both identity owners and requesting entities for discovering personal data integrity anomalies that potentially indicate tampering or even identity theft. Without the need for prior knowledge through centralized authorities, the TE component enables public verification of personal data integrity via URI. That is, the verification happens without external dependencies because signature data is stored as part of the personal data of an identity owner. Combined with the complementary FGF component, integrity checks are performed on customized views rather than on personal data per se.

Finally, a third component against misuse of authority in delegation scenarios facilitates verifying the scope of delegates that act on behalves of other entities, with delegators being enabled to control the scope through a set of constraints. With our dedicated approach to delegation, entities keep their *common* identities, like manager and secretary in a business context, also in delegation scenarios. This fosters reuse of already available personal data. Relying only on existing WebID artifacts with RDF-based delegation specifications inside a delegator's profile, a delegate enters a delegation context by using a particular enabling credential, yet the delegate maintains the desired identity and related personal data. For assistance and simplification of credential creation, the CAC component takes account of user preferences including individual conditions, privacy need, signing type

and cryptographic strength. Combined with the FGF component, distinct identities for delegator, delegate and service permit creating customized views on delegator's personal data for a delegate or an employed service. Combined with the complementary TE component, delegators can make sure that unauthorized modifications of personal data by delegates go not without notice by concerned delegators and services.

For proof-of-concept implementations and acceptance tests, the Sociddea integration platform holistically put the constituents of the entire solution, including the key components for enhanced security, into effect. The complementary character of these components enables synergies through combination, which contributes to enhance security beyond utilizing each security feature separately on its own. Rather than breaking with proven methods, languages, frameworks or tools, our proposal assists in assuring a high level of quality and maintainability by building upon already existing security artifacts of WebID and seamlessly integrating into the given semantic landscape established through RDF.

With results retrieved from the overall evaluation (cf. Chapter 8), the following section critically reviews the research contributions of this work.

9.2 Review

While the evaluation of the solution as whole indicates an overall positive effect on enhancing security in managing personal data by web systems, it also points towards a number of drawbacks related to the research contributions. By enabling to identify, describe and interlink web systems and underlying components via WAM and other eligible semantic vocabularies, this dissertation contributed to establish a foundation for a systematic engineering of web-based solutions with focus on security aspects, yet further

efforts are required in terms of discovery and selection of suitable components, and automatic generation of software building blocks from semantic specifications of compositions and components. Moreover, companies must be willing to engage in this kind of modeling in accordance with the IdM life cycle, which may interfere with process models already employed by them.

Placing reliance upon WebID as part of the proposed solution necessitates to embark on a not yet widely adopted IdMS and to turn away from today's common knowledge-based authentication for the sake of 1) ensuring compatibility with machine entities, 2) resolving the password fatigue issue, and 3) providing a higher cryptographic strength of credentials. Despite the research contribution for CAC, ownership-based authentication still implies some overhead in creating, exchanging and using credentials especially when identity owners rely on miscellaneous user agents, operating systems, platforms or devices. Here, human entities may benefit from physically outsourcing credentials to external devices they can carry with them, like regular keys or key cards.

The proposal reinforces a self-deterministic, user-centric IdM. With focus on individual identity owners, it does therefore not make use of central authorities to intensively prove identities claimed by entities before allowing for creating corresponding credentials. While self-asserted identities enable to imitate entities through illegitimately taking over specific attribute sets, this problem also occurs in SP-centric identity management systems, when aggressors exploit unclaimed identities or reuse identities after associated owners have permanently left specific service providers. However, our proposal prevents aggressors from proclaiming already used identities for themselves without having profound skills and technical capabilities. As in the proposed solution a single identifier refers to all personal data of an identity owner independently from service providers and other entities, an identity owner has to manage only one personal data repository per identity.

Self-deterministic IdM as per our proposal involves personal data repositories that are either in full control of the corresponding identity owners or in control of sufficiently trusted entities. In contrast, distributed management of personal data by service providers bears risks that are out of control of the entities concerned, i.e., every new aspect outsourced to a third party adds to the attractiveness for data disclosure, tampering and unauthorized use. The proposed solution consequently redistributes personal data management from the service providers to the individuals that actually own the data. Although this makes it more difficult for aggressors to obtain a large set of personal data by exploiting a specific vulnerability, the issue of protecting personal data is also outsourced to individual identity owners. By not relying on third parties any longer, identity owners are now held accountable to secure their personal data on their own, e.g., through keeping their systems up-to-date. It is our conviction that this increases not only the sense of responsibility of individual entities in managing and protecting their personal data but also their awareness about quality and quantity of the data they have aggregated and potentially provide requesting entities with. Both the high heterogeneity of systems and the low risk-reward ratios of potential attacks contribute to restrict the endeavors of aggressors, yet such heterogeneity and the varied skills among identity owners may make it difficult for identity owners to establish an appropriate level of protection.

Three dedicated components extend the proposed solution in order to support identity owners to reach such an appropriate level of protection through offering means for fine-grained filtering, tamper-evidentness and context-aware control. However, all security enhancements proposed in this dissertation entail additional efforts by concerned entities. The filters identity owners can specify to create customized views on their personal data using the FGF component may require maintenance once attributes are added, changed or removed. Furthermore, the TE component involves special identifiers that include key information. They are therefore diffi-

cult to memorize for human entities—using different representations for identifier, like QR codes, may alleviate this issue. For producing tamper-evident personal data, identity owners must employ particular user agents responsible for signature creation. Similar to FGF, modifications to personal data make it necessary to update signatures. Finally, the CAC component allows for detecting only some contextual conditions at the moment and users have to manually specify their preferences, even though they might already stated them as part of their identity data.

To conclude, security comes at a cost concerned entities must be willing to invest. The potential impact through direct effects and aftereffects of successful attacks (cf. Section 1.2) exceeds the effort necessary for identity owners to properly protect their personal data. Conducting an exhaustive evaluation beyond the overall evaluation of Chapter 8 might facilitate precisely assessing the level of contribution of this dissertation to attain the overall objective.

9.3 Outlook

Based on the research contributions and the results obtained by critically reflecting the work accomplished, there is a multitude of ways to drive the evolution from this point of achievement. For this reason, we just present an exemplary set of selected paths, starting with tangible ones and then approaching towards more visionary ones.

In this work, identities are characterized by rather static personal data, yet this ignores important variable aspects of entities. More dynamic identities would allow for covering current conditions of entities as part of their personal data, e.g., human mood, health data or utilization in case of non-human entities. Data from sensors specific to a particular entity can hereby

provide the necessary foundation for dynamic identities representing this entity in different contexts. It would furthermore enable dynamic adjustments of web systems and involved components, like web services, on the basis of identity data updated by SPs or the entities themselves, e.g., when reaching certain thresholds in utilization. Not only would this facilitate systematically propagating data on dynamic reconfigurations of machine entities, like interface adjustments due to protocol changes, but also retrieving an always up-to-date big pictures of web systems. Monitoring and analyzing data on compositions could assist in predicting the evolution of web systems and, consequently, enable to proactively make profound decisions to guide the evolution in a suitable direction. Here, researching the topic of dynamically replacing components or delegating their tasks to compatible ones might be beneficial for automating activities related to the evolution of web systems.

Enabling delegation across entity classes might support a closer cooperation between human and machine provided services, e.g., in maintenance scenarios. To manage delegating tasks to a large yet undefined group of individual entities via an open call, like in web-based crowdsourcing scenarios, the proposed delegation procedure needs to relax the restriction of assigning a task to a particular delegate. Involving groups of known as well as unknown delegates while protecting a delegator's personal data increases the demands on context-aware control, tamper-evidentness and fine-grained filtering. In addition to more precise scope specifications and advanced detection of dynamic conditions during credential creation, cascades of filters and morphing views on personal data might become necessary. Moreover, making tamper-evidentness an essential element of the WebID authentication sequence would increase trust in personal data because requesting entities could publicly verify the binding between personal data and identifier that denotes a claimed identity. Therefore, a simplification of the hard-to-memorize identifiers, like short URLs, and applying protective means from within common user agents, like web browsers, are

desired contributions of future work. Similar to anti-virus software, specialized tools could assist individual entities in various matters concerning the protection of their personal data, such as checking that 1) the system offering their data has the latest security patches installed, 2) tamper-evidence measures and customized views are up-to-date and according to the identity owner's preferences, and 3) scope specifications are obeyed by delegates.

Only when reaching a critical mass of identity owners that employ WebID, it becomes possible to obtain sustainable validation results, like in terms of scalability, and to benefit from a plethora of universally identifiable, linkable and semantically described identities that represent entities of diverse classes. As a consequence, it is important to add further stimuli for users beyond mere security enhancements in managing personal data, e.g., through services that provide convincing feature sets.

By making semantic identity data on persons, web systems and components available in line with the security needs of individual identity owners, the proposed solution contributes not only to the growth of the Linked Data cloud in a controlled way but also to enlarge the basis of semantic information to perform cognitive computing on. Supporting machine learning through open data repositories allows for a variety of promising applications in the future.

LFA Artifacts

A

To describe the basis for the systematic discussion of the challenges conducted in Chapter 2, this appendix contains the relevant LFA artifacts. While Sections 1.3 and 2.2 present the outcome of the LFA stakeholder analysis, we detail the underlying findings in Table A.1. This table, named stakeholder analysis matrix, outlines the stakeholders which are affected by the central problem (cf. Section 1.2) and thus by possible endeavors for solving it, such as this dissertation project. Furthermore, we describe the stakeholders by 1) their basic characteristics, 2) their interests and concerns in terms of the problem, 3) their capacities to overcome the problem, and 4) possible actions to attract their interests in order to address the problem (EC, 2004).

Maintaining the same structure as the discussion in Chapter 2, we continue with illustrating the problem tree. This problem tree originates from interlinking the cause-effect relationships between all problems explained in Section 2.2. That is, Figure A.1 depicts both the problem causes, which pile up to the central problem, and the problem effects, which arise from the central problem.

Similar to the problem tree, the objective tree visualizes the interlinked means-ends relationships between all objectives presented in Section 2.3. Consequently, Figure A.2 illustrates the means, which enable to achieve the purpose, as well as ends, which contribute to reach the overall objectives. Figure A.3 represents a consolidated version of Figure A.2, in which closely related objectives have been combined.

The so-called Logframe matrix represents the documented product of the objective analysis. Represented by Table A.2, the Logframe matrix does not only show all identified objectives at different levels in the means-ends hierarchy, but also lists associated success indicators, verification sources and assumptions.

Table A.1: Stakeholder Matrix

Characterization	Interests & Problem Impact	Motivation to Tackle Problem	Possible Actions
<p>Web users Users of all characteristics that use the web for their particular purposes</p>	<ul style="list-style-type: none"> • Want to use the web in a customized way for various purposes incl. shopping, news, self-expression, entertainment and education • Concerned about their privacy especially due to recent news coverage about disclosures and leaks • Critical regard the sphere of influence of large IT and/or web companies 	<ul style="list-style-type: none"> • Keen interest on an open, free and unrestricted web • Improved customization and user experience when browsing the web; Focus on the individual • Regain control about personal data and, thus, more privacy 	<ul style="list-style-type: none"> • Provide means to control use of personal data • Provide means to protect personal data • Provide added value through use of profile data independently from particular platform

Continued on next page

Table A.1 — continued from previous page

Characterization	Interests & Problem Impact	Motivation to Tackle Problem	Possible Actions
<p>Online social network members</p> <p>Web users who actively manage and maintain their UGCs (incl. personal data) with the facilities provided by online Social Networking Services (SNSs)</p>	<ul style="list-style-type: none"> • Want to publish UGCs, present and promote themselves or their work to a broad (specific or unspecific) audience, receive feedback (comments, encouragements, rewards etc.) from a particular group of people • Want to make particular personal data, like curriculum vitae (CV) information, accessible to persons and groups across SNS boundaries • Affected by data silos, vendor/customer lock-in, isolation/monopoly characteristics of popular OSNs; memberships in multiple social networks • Concerned about missing or unclear ownership of personal data (incl. user profile data and UGCs) • Affected by missing personal data exchangeability, reusability and difficult migration of UGCs to other OSNs 	<ul style="list-style-type: none"> • Capabilities/features similar to widely accepted OSNs are provided • Privacy and security needs are respected and are properly taken into account • Ownership of user profile data is regained • Identity theft is detectable • Extensibility, migration, exchangeability and fine-grained access control to user profile data is enabled (also incl. OSN operators) 	<ul style="list-style-type: none"> • Enable users to self-manage their user profile data • Provide means to allow users for deciding whom to make what personal data available to • Increase protection of user profile data from manipulation and identity theft • Achieve context-aware identity creation, filtering and delegation (on-behalf-of)

Continued on next page

Table A.1 — continued from previous page

Characterization	Interests & Problem Impact	Motivation to Tackle Problem	Possible Actions
	<ul style="list-style-type: none"> • Affected by missing data extensibility beyond the scope pre-defined by OSN operators • Affected by high heterogeneity of structure and amount of personal data to be stored in different social networks (e.g., Facebook vs. Twitter) • Have to rely on security facilities provided by Social Networking Service Providers (SNSPs) • Affected by missing facility to establish fine-grained access control to personal data for requesters (which are not members of the social network) and OSN operators • Concerned about analysis of personal data for purposes unaware or unwanted 	<ul style="list-style-type: none"> • Make more out of their user profile data in other scenarios as considered by SNSPs 	
<p>User profile requesters Web users or services that access or monitor the contents (including personal profile data) generated by other web users</p>	<ul style="list-style-type: none"> • Want to stay informed about activities of personal networks, discover and connect to persons with specific interests and activities • Concerned about being member of a social network to get more information/better experience • Affected by dispersed user profile data, different amounts of available data depending on specific OSN, and missing interlinkability between different accounts (identities) of the same entity (person) 	<ul style="list-style-type: none"> • Discovery without being member of OSN • Accessibility to UGCs and user profile data without deep knowledge of specific underlying interfaces (like APIs) • Increase in connectivity and interlinkability 	<ul style="list-style-type: none"> • Improve discovery and accessibility beyond the scope particular OSNs through open standards • Enable unified access to a central, linkable information repository containing all relevant data about the user in an easy to process way

Continued on next page

Table A.1 — continued from previous page

Characterization	Interests & Problem Impact	Motivation to Tackle Problem	Possible Actions
<p>Social Networking Service Providers (enterprises) that enable users to host their contents including profile data at the service provider's premises</p>	<ul style="list-style-type: none"> • Want to increase their social capital (create large user basis by providing facilities to attract and attach users) • Want to maintain or increase profits by selling or sharing insights or information on personal data of users • Want to team up with third-party beneficiaries (like ad companies as income) • Strive for centralization without <i>real</i> openness 	<ul style="list-style-type: none"> • Not interested in change because their centralized social capital would decrease through distribution, with the consequence of both a loss of control and a loss of user data monopoly • Would need to provide/win users with convincing features independent of specific user basis 	<ul style="list-style-type: none"> • (Contrary position) • Ease migration/shift to new business model by enabling openness, relying on standards, creating unified access and representations etc.
<p>Third-party beneficiaries (incl. organizations like ad companies) that team up with or are enabled by SNSPs and/or users (profile owners) to access UGCs or metadata for their own purposes (e.g., advertisement creation)</p>	<ul style="list-style-type: none"> • Concerned about user loss • Want to promote products tailored/customized to the users ("behavioral advertising" or "ad customization") • Want to obtain access to the user's personal characteristics for orientating towards target groups and feedback • Want to maintain or increase profits • Concerned about losing access to insights about a user's behavior and, thus, less means to attract users through tailored offerings 	<ul style="list-style-type: none"> • Independence from OSN operators or support across OSNs • Cost reduction • Greater user basis to gain insights from • Unified access to user data 	<ul style="list-style-type: none"> • Reduce dependency from specific OSNs by enabling unified access to user profile data • Reduce intermediate between them and the users

Continued on next page

Table A.1 — continued from previous page

Characterization	Interests & Problem Impact	Motivation to Tackle Problem	Possible Actions
<p>Web system architects and engineers</p> <p>Group of persons that model web systems, incl. web applications and web services</p>	<ul style="list-style-type: none"> • Want to ease error-prone and time-consuming activities of manually constructing and maintaining web systems through automation • Concerned about weak interoperability of existing modeling solutions and the weak consideration of security 	<ul style="list-style-type: none"> • More independence from domain-specific or proprietary modeling solutions • Modeling with a strong focus on security related aspects • Homogeneity of access to web system, web application and web service models 	<ul style="list-style-type: none"> • Provide means for unified modeling of web systems, including involved applications and services, with a strong focus on security • Provide machine-readable, expressive and extensible descriptions of web entities and means for discovery and interlinkage
<p>Governments</p> <p>Institutions of states or state unions, like the EU, with democratic attributes that represent the intent of their citizens by directing, controlling and administering authority</p>	<ul style="list-style-type: none"> • Concerned about the fact that data disclosures through web systems attacks increasingly clash with efforts to protect civil rights (EC, 2012) • Discontent of citizens that governments cannot fully meet their mandated responsibilities regarding protection of their personal data (missing alternatives to disclosing personal data to third parties for using specific services) (EC, 2011) • Want to assure civil rights and remain in control about what foreign companies are doing with information about (local) citizens • Want to make companies accountable for dealing with personal data of citizens 	<ul style="list-style-type: none"> • Reduction of costs and administrative burden for difficult and time-consuming prosecution of cybercriminals • Establish consistent protection rules across countries 	<ul style="list-style-type: none"> • Support attempt to force the principles of “privacy by design” and “privacy by default” • Assist companies in integrating mechanisms to safeguard personal data into their products beginning from the earliest stage of development (EC, 2012) • Establish the basis for making access, transfer and control of personal data fundamental citizens’ rights

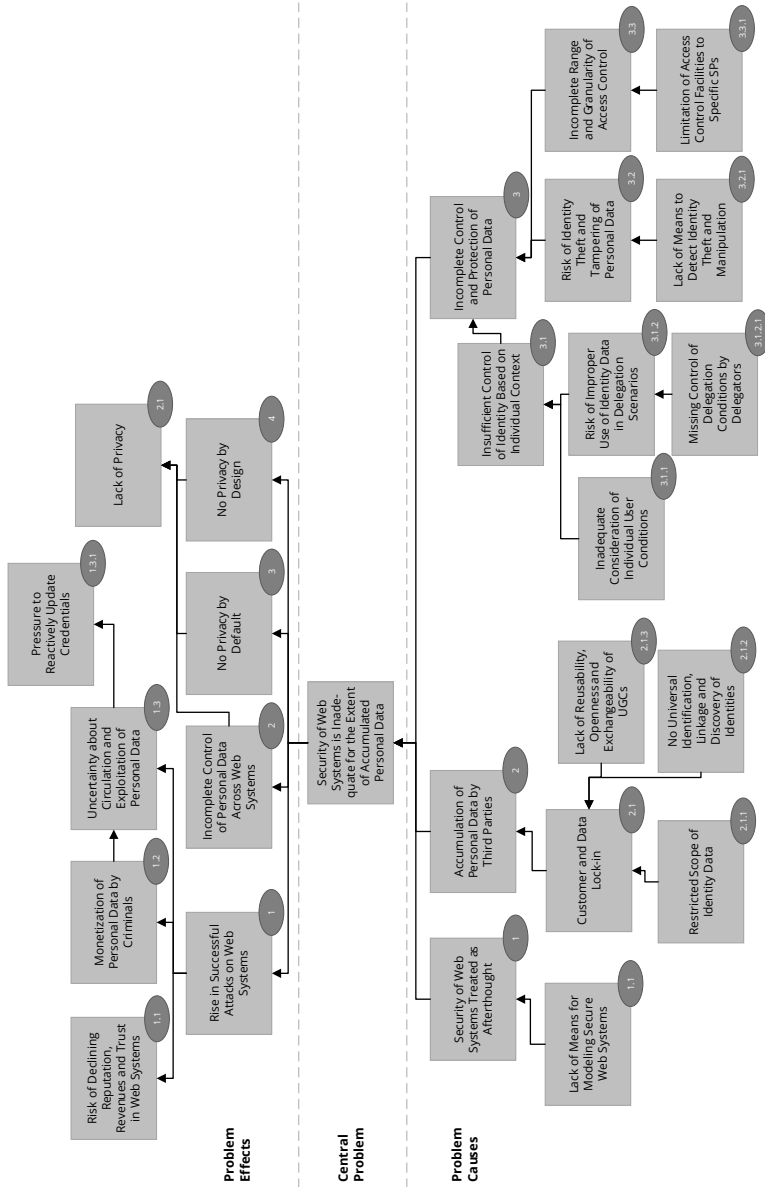


Figure A.1: Cause-Effect Relationships of Analyzed Problems

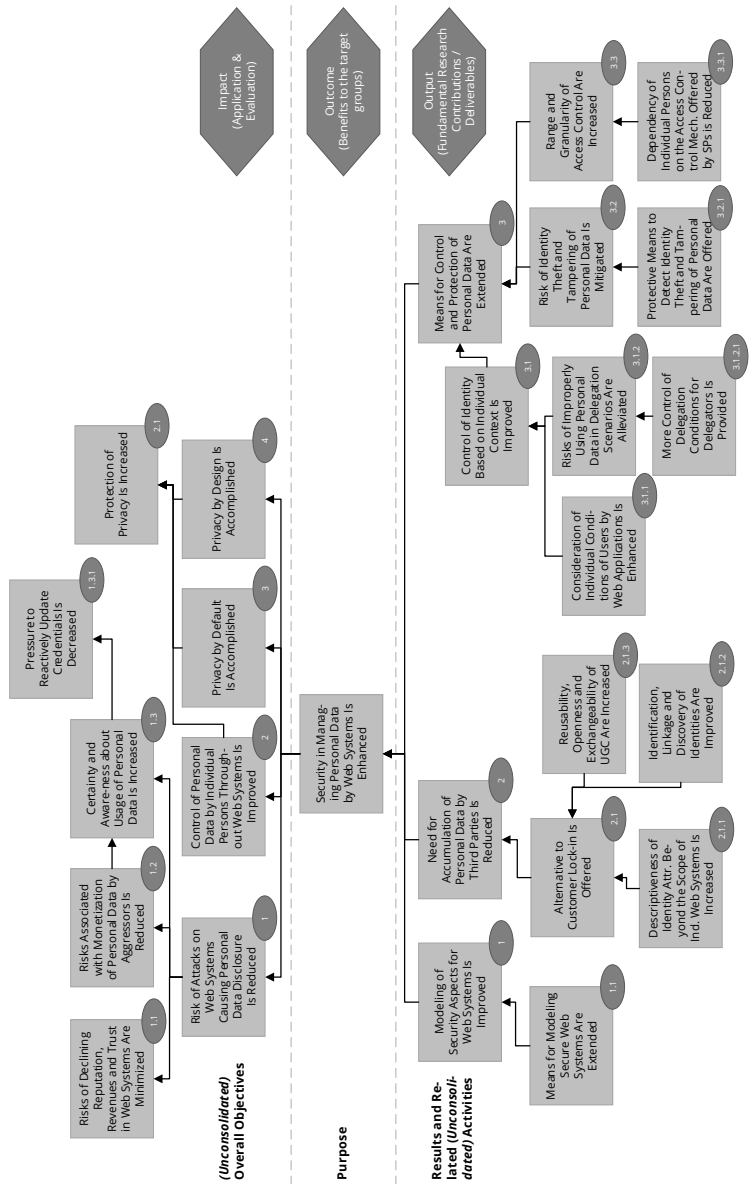


Figure A.2: Means-Ends Relationships of Unconsolidated Objectives

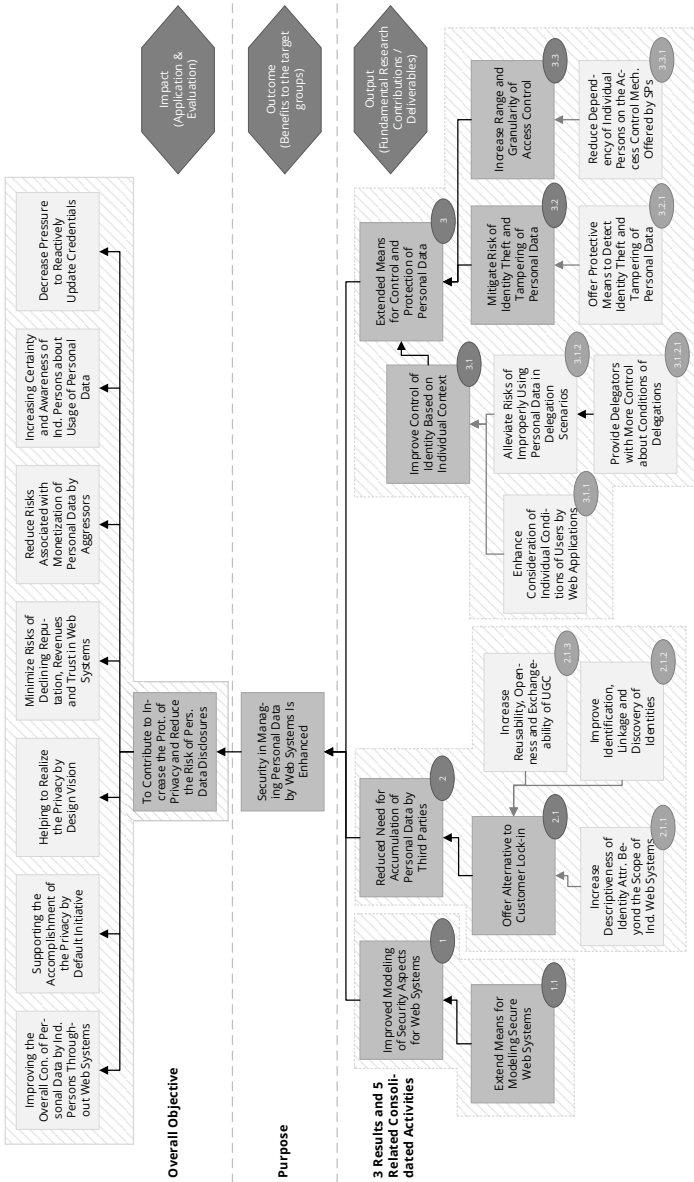


Figure A.3: Means-Ends Relationships Among Consolidated Objectives

Table A.2: Logframe Matrix

Project Description	Success Indicators	Verification Sources	Assumptions
<p>Overall Objective To Contribute to Increase the Protection of Privacy and Reduce the Risk of Personal Data Disclosures</p> <p>Purpose Enhanced Security in Managing Personal Data by Web Systems for the Benefit of Individual Persons, Companies and Governments</p>	<ul style="list-style-type: none"> • Attest of an increase in privacy protection and reduction of breaches and data disclosures • Availability of all required results • Prototypical integration of results into a demonstrator that proves the feasibility of the underlying concepts and illustrates the benefits of an enhanced security in managing personal data 	<ul style="list-style-type: none"> • User studies and reports on analyzed risks and threats • Integration and operational acceptance tests 	<ul style="list-style-type: none"> • Individual persons apply security enhancements for personal data management • Companies employ enhancements towards taking greater account of security aspects during design and runtime of web systems • Governments keep up with their support towards enabling more self-determined control about personal data for individual persons
<p>Result 1 Improved Modeling of Security Aspects for Web Systems</p>	<ul style="list-style-type: none"> • Successful implementation of all necessary activities in terms of modeling support with strong focus on security-related aspects 	<ul style="list-style-type: none"> • Proof of concept • Unit and operational acceptance tests 	<ul style="list-style-type: none"> • Companies use extended means for modeling web systems with special focus on security-related aspects

Continued on next page

Table A.2 — continued from previous page

Project Description	Success Indicators	Verification Sources	Assumptions
<p>Result 2 Reduced Need for Accumulation of Personal Data by Third Parties</p>	<ul style="list-style-type: none"> • Successful implementation of all necessary activities in terms of an approach for accessing personal data that enables more control for individual persons by taking account of their preferences towards protection 	<ul style="list-style-type: none"> • Proof of concept • Unit and operational acceptance tests 	<ul style="list-style-type: none"> • Companies employ enhancements towards reducing necessitation of personal data accumulation
<p>Result 3 Extended Means for Control and Protection of Personal Data</p>	<ul style="list-style-type: none"> • Successful implementation of all necessary activities in terms of extended protective means to safeguard personal data 	<ul style="list-style-type: none"> • Proof of concept • Unit and operational acceptance tests 	<ul style="list-style-type: none"> • Individual persons apply security enhancements for personal data management • Companies employ enhancements towards protecting personal data
<p>Activity 1.1 Extend Means for Modeling Secure Web Systems</p>			<ul style="list-style-type: none"> • Companies use extended means for modeling web systems with special focus on security-related aspects

Continued on next page

Table A.2 — continued from previous page

Project Description	Success Indicators	Verification Sources	Assumptions
<p>Activity 2.1 Offer Alternative to Customer Lock-in</p>			<ul style="list-style-type: none"> • Individual persons regain ownership and self-manage their personal data by maintaining a larger set of identity attributes yet with a lower amount of distributed copies • Companies open up for universal identification, extended identity attributes, linkage and discovery across web systems, web applications and web services • Companies settle with just obtaining access to user-generated contents rather than demanding storage within own premises and, hence, also giving up own application-specific facilities for controlling access to personal user data • Governments keep up with their support towards enabling more self-determined control about personal data for individual persons
<p>Activity 3.1 Improve Control of Identity Based on Individual Context</p>			<ul style="list-style-type: none"> • Individual persons support detection and use of individual context and conditions • Individual persons employ measures for extended access control • Individual persons make additional efforts for protecting their identity data • Companies integrate means into their web systems for delegated access whilst obeying the scope defined by delegators

Continued on next page

Table A.2 — continued from previous page

Project Description	Success Indicators	Verification Sources	Assumptions
<p>Activity 3.2 Mitigate Risk of Identity Theft and Tampering of Personal Data</p>			<ul style="list-style-type: none"> • Individual persons employ measures for integrity protection • Individual persons make additional efforts for protecting their identity data • Companies integrate means for integrity protection into their web systems
<p>Activity 3.3 Increase Range and Granularity of Access Control</p>			<ul style="list-style-type: none"> • Individual persons employ measures for extended access control • Individual persons make additional efforts for protecting their identity data • Companies settle with just obtaining access to user-generated contents rather than demanding storage within own premises and, hence, also giving up <i>own</i> application-specific facilities for controlling access to personal user data

Glossary

Access Control Process of determining which principals have what kind of access to which resources based on a set of rules.

Agent “Entity that acts on behalf of another entity” (ITU, 2010).

Anonymity State of being unable to identify a particular entity among other entities.

Attack Malicious attempt to compromise security.

Attribute Contextual characteristic of an entity specified through a type/-value statement.

Authentication Process of achieving “sufficient confidence in the binding between the entity and the presented identity” (ITU, 2010).

Authorization Process of granting a principal the privilege to pass through safeguard measures taken against entities having insufficient permissions.

Breach Incident resulting in exposure of data.

Company Organization that provides something, e.g., goods or services, in exchange for a compensation, e.g., money.

Context “Environment with defined boundary conditions” (ITU, 2010).

Credential Evidence for a claimed identity.

Data Disclosure Breach with confirmed data revelation to an unauthorized party.

Delegate Agent entrusted with authority, responsibility or a function.

Delegation Process of assigning a delegate to do something on behalf of a delegator.

Delegator Entity that initiates a delegation.

Entity “Something that has a separate and distinct existence and that can be identified in context” (ITU, 2010).

Government Political organization which exercises control and makes decisions for persons, called citizens, who usually occupy and legally belong to a specific territory, called state.

Identification Process of entity recognition within context by a set of attributes.

Identifier Particular subset of attributes enabling to distinctly recognize an entity.

Identity Representation of an entity within context using a set of attributes.

Identity (Service) Provider Service provider that manages identities for other entities or puts them in the position to do so on their own.

Identity Management “Set of functions and capabilities used for assurance of identity information [...] and supporting business and security applications” (ITU, 2010).

Identity Management System System that allows identity management.

Identity Owner Entity an identity is issued to.

Incident Security event compromising an information asset.

Information Security “Protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability” (CNSS, 2010).

Ontology “[C]omplex, and possibly quite formal collection of terms”, yet not clearly distinguishable from vocabulary that refers to a possibly less formal or more loose collection of terms (W3C, 2015b).

Organization Entity formed by a group of persons to pursue a collective purpose.

Personal Data Any information, including PII as well as information on social capital, relating to a human entity, where this particular entity is either identified or identifiable without unreasonable effort (cf. identification).

Personally Identifiable Information Any information that allows for detecting the identity of an individual entity.

Principal “Entity whose identity can be authenticated” (ITU, 2010).

Privacy Privilege to control what personal data may be accessed by third parties for which purposes.

Privilege Right that allows an entity to perform an action.

Project “Temporary endeavor undertaken to create a unique product, service, or result” (PMI, 2009).

Protection State of keeping something safe from adverse effects.

RDF Triple Descriptive statement about a resource using subject-predicate-object logic.

Relying Party Service provider that relies on the identity claimed by a requesting entity.

Requesting Entity Entity that claims an identity to a RP in the context of a request.

Resource Entity that provides something of use for someone.

Security Cf. information security (for a definition of security specific to the domain of this work).

Security Assertion “Assertion that is scrutinized in the context of a security architecture” (Hodges et al., 2005).

- Self-Asserted Identity** Identity that an entity declares to be its own.
- Service Provider** Entity that offers a service via web-based means (cf. resource, web application and web service).
- Single Sign-On** Option that allows a user for plausibly identifying to a federation of multiple web applications and web services with a single authentication.
- Social Age** Age characterized by the adoption of the outcomes of the information age with a strong emphasis on social interactions between entities over the WWW.
- Social Capital** Type of capital resulting from an entity's position within a social network as well as the quantity and the quality of social relationships an entity maintains.
- Social Media** Set of resources human entities bear social relations to.
- Social Web** Set of social relationships among entities over the WWW.
- Stakeholder** Entity that has a stake in something, e.g., outcome of a project.
- Third-Party Asserted Identity** Identity that a trusted authority assigns an entity after successfully confirming relevant identity claims made by this particular entity.
- Trust** "The firm belief in the reliability and truth of information or in the ability and disposition of an entity to act appropriately, within a specified context" (ITU, 2010).
- User** "Entity that makes use of a resource" (ITU, 2010).
- User Agent** Entity that acts on behalf of another entity in order to make use of resources.
- User-Generated Content** All kinds of social media content produced by users of social web applications.
- Vocabulary** Cf. ontology.
- Web** Cf. WWW.

Web Application Entity that runs on a web server and is accessible over networks via standardized interfaces by user agents including web browsers and other services.

Web Engineering “Application of systematic and quantifiable approaches (concepts, methods, techniques, tools) to cost-effective requirement analysis, design, implementation, testing, operation, and maintenance of high-quality web applications” (Kappel et al., 2006).

Web Service Entity that is “designed to support interoperable machine-to-machine interaction over a network” (W3C, 2004).

Web System Entity that orchestrates other (web) entities, like web applications and web services, to provide one or multiple more complex services with extended functionality.

Bibliography

Printed References

- Akhawe, D.; Li, F.; He, W.; Saxena, P.; Song, D. (2013): *Data-Confined HTML5 Applications*. Tech. rep. Electrical Engineering and Computer Sciences, University of California at Berkeley. ISBN: UCB/EECS-2013-20.
- Anderson, R. J. (2008): *Security Engineering. A Guide to Building Dependable Distributed Systems*. 2nd. Wiley. ISBN: 978-0-470-06852-6.
- Ast, M.; Wild, S.; Gaedke, M. (2013): “The SWAC Approach for Sharing a Web Application’s Codebase Between Server and Client”. In: *Web Engineering*. Ed. by Daniel, F.; Dolog, P.; Li, Q. Vol. 7977. Lecture Notes in Computer Science. Heidelberg: Springer, pp. 84–98. ISBN: 978-3-642-39199-6.
- Ast, M.; Wild, S.; Gaedke, M. (2014): “Efficient Development of Progressively Enhanced Web Applications by Sharing Presentation and Business Logic Between Server and Client”. In: *Journal of Web Engineering 13.3 & 4: Component-Based, Client-Oriented Web Engineering. Issues, Advancements and Opportunities*. Ed. by Daniel, F.; Dolog, P.; Li, Q., pp. 223–242. ISSN: 1540-9589.
- Azua, M. (2009): *The Social Factor. Innovate, Ignite, and Win through Mass Collaboration and Social Networking*. IBM Press. ISBN: 978-0-13-701890-1.

- Baden, R.; Bender, A.; Spring, N.; Bhattacharjee, B.; Starin, D. (2009): “Persona: An Online Social Network with User-defined Privacy”. In: *SIGCOMM Computer Communication Review* 39.4, pp. 135–146. ISSN: 0146-4833.
- Bai, G.; Lei, J.; Meng, G.; Venkatraman, S. S., et al. (2013): “AuthScan: Automatic Extraction Of Web Authentication Protocols from Implementations”. In: *Proceedings of 20th Annual Network & Distributed System Security Symposium*. The Internet Society.
- Barisch, M. A. (2012): “Design and Evaluation of a System to Extend Identity Management to Multiple Devices”. PhD thesis. Institut für Kommunikationsnetze und Rechnersysteme der Universität Stuttgart.
- Barker, E.; Barker, W.; Burr, W.; Polk, W.; Smid, M. (2012): *Recommendation for Key Management - Part 1: General*. Revision 3. NIST Special Publication 800-57. National Institute of Standards and Technology (NIST).
- Bell, G. (2009): *Building Social Web Applications*. Ed. by Laurent, S. S. First Edition. O'Reilly. ISBN: 978-0-596-51875-2.
- Bertino, E.; Martino, L. D. (2007): “A Service-Oriented Approach to Security - Concepts and Issues”. In: *Proceedings of the 11th IEEE International Workshop on Future Trends of Distributed Computing Systems*. IEEE, pp. 31–40. ISBN: 0-7695-2810-4.
- Bielenberg, A.; Helm, L.; Gentilucci, A.; Stefanescu, D.; Zhang, H. (2012): “The Growth of Diaspora - A Decentralized Online Social Network in the Wild”. In: *Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, pp. 13–18. ISBN: 978-1-4673-1016-1.
- Bojars, U.; Passant, A.; Breslin, J. G.; Decker, S. (2008): “Social Network and Data Portability using Semantic Web Technologies”. In: *BIS 2008 Workshops Proceedings. 2nd Workshop on Social Aspects of the Web (SAW 2008)*. Ed. by Flejter, D.; Grzonkowski, S.; Kaczmarek, T.; Kowalkiewicz, M., et al. Vol. 333. CEUR Workshop Proceedings. ISSN: 1613-0073. Department of Information Systems, Poznań University of Economics, pp. 5–19.

- Bonneau, J.; Anderson, J.; Anderson, R.; Stajano, F. (2009): “Eight Friends are Enough: Social Graph Approximation via Public Listings”. In: *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*. SNS '09. ACM, pp. 13–18. ISBN: 978-1-60558-463-8.
- Brambilla, M.; Fraternali, P. (2014): “Large-Scale Model-Driven Engineering of Web User Interaction: The WebML and WebRatio Experience”. In: *Science of Computer Programming 89*, Part B. Special Issue on Success Stories in Model Driven Engineering, pp. 71–87. ISSN: 0167-6423.
- Braune, F.; Wild, S.; Gaedke, M. (2014): “Using Linked Data for Modeling Secure Distributed Web Applications and Services”. In: *Web Engineering*. Ed. by Casteleyn, S.; Rossi, G.; Winckler, M. Vol. 8541. Lecture Notes in Computer Science. Springer, pp. 540–544. ISBN: 978-3-319-08244-8.
- Bria, F. (2012): “New Governance Models Towards a Open Internet Ecosystem for Smart Connected European Cities and Regions”. In: *Open Innovation 2012*. European Union. Chap. 1 - Policy Development, pp. 62–71. ISBN: 978-92-79-21461-5.
- Burke, M.; Kraut, R.; Marlow, C. (2011): “Social Capital on Facebook: Differentiating Uses and Users”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. New York, NY, USA: ACM, pp. 571–580. ISBN: 978-1-4503-0228-9.
- Busch, M.; Koch, N.; Masi, M.; Pugliese, R.; Tiezzi, F. (2012): “Towards Model-Driven Development of Access Control Policies for Web Applications”. In: *Proceedings of the Workshop on Model-Driven Security*. ACM, pp. 1–6. ISBN: 978-1-4503-1806-8.
- Caronni, G. (2000): “Walking the Web of Trust”. In: *IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2000)*. IEEE, pp. 153–158. ISBN: 0-7695-0798-0.
- Carroll, J. J. (2003): “Signing RDF Graphs”. In: *The Semantic Web - ISWC 2003*. Ed. by Fensel, D.; Sycara, K.; Mylopoulos, J. Vol. 2870. Lecture Notes in Computer Science. Springer, pp. 369–384. ISBN: 978-3-540-20362-9.
- Ceri, S.; Fraternali, P.; Bongio, A. (2000): “Web Modeling Language (WebML): A Modeling Language for Designing Web Sites”. In: *Computer Networks* 33.1–6, pp. 137–157. ISSN: 1389-1286.

- Chudnovskyy, O.; Nestler, T.; Daniel, M. G. F.; Fernández-Villamor, J. I., et al. (2012): “End-User-Oriented Telco Mashups: The OMELETTE Approach”. In: *Proceedings of the 21st International Conference Companion on World Wide Web. WWW '12 Companion*. ACM, pp. 235–238. ISBN: 978-1-4503-1230-1.
- Chudnovskyy, O.; Wild, S.; Gebhardt, H.; Gaedke, M. (2012): “Data Portability Using WebComposition/Data Grid Service”. In: *International Journal on Advances in Internet Technology* 4.3 & 4, pp. 123–132. ISSN: 1942-2652.
- CNSS, ed. (2010): *National Information Assurance (IA) Glossary. CNSS Instruction No. 4009*. Committee on National Security Systems (CNSS).
- Creswell, J. W. (2012): *Educational Research. Planning, Conducting and Evaluating Quantitative and Qualitative Research*. Fourth Edition. Pearson. ISBN: 978-0-13-136739-5.
- Dane, K. (2012): “Considering Data Breaches: Public Information, Corporate Responsibility, and Market Valuations”. Master Thesis. University of Washington.
- Daugherty, T.; Eastin, M. S.; Bright, L. (2008): “Exploring Consumer Motivations for Creating User-Generated Content”. In: *Journal of Interactive Advertising* 8.2, pp. 16–25. ISSN: 1525-2019.
- Dhamija, R.; Dussault, L. (2008): “The Seven Flaws of Identity Management: Usability and security Challenges”. In: *IEEE Security & Privacy* 6.2, pp. 24–29. ISSN: 1540-7993.
- Díaz, O.; Irastorza, A.; Cuadrado, J. S.; Alonso, L. M. (2008): “From page-centric to portlet-centric Web development: Easing the transition using MDD”. In: *Information and Software Technology* 50.12, pp. 1210–1231. ISSN: 0950-5849.
- Dietz, M.; Wallach, D. S. (2014): “Hardening Persona—Improving Federated Web Login”. In: *21st Annual Network and Distributed System Security Symposium, NDSS 2014*. ISBN: 1-891562-35-5.
- Dinger, J.; Hartenstein, H. (2008): *Netzwerk- und IT-Sicherheitsmanagement. Eine Einführung*. German. Universitätsverlag Karlsruhe. ISBN: 978-3-86644-209-2.
- EC (2004): *Project Cycle Management Guidelines*. Volume 1. European Commission (EC).

- EC (2011): *Attitudes on Data Protection and Electronic Identity in the European Union*. Special Eurobarometer Report 359. Wave 74.3. European Commission (EC).
- El Maliki, T.; Seigneur, J.-M. (2007): "A Survey Of User-centric Identity Management Technologies". In: *The International Conference on Emerging Security Information, Systems, and Technologies, 2007. SecureWare 2007*. Ed. by Peñalver, L.; Dini, O.; Mulholland, J.; Nieto-Taladriz, O. IEEE, pp. 12–17. ISBN: 978-0-7695-2989-9.
- Ellis, T. J.; Levy, Y. (2009): "Towards a Guide for Novice Researchers on Research Methodology: Review and Proposed Methods". In: *Issues in Informing Science and Information Technology* 6, pp. 323–337. ISSN: 1547-5840.
- Ellison, C.; Schneier, B. (2000): "Ten Risks of PKI: What You're not Being Told about Public Key Infrastructure". In: *Computer Security* 16.1, pp. 1–7.
- Ellison, N. B.; Steinfield, C.; Lampe, C. (2011): "Connection Strategies: Social Capital Implications of Facebook-enabled Communication Practices". In: *New Media & Society* XX.X, pp. 1–20.
- European Parliament (1995): "EU Directive 95/46/EC - The Data Protection Directive. Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of Such Data". In: *Official Journal of the European Communities* 281. Ed. by The European Parliament and the Council of the European Union, pp. 31–50.
- Fielding, R. T. (2000): "Architectural styles and the design of network-based software architectures". PhD thesis. University of California, Irvine.
- Florêncio, D.; Herley, C. (2007): "A Large-scale Study of Web Password Habits". In: *Proceedings of the 16th international conference on World Wide Web*. ACM Press, pp. 657–666. ISBN: 978-1-59593-654-7.
- FRA (2014): *Handbook on European Data Protection Law*. Ed. by European Union Agency for Fundamental Rights (FRA). 2nd Edition. ISBN: 978-92-9239-461-5.
- Gaedke, M. (2000): *Komponententechnik für Entwicklung und Evolution von Anwendungen im World Wide Web*. German. Shaker. ISBN: 3-8265-8059-1.

- Gaedke, M.; Gräf, G. (2000): “WebComposition Process Model: Ein Vorgehensmodell zur Entwicklung und Evolution von Web-Anwendungen”. German. In: *Tagungsband 2. Workshop Komponentenorientierte betriebliche Anwendungssysteme (WKBA 2)*. Ed. by Flatscher, R. G.; Turowski, K. Vienna, pp. 21–38.
- Gaedke, M.; Gräf, G. (2001): “Development and Evolution of Web-Applications Using the WebComposition Process Model”. English. In: *Web Engineering*. Ed. by Murugesan, S.; Deshpande, Y. Vol. 2016. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 58–76. ISBN: 978-3-540-42130-6.
- Gaedke, M.; Rehse, J. (2000): “Supporting Compositional Reuse in Component-based Web Engineering”. In: *Proceedings of the 2000 ACM Symposium on Applied Computing - Volume 2. SAC '00*. ACM, pp. 927–933. ISBN: 1-58113-240-9.
- Gangewere, W. (2013): “Assessing the Impact of a Privacy Breach on a Firm’s Market Value”. Bachelor Thesis. Duquesne University.
- Garzotto, F; Paolini, P; Schwabe, D. (1993): “HDM - A Model-Based Approach to Hypertext Application Design”. In: *ACM Transactions on Information Systems (TOIS)* 11.1, pp. 1–26. ISSN: 1046-8188.
- Gellersen, H.-W.; Gaedke, M. (1999): “Object-Oriented Web Application Development”. In: *IEEE Internet Computing* 3.1, pp. 60–68. ISSN: 1089-7801.
- Hackett, M.; Hawkey, K. (2012): “Security, Privacy and Usability Requirements for Federated Identity”. In: *Workshop on Web 2.0 Security & Privacy*.
- Halpin, H. (2014): *D4.1 - State of the Art of Social Networking Systems, Identity Ecosystem and Social Data Stores*. Deliverable. Version 8. FP7 - CAPS - 2013, Project no. 610349. D-CENT Decentralised Citizens ENgagement Technologies.
- Heil, A. (2012): *Anwendungsentwicklung für Intelligente Umgebungen im Web Engineering*. German. Springer Fachmedien Wiesbaden. ISBN: 978-3-8348-2550-6.
- Heil, S.; Wild, S.; Gaedke, M. (2014a): “Collaborative Adaptive Case Management with Linked Data”. In: *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*. WWW Companion '14. Seoul, Korea: International World Wide Web Conferences Steering Committee, pp. 99–102. ISBN: 978-1-4503-2745-9.

- Heil, S.; Wild, S.; Gaedke, M. (2014b): “CRAWL-E: Distributed Skill Endorsements in Expert Finding”. In: *Web Engineering*. Ed. by Casteleyn, S.; Rossi, G.; Winckler, M. Vol. 8541. Lecture Notes in Computer Science. Springer, pp. 57–75. ISBN: 978-3-319-08244-8.
- Heitmann, B.; Kim, J. G.; Passant, A.; Hayes, C.; Kim, H.-G. (2010): “An Architecture for Privacy-enabled User Profile Portability on the Web of Data”. In: *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*. HetRec '10. ACM, pp. 16–23. ISBN: 978-1-4503-0407-8.
- Hollenbach, J.; Presbrey, J.; Berners-Lee, T. (2009): “Using RDF Metadata To Enable Access Control on the Social Semantic Web”. In: *Proceedings of the Workshop on Collaborative Construction, Management and Linking of Structured Knowledge (CK2009)* 514.
- Holt, T. J.; Smirnova, O. (2014): *Examining the Structure, Organization, and Processes of the International Market for Stolen Data*. Report (Study/Research) ; Research (Applied/Empirical), p. 156.
- Houben, G.-J.; Barna, P.; Frasincar, F.; Vdovjak, R. (2003): “Hera: Development of Semantic Web Information Systems”. English. In: *Web Engineering*. Ed. by Lovelle, J. M. C.; Rodríguez, B. M. G.; Gayo, J. E. L.; Puerto Paule Ruiz, M. del; Aguilar, L. J. Vol. 2722. Lecture Notes in Computer Science. Springer, pp. 529–538. ISBN: 978-3-540-40522-1.
- Hubbard, D. W. (2009): *The Failure of Risk Management. Why It's Broken and How to Fix It*. John Wiley & Sons. ISBN: 978-0-470-38795-5.
- Hühnlein, D.; Hornung, G.; Kubach, M.; Mladenov, V., et al. (2014): “SkIDentity - Trusted Identities for the Cloud”.
- Hühnlein, D.; Schmölz, J.; Drabik, M.; Ituarte, N., et al. (2014): *Reference Architecture*. Deliverable D21.4. Version 1.2. FutureID. Final.
- Hühnlein, D.; Wich, T.; Schmölz, J.; Haase, H.-M. (2014): “The Evolution of Identity Management Using the Example of Web-based Applications”. In: *it - Information Technology* 56 (3). Ed. by Molitor, P., pp. 134–140. ISSN: 1611-2776.
- IIBA (2009): *A Guide to the Business Analysis Body of Knowledge (BABOK Guide). Version 2.0*. International Institute of Business Analysis (IIBA), Toronto, Ontario, Canada. ISBN: 978-0-9811292-1-1.

- ISO/IEC, ed. (2001): *Software Engineering. Product Quality. Part 1: Quality Model*. ISO/IEC 9126-1:2001.
- ISO/IEC, ed. (2008): *Software Engineering. Software Quality Requirements and Evaluation (SQuaRE). Data Quality Model*. ISO/IEC 25012.
- ISO/IEC, ed. (2011): *Systems and Software Engineering. Systems and Software Quality Requirements and Evaluation (SQuaRE). System and Software Quality Models*. ISO/IEC 25010:2011.
- ITU, T. S. S. of, ed. (2010): *Series X: Data Networks, Open System Communications and Security. Cyberspace Security - Identity Management. Recommendation ITU-T X.1252. Baseline of Identity Management Terms and Definitions*.
- Jahid, S.; Nilizadeh, S.; Mittal, P; Borisov, N.; Kapadia, A. (2012): “DECENT: A Decentralized Architecture for Enforcing Privacy in Online Social Networks”. In: *Pervasive Computing and Communications Workshops (PERCOM Workshops)*. IEEE, pp. 326–332. ISBN: 978-1-4673-0905-9.
- Jones, M. B.; McIntosh, M. (2008): *Identity Metasystem Interoperability*. Committee Draft 1. Version 1.0. OASIS.
- Jordan, K.; Hauser, J.; Foster, S. (2003): “The Augmented Social Network: Building Identity and Trust into the Next-Generation Internet”. In: *First Monday* 8.8. ISSN: 1396-0466.
- Jøsang, A. (2014): “Identity Management and Trusted Interaction in Internet and Mobile Computing”. In: *IET Information Security* 8 (2), pp. 67–79. ISSN: 1751-8709.
- Jøsang, A.; Rosenberger, C.; Miralabé, L.; Klevjer, H., et al. (2015): “Local User-Centric Identity Management”. In: *Journal of Trust Management* 2.1, pp. 1–28. ISSN: 2196-064X.
- Jøsang, A.; Zomai, M. A.; Suriadi, S. (2007): “Usability and Privacy in Identity Management Architectures”. In: *Proceedings of the Fifth Australasian Symposium on ACSW Frontiers*. Vol. 68. Australian Computer Society, pp. 143–152. ISBN: 1-920-68285-X.

- Kaplan, A. M.; Haenlein, M. (2010): “Users of the world, unite! The challenges and opportunities of Social Media”. In: *Business Horizons* 53.1, pp. 59–68. ISSN: 0007-6813.
- Kappel, G.; Pröll, B.; Reich, S.; Retschitzegger, W. (2006): *Web Engineering. The Discipline of Systematic Development of Web Applications*. John Wiley & Sons. ISBN: 978-0-470-01554-4.
- Kasten, A.; Scherp, A. (2013): *Iterative Signing of RDF(S) Graphs, Named Graphs, and OWL Graphs: Formalization and Application*. Tech. rep. 3. Universität Koblenz-Landau, Koblenz, Germany, pp. 3–28.
- Kietzmann, J. H.; Hermkens, K.; McCarthy, I. P.; Silvestre, B. S. (2011): “Social media? Get serious! Understanding the functional building blocks of social media”. In: *Business Horizons* 54.3, pp. 241–251. ISSN: 0007-6813.
- Koch, N.; Knapp, A.; Zhang, G.; Hubert (2008): “UML-based Web Engineering”. In: *Web Engineering: Modelling and Implementing Web Applications*. Springer, pp. 157–191.
- Koch, N.; Meliá-Beigbeder, S.; Moreno-Vergara, N.; Pelechano-Ferragud, V., et al. (2008): “Model-Driven Web Engineering”. In: *Upgrade - The European Journal for the Informatics Professional* 9.2, pp. 40–45.
- Krishnamurthy, B.; Wills, C. E. (2008): “Characterizing Privacy in Online Social Networks”. In: *Proceedings of the First Workshop on Online Social Networks*. WOSN '08. New York, NY, USA: ACM, pp. 37–42. ISBN: 978-1-60558-182-8.
- Lee, Y.-J.; Kim, C.-S. (2010): “Building Semantic Ontologies for RESTful Web Services”. In: *International Conference on Computer Information Systems and Industrial Management Applications (CISIM)*. IEEE, pp. 383–386. ISBN: 978-1-4244-7818-7.
- Leedy, P. D.; Ormrod, J. E. (2010): *Practical Research*. Ninth Edition. Pearson. ISBN: 978-0-13-715242-1.
- Maamar, Z.; Hacid, H.; Huhns, M. N. (2011): “Why Web Services Need Social Networks”. In: *IEEE Internet Computing* 15.2, pp. 90–94. ISSN: 1089-7801.

- Maler, E.; Reed, D. (2008): “The Venn of Identity: Options and Issues in Federated Identity Management”. In: *IEEE Security & Privacy* 6.2, pp. 16–23. ISSN: 1540-7993.
- Mayer, A. (2013): “On the Security of Web Single Sign-On”. PhD thesis. Fakultät für Elektrotechnik und Informationstechnik, Ruhr-Universität Bochum.
- McCallister, E.; Grance, T.; Scarfone, K. (2010): *Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)*. NIST Special Publication 800-122. National Institute of Standards and Technology (NIST).
- Meinecke, J. (2008): “Unterstützung der Evolution Föderativer Systeme im Web Engineering”. German. PhD thesis. Fakultät für Informatik, Technische Universität Chemnitz.
- Meinecke, J.; Gaedke, M. (2005): “Modeling Federations of Web Applications with WAM”. In: *Third Latin American Web Congress, LA-WEB 2005*. IEEE, pp. 23–31. ISBN: 0-7695-2471-0.
- Meinecke, J.; Gaedke, M.; Thiele, C. (2007): “Enabling Architecture Changes in Distributed Web-Applications”. In: *Web Conference, 2007. LA-WEB 2007. Latin American*. IEEE, pp. 92–99. ISBN: 978-0-7695-3008-6.
- Meliá, S.; Gómez, J. (2005): “Applying Transformations to Model Driven Development of Web Applications”. In: *Perspectives in Conceptual Modeling*. Ed. by Akoka, J.; Liddle, S. W.; Song, I.-Y.; Bertolotto, M.; Comyn-Wattiau, I., et al. Vol. 3770. Lecture Notes in Computer Science. Springer, pp. 63–73. ISBN: 978-3-540-29395-8.
- Miklič, P. (2008): *Guide to the Logical Framework Approach: A Key Tool to Project Cycle Management*. Sector for Programming and Management of the EU Fund and Development. DIAL. ISBN: 978-86-87219-01-4.
- Müller, L. (2008): “Authentication and Transaction Security in E-business”. In: *The Future of Identity in the Information Society*. Ed. by Fischer-Hübner, S.; Duquenoy, P.; Zuccato, A.; Leonardo. Vol. 262. IFIP — The International Federation for Information Processing. Springer, pp. 175–197. ISBN: 978-1-4419-4629-4.
- Nelson, T. H. (1994): *Literary Machines*. Mindful Press. ISBN: 978-0893470562.
- Nielsen and NM Incite (2012): *State of the Media: Social Media Report 2012*.

- NORAD (1999): *The Logical Framework Approach (LFA): Handbook for objectives-oriented planning*. 4th Edition. Norwegian Agency for Development Cooperation (NORAD). ISBN: 82-7548-160-0.
- OECD, ed. (2013): *The OECD Privacy Framework*. Organisation for Economic Co-operation and Development (OECD).
- Papazoglou, M. P.; Traverso, P.; Dustdar, S.; Leymann, F. (2007): “Service-Oriented Computing: State of the Art and Research Challenges”. In: *Computer* 40.11, pp. 38–45. ISSN: 0018-9162.
- Pérez, J.; Arenas, M.; Gutierrez, C. (2009): “Semantics and Complexity of SPARQL”. In: *ACM Transactions on Database Systems* 34.3. Ed. by Özsoyoğlu, Z. M. et al., pp. 1–45. ISSN: 0362-5915.
- PMI (2009): *A Guide to the Project Management Body of Knowledge*. Fourth. Project Management Institute (PMI). ISBN: 978-1-933890-51-7.
- Porter, J. (2008): *Designing for the Social Web*. Ed. by Nolan, M. J. New Riders. ISBN: 978-0-321-53492-7.
- Pronin, E.; Gilovich, T.; Ross, L. (2004): “Objectivity in the Eye of the Beholder: Divergent Perceptions of Bias in Self Versus Others”. In: *Psychological Review* 111.3, pp. 781–799. ISSN: 0033-295X.
- Rafique, I.; Lew, P.; Abbasi, M. Q.; Li, Z. (2012): “Information Quality Evaluation Framework: Extending ISO 25012 Data Quality Model”. In: *International Science Index* 65 (2012) 6.5, pp. 501–506. ISSN: 1307-6892.
- Rienäcker, M.; Wild, S.; Gaedke, M. (2014): “Building Bridges between Diverse Identity Concepts Using WebID”. In: *Web Engineering*. Ed. by Casteleyn, S.; Rossi, G.; Winckler, M. Vol. 8541. Lecture Notes in Computer Science. Springer, pp. 498–502. ISBN: 978-3-319-08244-8.
- Rivero, J. M.; Grigera, J.; Rossi, G.; Luna, E. R.; Nora (2012): “Towards Agile Model-Driven Web Engineering”. In: *IS Olympics: Information Systems in a Diverse World. CAiSE Forum 2011, London, UK, June 20-24, 2011, Selected Extended Paper*. Ed. by Nurcan, S. Vol. 107. Lecture Notes in Business Information Processing. Springer, pp. 142–155. ISBN: 978-3-642-29748-9.

- Röder, H.; Franke, S.; Müller, C.; Przybylski, D. (2009): “Ein Kriterienkatalog zur Bewertung von Anforderungsspezifikationen”. German. In: *Softwaretechnik-Trends* 29.4. ISSN: 0720-8928.
- Rossi, G.; Schwabe, D. (2008): “Modeling and Implementing Web Applications with OO-HDM”. In: *Web Engineering: Modelling and Implementing Web Applications*. Ed. by Rossi, G.; Pastor, O.; Schwabe, D.; Olsina, L. Human-Computer Interaction Series. Springer, pp. 109–155. ISBN: 978-1-84628-922-4.
- Saleem, M. Q.; Jaafar, J.; Hassan, M. F. (2014): “Model Driven Software Development: An Overview”. In: *2014 International Conference on Computer and Information Sciences (ICCOINS)*. IEEE, pp. 1–5. ISBN: 978-1-4799-4391-3.
- Satzger, B.; Zabolotnyi, R.; Dustdar, S.; Wild, S., et al. (2014): “Toward Collaborative Software Engineering Leveraging the Crowd”. In: *Economics-Driven Software Architecture*. Ed. by Mistrik, I.; Bahsoon, R.; Kazman, R.; Zhang, Y. 1st Edition. Elsevier. Chap. 8, pp. 159–182. ISBN: 978-0-12-410464-8.
- Sauermann, L.; Cyganiak, R.; Völkel, M. (2007): “Cool URIs for the Semantic Web”. In: Technical memo / Deutsches Forschungszentrum für Künstliche Intelligenz 07.01. ISSN: 0946-0071.
- Sayers, C.; Eshghi, K. (2002): *The Case for Generating URIs by Hashing RDF Content*. Tech. rep. Intelligent Enterprise Technology Laboratory, HP Laboratories, Palo Alto, USA. ISRN: HPL-2002-216.
- Schill, A.; Springer, T. (2012): *Verteilte Systeme. Grundlagen und Basistechnologien*. German. eXamen.press. Springer. ISBN: 978-3-642-25795-7.
- Scholtz, A.; Wild, S.; Gaedke, M. (2015a): “Scope-Aware Delegations in Distributed Social Networks”. In: *Engineering in the Web in the Big Data Era*. Ed. by Cimiano, P. et al. Vol. 9114. Lecture Notes in Computer Science. Springer, pp. 709–712. ISBN: 978-3-319-19890-3.
- Scholtz, A.; Wild, S.; Gaedke, M. (2015b): “Systematic Composition of Web-based Applications with Focus on Security”. In: *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services*. iiWAS '15. ACM, pp. 637–641. ISBN: 978-1-4503-3491-4.

- Speicher, M. (2015): *What Is Usability? A Characterization Based on ISO 9241-11 and ISO/IEC 25010*. Tech. rep. Fakultät für Informatik, Technische Universität Chemnitz. ISBN: CSR-15-02.
- Stallings, W. (2010): *Cryptography and Network Security: Principles and Practice*. 5th. Prentice Hall Press. ISBN: 978-0-13-609704-4.
- Stufflebeam, D. L. (2001): "Evaluation Models". In: *New Directions for Evaluation* 89, pp. 7–98. ISSN: 1534-875X.
- Sun, S.-T.; Boshmaf, Y.; Hawkey, K.; Beznosov, K. (2010): "A Billion Keys, but Few Locks: The Crisis of Web Single Sign-on". In: *Proceedings of the 2010 Workshop on New Security Paradigms*. NSPW '10. New York, NY, USA: ACM, pp. 61–72. ISBN: 978-1-4503-0415-3.
- Symantec (2014): *Internet Security Threat Report 2014. 2013 Trends*. Volume 19.
- The White House, ed. (2011): *National Strategy for Trusted Identities in Cyberspace*. The White House, Washington.
- Tomaszok, D.; Gaedke, M.; Gebhardt, H. (2011): "WebID+ACO: A Distributed Identification Mechanism for Social Web". In: *Proceedings of the Federated Social Web Europe*.
- Tomaszok, D.; Rybiński, H. (2011): "OAuth+UAO: A Distributed Identification Mechanism for Triplestores". In: *Computational Collective Intelligence. Technologies and Applications*. Ed. by Jędrzejowicz, P.; Nguyen, N. T.; Hoang, K. Vol. 6922. Lecture Notes in Computer Science. Springer, pp. 275–284. ISBN: 978-3-642-23934-2.
- Tramp, S.; Frischmuth, P.; Ermilov, T.; Shekarpour, S.; Auer, S. (2012): "An Architecture of a Distributed Semantic Social Network". In: *Semantic Web 5.1*, pp. 77–95. ISSN: 1570-0844.
- Tramp, S.; Story, H.; Samba, A.; Frischmuth, P., et al. (2012): "Extending the WebID Protocol with Access Delegation". In: *Proceedings of the Third International Workshop on Consuming Linked Data (COLID2012)*. Vol. 905. CEUR-WS.org, pp. 99–111.

- Troyer, O. D.; Casteleyn, S. (2003): “Modeling Complex Processes for Web Applications using WSDM”. In: *Proceedings of the 3rd International Workshop on Web-Oriented Software Technologies*. Ed. by Schwabe, D.; Pastor, O.; Rossi, G.; Olsina, L., pp. 1–12.
- Trujillo, S.; Batory, D.; Diaz, O. (2007): “Feature Oriented Model Driven Development: A Case Study for Portlets”. In: *Proceedings of the 29th International Conference on Software Engineering. ICSE '07*. IEEE Computer Society, pp. 44–53. ISBN: 0-7695-2828-7.
- Tummarello, G.; Morbidoni, C.; Puliti, P.; Piazza, F. (2005): “Signing Individual Fragments of an RDF Graph”. In: *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web. WWW '05*. ACM. ACM, pp. 1020–1021. ISBN: 1-59593-051-5.
- Vallecillo, A.; Koch, N.; Castro, C. C.; Comai, S., et al. (2007): “MDWEnet: A Practical Approach to Achieving Interoperability of Model-Driven Web Engineering Methods”. In: *7th International Conference on Web Engineering, Workshop Proceedings*. Dipartimento di Elettronica e Informazione, Politecnico di Milano, pp. 246–254. ISBN: 978-88-902405-2-2.
- Vdovjak, R.; Frasinca, F.; Houben, G.-J.; Peter (2003): “Engineering Semantic Web Information Systems in Hera”. In: *Journal of Web Engineering* 2.1, pp. 3–26. ISSN: 1540-9589.
- Venter, H. S.; Eloff, J. H. P. (2003): “A Taxonomy for Information Security Technologies”. In: *Computers & Security* 22.4, pp. 299–307. ISSN: 0167-4048.
- Webster, J. G. (2010): “User Information Regimes: How Social Media Shape Patterns of Consumption”. In: *Northwestern University Law Review* 104.2, pp. 593–612. ISSN: 0029-3571.
- WEF (2011): *Personal Data: The Emergence of a New Asset Class*. Report. World Economic Forum (WEF).

- Wild, S.; Ast, M.; Gaedke, M. (2013): “Towards a Context-Aware WebID Certificate Creation Taking Individual Conditions and Trust Needs into Account”. In: *Proceedings of the 15th International Conference on Information Integration and Web-based Applications & Services (iiWAS2013)*. Ed. by Weippl, E.; Indrawan-Santiago, M.; Steinbauer, M.; Kotsis, G.; Khalil, I. IIWAS '13. Vienna, Austria: ACM, pp. 532–541. ISBN: 978-1-4503-2113-6.
- Wild, S.; Braune, F.; Pretzsch, D.; Rienäcker, M.; Gaedke, M. (2014): “Tamper-Evident User Profiles for WebID-Based Social Networks”. In: *Web Engineering*. Ed. by Casteleyn, S.; Rossi, G.; Winckler, M. Vol. 8541. Lecture Notes in Computer Science. Springer, pp. 470–479. ISBN: 978-3-319-08244-8.
- Wild, S.; Chudnovskyy, O.; Heil, S.; Gaedke, M. (2013a): “Customized Views on Profiles in WebID-Based Distributed Social Networks”. In: *Web Engineering*. Ed. by Daniel, F.; Dolog, P.; Li, Q. Vol. 7977. Lecture Notes in Computer Science. Heidelberg: Springer, pp. 498–501. ISBN: 978-3-642-39199-6.
- Wild, S.; Chudnovskyy, O.; Heil, S.; Gaedke, M. (2013b): “Protecting User Profile Data in WebID-Based Social Networks Through Fine-Grained Filtering”. In: *Current Trends in Web Engineering*. Ed. by Sheng, Q. Z.; Kjeldskov, J. Vol. 8295. Lecture Notes in Computer Science. Springer, pp. 269–280. ISBN: 978-3-319-04243-5.
- Wild, S.; Gaedke, M. (2009): “WebComposition/EMS: A Value-Driven Approach to Evolution”. In: *ICWE'09 Doctoral Consortium*. Ed. by Rossi, G.; Iturrioz, J. CEUR Workshop Proceedings. ISSN: 1613-0073. Onekin Research Group; University of the Basque Country, pp. 39–43.
- Wild, S.; Gaedke, M. (2014): “Utilizing Architecture Models for Secure Distributed Web Applications and Services”. In: *it - Information Technology* 56.3: *Architecture of Web Application / René Peinl*. Ed. by Molitor, P, pp. 112–118. ISSN: 1611-2776.
- Wild, S.; Wiedemann, F.; Heil, S.; Tschudnowsky, A.; Gaedke, M. (2015): “ProProtect3: An Approach for Protecting User Profile Data from Disclosure, Tampering, and Improper Use in the Context of WebID”. In: *Transactions on Large-Scale Data- and Knowledge-Centered Systems*. Lecture Notes in Computer Science 8990: *Special Issue on Big Data and Open Data XIX*. Ed. by Hameurlain, A.; Küng, J.; Wagner, R.; Bianchini, D., et al., pp. 87–127. ISSN: 0302-9743.

Wimmer, M.; Schauerhuber, A.; Schwinger, W.; Kargl, H. (2007): “On the Integration of Web Modeling Languages: Preliminary Results and Future Challenges”. In: *7th International Conference on Web Engineering, Workshop Proceedings*. Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy, pp. 255–269. ISBN: 978-88-902405-2-2.

Winkels, M. (2013): *The Global Social Network Landscape. A Country-by-Country Guide to Social Network Usage*.

Yeung, C.-m. A.; Liccardi, I.; Lu, K.; Seneviratne, O.; Berners-Lee, T. (2009): “Decentralization: The Future of Online Social Networking”. In: *W3C Workshop on the Future of Social Networking Position Papers*. Vol. 2, pp. 2–7.

Online References

Anders, G. (2014): *You're Worth \$128 On Facebook; Sorry About That LinkedIn Drop*. Forbes. URL: <http://www.forbes.com/sites/georgeanders/2014/02/07/youre-worth-128-on-facebook-sorry-about-that-linkedin-drop/> (visited on Sept. 18, 2014).

Appelquist, D.; Brickley, D.; Carvaholo, M.; Iannella, R., et al. (2010): *A Standards-based, Open and Privacy-aware Social Web. W3C Incubator Group Report 6th December 2010*. Ed. by Halpin, H.; Tuffield, M. W3C. URL: <http://www.w3.org/2005/Incubator/socialweb/XGR-socialweb-20101206/> (visited on Sept. 14, 2014).

Appraisal Economics (2014): *Social Media Valuation and the Value of a User*. URL: <http://www.appraisaleconomics.com/wp-content/uploads/Social-Media-Valuation-and-the-Value-of-a-User.pdf> (visited on Mar. 18, 2016).

Bamberg, W. et al. (2013): *Mozilla Persona - Protocol Overview*. Mozilla. URL: https://developer.mozilla.org/en-US/Persona/Protocol_Overview (visited on May 30, 2014).

- Bechhofer, S.; Harmelen, F. van; Hendler, J.; Horrocks, I., et al. (2004): *OWL Web Ontology Language*. Ed. by Dean, M.; Schreiber, G. W3C. URL: <http://www.w3.org/TR/2004/REC-owl-ref-20040210/> (visited on May 24, 2015). W3C Recommendation 10 February 2004.
- Beckett, D.; Berners-Lee, T.; Prud'hommeaux, E.; Carothers, G. (2014): *RDF 1.1 Turtle. Terse RDF Triple Language*. Ed. by Prud'hommeaux, E.; Carothers, G. W3C. URL: <http://www.w3.org/TR/2014/REC-turtle-20140225/> (visited on May 31, 2015). W3C Recommendation 25 February 2014.
- Berjon, R.; Faulkner, S.; Leithead, T.; Navara, E. D., et al. (2014): *HTML5. A vocabulary and associated APIs for HTML and XHTML*. W3C. URL: <https://www.w3.org/TR/2014/REC-html5-20141028/> (visited on June 5, 2015). W3C Candidate Recommendation 28 October 2014.
- Berners-Lee, T. (2001): *Contact: Utility Concepts for Everyday Life*. W3C. URL: <http://www.w3.org/2000/10/swap/pim/contact.rdf> (visited on May 23, 2015).
- Berners-Lee, T.; Connolly, D. (2011): *Notation3 (N3): A readable RDF syntax*. W3C. URL: <http://www.w3.org/TeamSubmission/2011/SUBM-n3-20110328/> (visited on May 31, 2015). W3C Team Submission 28 March 2011.
- Boag, S.; Chamberlin, D.; Fernández, M. F.; Florescu, D., et al. (2010): *XQuery 1.0. An XML Query Language (Second Edition)*. W3C. URL: <http://www.w3.org/TR/2010/REC-xquery-20101214/> (visited on May 23, 2015). W3C Recommendation 14 December 2010.
- Bogart, N. (2013): *Over 2 million stolen Facebook, LinkedIn and Google passwords leaked online*. Global News. URL: <http://globalnews.ca/news/1009800/stolen-facebook-linkedin-google-passwords-leaked-online/> (visited on June 11, 2014).
- Bradner, S. (1997): *Key Words for Use in RFCs to Indicate Requirement Levels. RFC 2119*. IETF. URL: <https://tools.ietf.org/html/rfc2119> (visited on Apr. 6, 2015).
- Brandom, R. (2014): *TweetDeck vulnerability lets attackers execute code remotely*. URL: <http://www.theverge.com/2014/6/11/5800370/tweetdeck-vulnerability-lets-attackers-execute-code-remotely> (visited on June 14, 2014).

- Bray, T. (2014): *The JavaScript Object Notation (JSON) Data Interchange Format*. RFC 7159. IETF. URL: <http://tools.ietf.org/html/rfc7159> (visited on May 24, 2015).
- Bray, T.; Paoli, J.; Sperberg-McQueen, C. M.; Maler, E.; Yergeau, F. (2008): *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. W3C. URL: <http://www.w3.org/TR/2008/REC-xml-20081126/> (visited on May 24, 2015). W3C Recommendation 26 November 2008.
- Brickley, D.; Guha, R. V.; McBride, B. (2014): *RDF Schema 1.1*. W3C. URL: <http://www.w3.org/TR/2014/REC-rdf-schema-20140225/> (visited on May 23, 2015). W3C Recommendation 25 February 2014.
- Brickley, D.; Miller, L. (2014): *FOAF Vocabulary Specification 0.99. Namespace Document 14 January 2014 - Paddington Edition*. URL: <http://xmlns.com/foaf/spec/> (visited on May 30, 2014).
- Cameron, K. (2005): *The Laws of Identity*. URL: <http://www.identityblog.com/?p=352> (visited on June 14, 2014).
- Cantor, S. et al. (2005a): *Shibboleth Architecture. Protocols and Profiles*. Ed. by Cantor, S. et al. URL: <http://shibboleth.internet2.edu/shibboleth-documents.html> (visited on Mar. 18, 2016).
- Cantor, S.; Kemp, J.; Philpott, R.; Maler, E. (2005b): *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*. Ed. by Cantor, S.; Kemp, J.; Philpott, R.; Maler, E. URL: <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf> (visited on Mar. 18, 2016).
- Çelik, T.; Suda, B. (2013): *hCard 1.0*. URL: <http://microformats.org/wiki/hcard> (visited on May 24, 2015).
- Chayka, K. (2014): *What Are You Worth as a Facebook, Twitter, Tumblr, or Instagram User?* Pacific Standard. URL: <http://www.psmag.com/navigation/business-economics/worth-facebook-twitter-tumblr-instagram-user-74539/> (visited on Sept. 18, 2014).
- Chinnici, R.; Moreau, J.-J.; Ryman, A.; Weerawarana, S. (2007): *Web Services Description Language (WSDL) Version 2.0 Part 1. Core Language*. W3C. URL: <https://www.w3.org/TR/2007/REC-wsd120-20070626/> (visited on June 5, 2014). W3C Recommendation 26 June 2007.

- Chowdhury, S. R.; Daniel, F.; Tschudnowsky, A.; Wild, S.; Gaedke, M., et al. (2013): *Final Specification of Mashup Description Language and Telco Mashup Architecture. Deliverable 2.3*. URL: http://www.ict-omelette.eu/c/document_library/get_file?p_l_id=48742&folderId=165188&name=DLFE-12333.pdf (visited on July 1, 2016).
- Clark, J.; DeRose, S. (1999): *XML Path Language (XPath)*. W3C. URL: <http://www.w3.org/TR/1999/REC-xpath-19991116/> (visited on May 23, 2015). W3C Recommendation 28 October 2004.
- Connolly, D. (2007): *Gleaning Resource Descriptions from Dialects of Languages (GRDDL)*. W3C. URL: <http://www.w3.org/TR/2007/REC-grddl-20070911/> (visited on May 29, 2015). W3C Recommendation 11 September 2007.
- Cooper, D.; Santesson, S.; Farrell, S.; Boeyen, S., et al. (2008): *Internet X.509 Public Key Infrastructure Certificate And Certificate Revocation List (CRL) Profile. RFC 5280*. IETF. URL: <http://tools.ietf.org/html/rfc5280> (visited on June 3, 2014).
- Darwell, B. (2012): *Facebook policy now clearly bans exporting user data to competing social networks*. URL: <http://www.insidefacebook.com/2012/08/09/facebook-platform-policy-now-clearly-bans-exporting-user-data-to-competing-social-networks/> (visited on June 14, 2014).
- DataBreaches.net (2016): *DataBreaches.net*. URL: <https://www.databreaches.net/> (visited on July 1, 2016).
- DCMI (2012): *DCMI Metadata Terms*. Ed. by Board, D. U. Dublin Core Metadata Initiative (DCMI). URL: <http://dublincore.org/documents/2012/06/14/dcmi-terms/> (visited on May 31, 2015).
- Delo, C. (2014): *How Much Are You Really Worth to Facebook and Google? Advertising Age*. URL: <http://adage.com/article/digital/worth-facebook-google/293042/> (visited on Sept. 18, 2014).
- Dierks, T.; Rescorla, E. (2008): *The Transport Layer Security (TLS) Protocol. RFC 5246*. Version 1.2. IETF. URL: <http://tools.ietf.org/html/rfc5246> (visited on May 31, 2014).

- EC (2012): *Commission Proposes a Comprehensive Reform of the Data Protection Rules*. Including referred fact sheets. European Commission (EC). URL: http://ec.europa.eu/justice/newsroom/data-protection/news/120125_en.htm (visited on Oct. 17, 2014).
- Facebook (2015): *Accessing Your Facebook Data*. URL: <https://www.facebook.com/help/405183566203254> (visited on May 21, 2015).
- Fallside, D. C.; Walmsley, P. (2014): *XML Schema. Part 0: Primer Second Edition*. W3C. URL: <http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/> (visited on May 23, 2015). W3C Recommendation 28 October 2004.
- Finkle, J.; Saba, J. (2012): *LinkedIn suffers data breach - security experts*. Reuters. URL: <http://www.reuters.com/article/2012/06/06/net-us-linkedin-breach-idUSBRE85511820120606> (visited on June 11, 2014).
- Fitzpatrick, B.; Recordon, D.; Hardt, D.; Hoyt, J. (2007): *OpenID Authentication 2.0 - Final*. Ed. by Recordon, D.; Bufu, J.; Hoyt, J. OpenID Foundation. URL: http://openid.net/specs/openid-authentication-2_0.html (visited on June 2, 2014).
- Galiegue, F; Zyp, K.; Court, G. (2013): *JSON Schema. Core Definitions and Terminology*. URL: <http://json-schema.org/latest/json-schema-core.html> (visited on May 23, 2015).
- Gandon, F; Schreiber, G. (2014): *RDF 1.1 XML Syntax*. W3C. URL: <http://www.w3.org/TR/2014/REC-rdf-syntax-grammar-20140225/> (visited on May 31, 2015). W3C Recommendation 25 February 2014.
- Goessner, S. (2007): *JSONPath - XPath for JSON*. URL: <http://goessner.net/articles/JsonPath/> (visited on May 23, 2015).
- Gutwirth, A. (2015): *Die eID-Funktion (Online-Ausweisfunktion) des neuen Personalausweises (nPA)*. German. URL: <http://www.die-eid-funktion.de/> (visited on May 21, 2015).
- Hadley, M. (2009): *Web Application Description Language*. W3C. URL: <https://www.w3.org/Submission/2009/SUBM-wadl-20090831/> (visited on June 5, 2014). W3C Member Submission 31 August 2009.

- Hardt, D. (2012): *The OAuth 2.0 Authorization Framework. RFC 6749*. IETF. URL: <http://tools.ietf.org/html/rfc6749> (visited on June 2, 2014).
- Hardt, D.; Bufu, J.; Hoyt, J. (2007): *OpenID Attribute Exchange 1.0 - Final*. OpenID Foundation. URL: http://openid.net/specs/openid-attribute-exchange-1_0.html (visited on June 2, 2014).
- Harris, S.; Seaborne, A. (2013): *SPARQL 1.1 Query Language*. W3C. URL: <https://www.w3.org/TR/2013/REC-sparql11-query-20130321/> (visited on May 31, 2014). W3C Recommendation 21 March 2013.
- Herman, I.; Adida, B.; Sporny, M.; Birbeck, M. (2015): *RDFa 1.1 Primer - Third Edition. Rich Structured Data Markup for Web Documents*. W3C. URL: <http://www.w3.org/TR/2015/NOTE-rdfa-primer-20150317/> (visited on May 23, 2015). W3C Working Group Note 17 March 2015.
- Hickson, I. (2013): *HTML Microdata*. W3C. URL: <http://www.w3.org/TR/2013/NOTE-microdata-20131029/> (visited on May 29, 2015). W3C Working Group Note 29 October 2013.
- Hodges, J.; Philpott, R.; Maler, E. (2005): *Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0*. Ed. by Hodges, J.; Philpott, R.; Maler, E. URL: <http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf> (visited on Mar. 18, 2016).
- Holm, M. (2015): *JSON: The JavaScript Subset That Isn't*. URL: <http://timelessrepo.com/json-isnt-a-javascript-subset> (visited on May 24, 2015).
- Horacek, L. (2013): *IBM X-Force 2013 Mid-Year Trend and Risk Report*. URL: <http://securityintelligence.com/cyber-attacks-research-reveals-top-tactics-xforce/> (visited on Sept. 22, 2014).
- Hughes, J.; Cantor, S.; Hodges, J.; Hirsch, F., et al. (2005): *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0*. Ed. by Hughes, J.; Cantor, S.; Hodges, J.; Hirsch, F., et al. URL: <http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf> (visited on Mar. 18, 2016).
- Hunt, T. (2014): *Adobe credentials and the serious insecurity of password hints*. URL: <http://www.troyhunt.com/2013/11/adobe-credentials-and-serious.html> (visited on June 11, 2014).

- Hunt, T. (2016): *Have I been pwned?* URL: <https://haveibeenpwned.com/> (visited on July 1, 2016).
- Iannella, R.; McKinney, J. (2014): *vCard Ontology - for describing People and Organizations*. W3C Interest Group Note 22 May 2014. W3C. URL: <http://www.w3.org/TR/2014/NOTE-vcard-rdf-20140522/> (visited on May 24, 2015).
- Ilyin, Y. (2014): *A confirmed eBay leak: another password alert*. Kaspersky Lab. URL: <http://business.kaspersky.com/a-confirmed-ebay-leak-another-password-alert/> (visited on June 11, 2014).
- Inkster, T.; Story, H.; Harbulot, B. (2014): *WebID-TLS. WebID Authentication over TLS*. Ed. by Story, H.; Corlosquet, S.; Sambra, A. W3C. URL: <http://www.w3.org/2005/Incubator/webid/spec/tls/> (visited on May 31, 2014). W3C Editor's Draft 05 March 2014.
- Josefsson, S. (2006): *The Base16, Base32, and Base64 Data Encodings*. RFC 4648. IETF. URL: <http://tools.ietf.org/html/rfc4648> (visited on June 3, 2014).
- Kopecký, J. (2007): *Web Services Description Language (WSDL) Version 2.0. RDF Mapping*. W3C. URL: <https://www.w3.org/TR/2007/NOTE-wsd120-rdf-20070626/> (visited on June 5, 2014). W3C Working Group Note 26 June 2007.
- LeakedIn (2016): *LeakedIn - Stories about Data Leaks and Related Stuff*. URL: <http://www.leakedin.com/> (visited on July 1, 2016).
- Longley, D. et al. (2014): *Forge. A native implementation of TLS (and various other cryptographic tools) in JavaScript*. URL: <https://github.com/digitalbazaar/forge> (visited on June 4, 2014).
- Lystad, T. A. (2013): *2013-12-22_an_update_on_oclhashcat*. URL: http://thepasswordproject.com/2013-12-22_an_update_on_oclhashcat (visited on June 14, 2014).
- McCandless, D.; Evans, T. (2013): *World's Biggest Data Breaches*. URL: <http://www.informationisbeautiful.net/2013/worlds-biggest-data-breaches/> (visited on Oct. 11, 2014).
- Merriam-Webster (2015): *Government*. URL: <http://www.merriam-webster.com/dictionary/government> (visited on July 1, 2016).

- Merriam-Webster (2016a): *Assumption*. URL: <http://www.merriam-webster.com/dictionary/assumption> (visited on July 1, 2016).
- Merriam-Webster (2016b): *Challenge*. URL: <http://www.merriam-webster.com/dictionary/challenge> (visited on July 1, 2016).
- Merriam-Webster (2016c): *Hypothesis*. URL: <http://www.merriam-webster.com/dictionary/hypothesis> (visited on July 1, 2016).
- Messina, C. (2009): *Does OpenID need to be hard?* URL: <http://factoryjoe.com/blog/2009/04/06/does-openid-need-to-be-hard/> (visited on June 14, 2014).
- Mozilla (2013): *BrowserID*. URL: <https://github.com/mozilla/id-specs/blob/prod/browserid/index.md> (visited on May 4, 2015).
- Openinfocard (2016): *Openinfocard*. URL: <https://code.google.com/archive/p/openinfocard/> (visited on July 2, 2016).
- OpenSSL Software Foundation (2016): *OpenSSL - Cryptography and SSL/TLS Toolkit*. URL: <https://www.openssl.org/> (visited on July 2, 2016).
- Oracle (2016): *keytool - Key and Certificate Management Tool*. URL: <http://docs.oracle.com/javase/6/docs/technotes/tools/windows/keytool.html> (visited on July 2, 2016).
- Perreault, S. (2011a): *vCard Format Specification. RFC 6350*. IETF. URL: <http://tools.ietf.org/html/rfc6350> (visited on May 23, 2015).
- Perreault, S. (2011b): *xCard: vCard XML Representation. RFC 6351*. IETF. URL: <http://tools.ietf.org/html/rfc6351> (visited on May 23, 2015).
- Reisinger, D. (2014): *eBay hacked, requests all users change passwords*. CNET. URL: <http://www.cnet.com/news/ebay-hacked-requests-all-users-change-passwords/> (visited on June 11, 2014).
- Riddell, K. (2011): *Security-Breach Costs Climb 7% to \$7.2 Million per Incident*. Bloomberg. URL: <http://www.bloomberg.com/news/2011-03-08/security-breach-costs-climb-7-to-7-2-million-per-incident.html> (visited on Oct. 14, 2014).

- Rissanen, E. (2013): *eXtensible Access Control Markup Language (XACML) Version 3.0*. Ed. by Rissanen, E. URL: <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-en.html> (visited on Aug. 14, 2015).
- Sakimura, N. (2013): *Introduction to OpenID Connect*. URL: http://www.slideshare.net/nat_sakimura/introduction-to-openid-connect (visited on July 6, 2014).
- Sakimura, N.; Bradley, J.; Jones, M. B.; Medeiros, B. de; Mortimore, C. (2014): *OpenID Connect Core 1.0*. The OpenID Foundation. URL: http://openid.net/specs/openid-connect-core-1_0.html (visited on May 10, 2015).
- Sambra, A.; Story, H.; Berners-Lee, T. (2014): *WebID 1.0. Web Identity and Discovery*. Ed. by Sambra, A.; Corlosquet, S. W3C. URL: <http://www.w3.org/2005/Incubator/webid/spec/identity/> (visited on May 17, 2015). W3C Editor's Draft 05 March 2014.
- Schreiber, G.; Raimond, Y. (2014): *RDF 1.1 Primer*. W3C. URL: <https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/> (visited on July 3, 2014). W3C Working Group Note 24 June 2014.
- Shalal-Esa, A. (2012): *Scores of U.S. firms keep quiet about cyber attacks*. Reuters. URL: <http://www.reuters.com/article/2012/06/13/net-us-media-tech-summit-cyber-disclosur-idUSBRE85C1E320120613> (visited on June 14, 2014).
- Sporny, M.; Longley, D.; Kellogg, G.; Lanthaler, M.; Lindström, N. (2014): *JSON-LD 1.0. A JSON-based Serialization for Linked Data*. Ed. by Sporny, M.; Kellogg, G.; Lanthaler, M. W3C. URL: <http://www.w3.org/TR/2014/REC-json-ld-20140116/> (visited on May 24, 2015). W3C Recommendation 16 January 2014.
- Steube, J. (2016): *Hashcat*. URL: <https://hashcat.net/hashcat/> (visited on July 1, 2016).
- Story, H. (2008): *The Cert Ontology 1.0*. W3C. URL: <http://www.w3.org/ns/auth/cert#> (visited on May 31, 2015). Namespace Document 13 November 2008.
- SurveyMonkey Inc. (2015): *The keygen Element*. URL: <http://www.wufoo.com/html5/elements/4-keygen.html> (visited on Aug. 14, 2015).

- Tauberer, J. (2014): *What is RDF and what is it good for?* URL: <https://github.com/JoshData/rdfabout/blob/gh-pages/intro-to-rdf.md> (visited on May 24, 2015).
- Trevithick, P (2016): *Higgins*. URL: <http://www.eclipse.org/higgins/> (visited on July 2, 2016).
- Tschudnowsky, A.; Wild, S.; Gaedke, M., et al. (2013): *Final Dissemination and Standardization Report. Deliverable 8.5*. URL: http://www.ict-omelette.eu/c/document_library/get_file?p_l_id=48742&folderId=157989&name=DLFE-12320.pdf (visited on July 1, 2016).
- Turner, S. (2010): *The application/pkcs10 Media Type. RFC 5967*. IETF. URL: <http://tools.ietf.org/html/rfc5967> (visited on June 5, 2014).
- Verizon (2014): *2014 Data Breach Investigations Report*. URL: <http://www.verizonenterprise.com/DBIR/2014/> (visited on Sept. 22, 2014).
- W3C (2004): *Web Services Glossary. W3C Working Group Note 11 February 2004*. Ed. by Haas, H.; Brown, A. World Wide Web Consortium (W3C). URL: <https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/> (visited on Feb. 14, 2015).
- W3C (2015a): *Linked Data*. World Wide Web Consortium (W3C). URL: <http://www.w3.org/standards/semanticweb/data> (visited on May 29, 2015).
- W3C (2015b): *Vocabularies*. World Wide Web Consortium (W3C). URL: <http://www.w3.org/standards/semanticweb/ontology> (visited on May 29, 2015).

List of Acronyms

ACL	Access Control List.
ACO	Access Control Ontology.
AJAX	Asynchronous JavaScript and XML.
API	Application Programming Interface.
BPMN	Business Process Model and Notation.
CA	Certificate Authority.
CAC	Context-Aware Control.
CBWE	Component-Based Web Engineering.
CIM	Computational Independent Model.
CPU	Central Processing Unit.
CV	Curriculum Vitae.
DC	Dublin Core.
DSL	Domain-Specific Language.
EC	European Commission.
ER	Entity-Relationship.
EU	European Union.

FGF	Fine-Grained Filtering.
FOAF	Friend of a Friend.
FP7	Seventh Framework Programme for Research and Technological Development.
GPL	General-Purpose Language.
GPU	Graphics Processing Unit.
GRDDL	Gleaning Resource Descriptions from Dialects of Languages.
GUI	Graphical User Interface.
HDM	Hypertext-Design Model.
HTML	HyperText Markup Language.
HTTP	HyperText Transfer Protocol.
IAN	Issuer Alternative Name.
IdDL	Identity Description Language.
IdFS	Identity Federation System.
IdM	Identity Management.
IdMS	Identity Management System.
IdP	Identity (Service) Provider.
IEC	International Electrotechnical Commission.
ISO	International Organization for Standardization.
IT	Information Technology.
JSON	JavaScript Object Notation.
JSON-LD	JSON for Linked Data.
LFA	Logical Framework Approach.
MD5	Message-Digest Algorithm 5.
MDA	Model-Driven Architecture.
MDD	Model-Driven Development.
MIME	Multipurpose Internet Mail Extensions.

MSG	Minimum Self-contained Graph.
N3	Notation3.
OECD	Organisation for Economic Co-operation and Development.
OMELETTE	Open Mashup Enterprise service platform for LinkEd data in the Telco domain.
OOHDM	Object-Oriented Hypermedia Design Method.
OSN	Online Social Network.
OWL	Web Ontology Language.
PAD	Personal Authentication Device.
PCM	Project Cycle Management.
PII	Personally Identifiable Information.
PIM	Platform Independent Model.
PKI	Public Key Infrastructure.
PMBOK®	Project Management Body of Knowledge.
PMI	Project Management Institute.
PR	Public Relations.
PRINCE2®	PRojects IN Controlled Environments.
PSM	Platform Specific Model.
QR	Quick Response.
RDF	Resource Description Framework.
RDFa	RDF in attributes.
RDFS	RDF schema.
REST	REpresentational State Transfer.
RFC	Request for Comments.
RP	Relying Party.
SAML	Security Assertion Markup Language.
SAN	Subject Alternative Name.

SHA-1	Secure Hash Algorithm (160 bits).
SHA-256	Secure Hash Algorithm (256 bits).
SNS	Social Networking Service.
SNSP	Social Networking Service Provider.
SOA	Service-Oriented Architecture.
SOAP	Simple Object Access Protocol.
SP	Service Provider.
SPARQL	SPARQL Protocol and RDF Query Language.
SR	Security Realm.
SSO	Single Sign-On.
TE	Tamper-Evidentness.
TLS	Transport Layer Security.
UAO	User Access Ontology.
UGC	User-Generated Content.
UML	Unified Modeling Language.
URI	Uniform Resource Identifier.
URL	Uniform Resource Locator.
UWE	UML-based Web Engineering.
VSR	(Professur) Verteilte und Selbstorganisierende Rechnersysteme.
W3C	World Wide Web Consortium.
WAC	Web Access Control.
WADL	Web Application Description Language.
WAM	WebComposition Architecture Model.
WCCM	WebComposition Component Model.
WCML	WebComposition Markup Language.
WCPM	WebComposition Process Model.
WDL	WebID Delegation Language.

WEA	Web Engineering Approach.
WebML	Web Modeling Language.
WebSA	Web Software Architecture.
WEF	World Economic Forum.
WoT	Web of Trust.
WPFL	WebID Profile Filter Language.
WSDL	Web Service Description Language.
WSDM	Web-Site Design Method.
WWW	World Wide Web.
XACML	eXtensible Access Control Markup Language.
XML	eXtensible Markup Language.
XSD	XML Schema Definition.
XSS	cross-Site Scripting.

List of Figures

1.1	Organization of Dissertation	19
2.1	Methodology to Approach the Problem Using PCM and LFA .	27
2.2	Essential Terms and Relationships	29
2.3	Structure of Causes and Effects of the Central Problem	40
2.4	Problem Cause 1 and Associated Root Cause	41
2.5	Problem Cause 2 and Associated Root Causes	42
2.6	Problem Cause 3 and Associated Root Causes	45
2.7	Problem Effect 1 and Associated Aftereffects	50
2.8	Problem Effects 2 to 4 and Associated Aftereffect	53
2.9	Transformation of Problems into Objectives	54
2.10	Activity 2.1 and Directly Subordinated Activities	57
2.11	Activities 3.1 to 3.3 and Directly Subordinated Activities . . .	59
2.12	Strategy for Processing the Objectives	62
3.1	Remote Silo IdM Model	94
3.2	Common Domain IdM Model	96
3.3	Centralized SSO IdM Model	98
3.4	Federated SSO IdM Model	100
3.5	Omitted Silo IdM Model	103
3.6	Local Silo IdM Model	106

3.7	Open Silo IdM Model	108
4.1	WAM Ontology	123
4.2	Default WebID Authentication Sequence	127
4.3	Artifacts in WebID: Certificate, URI and Profile	130
4.4	Architecture for Self-Deterministic Identity Management	135
4.5	IdM Life Cycle as per Proposal	136
4.6	IdM Life Cycle for Compositions	139
4.7	Security-Enhanced WebID Authentication Sequence for SPs	143
4.8	Security-Enhanced WebID Authentication Sequence for IdPs	144
4.9	Tool Support for WAM-based Architecture Modeling	150
4.10	Representations of a WebID Profile Managed Using Sociddea	151
5.1	Case Differentiation for Certificate Generation	166
5.2	Delegation Process Model	167
5.3	Sequence of Verification for Scope-Compliant Delegation	168
5.4	WebID Artifacts For Context-Aware Control in Delegations	172
5.5	WebID Certificate Creation with Sociddea	174
5.6	Delegation Creation with Sociddea	175
6.1	Tamper-Evidentness Process Model	190
6.2	Sequence of Detecting Tampering and Identity Theft	194
6.3	Tool Support for Creating Tamper-Evident WebID Profiles	197
6.4	Results of Tamper-Evidentness Component in Sociddea	198
7.1	Fine-Grained Filtering Process Model	218
7.2	Creation of Filter Specification Based on User Selection	222
7.3	WebID Profile Filtered for Anonymous Requesting Entities	224
A.1	Cause-Effect Relationships of Analyzed Problems	276
A.2	Means-Ends Relationships of Unconsolidated Objectives	277
A.3	Means-Ends Relationships Among Consolidated Objectives	278

List of Listings

4.1	WebID Profile of a Web Service	131
5.1	Template of Delegation Specification as per WDL	173
7.1	Template of Filter Specification as per WPFL	219
7.2	Generic Filter Command Specification	220
7.3	Exemplary SPARQL CONSTRUCT Query with Property Paths	221

List of Symbols

A	Set of all attributes.
A_c	Subset of A that is specific to a context c .
A_i	Subset of A_c that allows for clearly identifying e .
a	Attribute out of A .
B	Set of all delegatable tasks.
b	Task out of B to be done by delegate denoted by identity i_2 .
C	Set of all contexts.
c	Context out of C .
D	Set of delegations.
D_1	Subset of D specified by identity i_1 .
d	Delegation out of D_1 that involves task b to be carried out by delegate i_2 in consideration of constraints Q .
E	Set of all entities.
e	Entity out of E .
\mathcal{G}	Set of all graphs.

\mathfrak{G}_i	Subset of \mathfrak{G} that represents all filtered variants of graph G .
G	Graph out of \mathfrak{G} that describes attributes of an identity i .
\tilde{G}	Canonical representation of graph G .
\tilde{G}'	Canonical representation of filtered graph G' .
\tilde{G}_*	Minimum self-contained graph (MSG) as per (Carroll, 2003; Tumarello et al., 2005) being a subset of graph \tilde{G} .
G'	Graph out of \mathfrak{G} being a filtered version of G , i.e., a subset of G .
\mathfrak{I}	Set of all identities.
I	Subset of \mathfrak{I} that is associated with an entity e .
I_o	Subset of \mathfrak{I} having no elements shared with subset I_z (cf. anonymity).
I_p	Subset of I_z containing all identities identity owner i is connected with.
I_s	Subset of I_z that contains specific requesters.
I_z	Subset of \mathfrak{I} that contains all authenticated requesters.
i	Identity out of \mathfrak{I} .
i_1	Delegator's identity out of \mathfrak{I} .
i_2	Delegate's identity out of \mathfrak{I} .
i_f	Fallback identity out of \mathfrak{I} .
i_{null}	Theoretical <i>null</i> identity out of \mathfrak{I} .
i_o	Generic identity out of I_o that represents unauthenticated entities, which are therefore <i>not</i> known to identity owner i .
i_p	Generic identity out of I_p that represents friends of identity owner i .
i_r	Requester's identity out of \mathfrak{I} .
i_z	Generic identity out of I_z that represents entities authenticatable to identity owner i .
\mathfrak{K}	Set of all asymmetric keys.
K	Subset of \mathfrak{K} that is owned by an identity i .
K_2	Subset of \mathfrak{K} that is owned by identity i_2 .
k	Key out of \mathfrak{K} .

k'^{-1}	Private key out of K .
$k_2'^{-1}$	Private key out of K_2 .
$k_*'^{-1}$	Main private key out of K chosen by an identity i .
k'	Public key out of K .
k_2'	Public key out of K_2 .
k_*'	Main public key out of K chosen by an identity i .
L	Set of all links, which are also referred to as edges.
L'	Subset of L .
M	Set of all messages.
m	Message out of M .
m_{dig}	Hash value (digest) out of M .
m_{inq}	Message out of M symbolizing an inquiry from server to client.
m_{res}	Message out of M symbolizing a response from client to server.
m_{sig}	Signature out of M .
Q	Set of constraints.
\mathfrak{T}	Set of all RDF triples.
T	Subset of \mathfrak{T} that describes attributes of an identity i .
T_1	Subset of \mathfrak{T} that describes attributes of identity i_1 .
T_{1,D_1}	Subset of T_1 that describes delegations D_1 .
$T_{1,d}$	Subset of T_{1,D_1} that describes delegation d .
T_2	Subset of \mathfrak{T} that describes attributes of identity i_2 .
\tilde{T}	Canonical representation of triple set T .
\tilde{T}'	Canonical representation of triple set T' .
T'	Subset of \mathfrak{T} that is a filtered version of RDF triple set T , i.e., a subset of T .
T_K	Subset of T that describes public keys K .
$T_{k'}$	Subset of T_K that describes a single public key k' .
t	RDF triple out of \mathfrak{T} .

t_1	Subject in RDF triple t .
t_2	Predicate in RDF triple t .
t_3	Object in RDF triple t .
U	Set of all URIs.
u	URI out of U .
V	Set of all vertices.
V'	Subset of V .
W	Set of all WebID URIs.
w	WebID URI out of W that refers to T .
w'	WebID URI out of W that refers to T and involves protection means.
w_1	Delegator's WebID URI out of W .
w_2	Delegate's WebID URI out of W .
\mathfrak{X}	Set of all WebID certificates.
X	Subset of \mathfrak{X} that is associated with an identity i .
X_2	Subset of \mathfrak{X} that is associated with identity i_2 .
$X_{k'}$	WebID certificate out of X for which i owns key pair (k', k'^{-1}) .
X_{2, k'_2}	WebID certificate out of X_2 for which i_2 owns key pair (k'_2, k'^{-1}_2) .
α	Function $\alpha(u)$ yielding T for URI u being a valid WebID URI w .
β	Function $\beta(k, m)$ yielding $m' \in M$ for inputs key k and message m .
γ	Function $\gamma(G, i_r)$ yielding G' for inputs graph G and identity i_r .
δ	Function $\delta(i_r, t)$ yielding either 0 or 1 for inputs identity i_r and triple t .
ϵ	Function $\epsilon(i_r)$ yielding i_f for input requester identity i_r .
ζ	Function $\zeta(w')$ yielding k'_* for input WebID URI w' .
η	Function $\eta(G)$ yielding \tilde{G} for input graph G .
θ	Function $\theta(G)$ yielding m_{dig} for inputs like triple set \tilde{T} or triple t .

List of Tables

3.1	Analysis Results of Data-Oriented Web Eng. Approaches . . .	82
3.2	Analysis Results of Hypertext-Oriented Web Eng. Approaches	84
3.3	Analysis Results of Object-Oriented Web Eng. Approaches . .	85
3.4	Analysis Results of Software-Oriented Web Eng. Approaches	87
3.5	Analysis Results of Component-Based Web Eng. Approaches .	90
3.6	Analysis Results of Web Engineering Approaches and Corre- sponding Tools in Terms of Collective Requirements	92
3.7	Analysis Results of Remote Silo Model	95
3.8	Analysis Results of Common Domain Model	97
3.9	Analysis Results of Centralized-SSO Model	98
3.10	Analysis Results of Multi-Domain SSO Model	100
3.11	Analysis Results of Federated SSO Model	103
3.12	Analysis Results of Omitted Silo Model	105
3.13	Analysis Results of Local Silo Model	107
3.14	Analysis Results of Open Silo Model	110
3.15	Analysis Results of Identity Management Systems in Terms of Collective Requirements	111
3.16	Analysis Results of Proprietary Languages	113
3.17	Analysis Results of Data Interchange Languages	114
3.18	Analysis Results of Markup Languages	115
3.19	Analysis Results of Semantic Languages	117

3.20 Analysis Results of Identity Description Languages in Terms
of Collective Requirements 118

A.1 Stakeholder Matrix 271

A.2 Logframe Matrix 279

Doctoral Dissertations in Web Engineering and Web Science

(1) Heinrich, Matthias (2014)

Enriching Web Applications Efficiently with Real-Time Collaboration Capabilities

ISBN 978-3-941003-25-9

Volltext: <http://nbn-resolving.de/urn:nbn:de:bsz:ch1-qucosa-149948>

(2) Speicher, Maximilian (2016)

Search Interaction Optimization: A Human-Centered Design Approach

ISBN 978-3-944640-99-0

Volltext: <http://nbn-resolving.de/urn:nbn:de:bsz:ch1-qucosa-208102>

(3) Wild, Stefan (2017)

Enhancing Security in Managing Personal Data by Web Systems

ISBN 978-3-96100-010-4

Volltext: <http://nbn-resolving.de/urn:nbn:de:bsz:ch1-qucosa-217284>