



**Automated VALidationN of Trust and Security
of Service-oriented ARchitectures**

FP7-ICT-2007-1, Project No. 216471

www.avantssar.eu

Deliverable D5.1

Problem Cases and their Trust and Security Requirements

Abstract

This deliverable defines a collection of SOA trust and security problem cases that we have extracted from our application scenarios from the e-business, e-government, and e-health application areas. To illustrate the problem cases, we describe all scenarios in detail, focusing on their trust and security requirements.

Deliverable details

Deliverable version: *v1.0*
Date of delivery: *31.07.2008*
Editors: *all*

Classification: *public*
Due on: *31.07.2008*
Total pages: *96*

Project details

Start date: *January 01, 2008*
Project Coordinator: *Luca Viganò*
Partners: UNIVR, ETH Zurich, INRIA, UPS-IRIT, UGDIST, IBM,
OpenTrust, IEAT, SAP, SIEMENS

Duration: *36 months*



Contents

1	Introduction	5
1.1	Scope and objectives	5
1.2	Deliverable structure	5
2	Application areas	8
2.1	E-Business	8
2.2	E-Government	8
2.3	E-Health	9
3	Application scenarios	10
3.1	Banking services	10
3.1.1	Scenario definition	10
3.1.2	Security and trust requirements	15
3.1.3	Authorization Policies	20
3.1.4	Accountability	20
3.1.5	Trust Management	20
3.1.6	Workflow Security	23
3.1.7	Privacy	25
3.1.8	Application Data Protection	27
3.1.9	Communication Security	27
3.2	Software Distribution Services	29
3.2.1	Scenario definition	29
3.2.2	Security and trust requirements	30
3.2.3	Authorization Policies	34
3.2.4	Accountability	34
3.2.5	Trust Management	34
3.2.6	Application Data Protection	34
3.3	Anonymous Shopping	35
3.3.1	Scenario definition	35
3.3.2	Security and trust requirements	38
3.3.3	Federation	40
3.3.4	Accountability	40
3.3.5	Privacy	40
3.4	Citizen and Service portals	41
3.4.1	Scenario definition	43
3.4.2	Security and trust requirements	50
3.4.3	Federation	54
3.4.4	Authorization Policies	55
3.4.5	Accountability	55

3.4.6	Trust Management	55
3.4.7	Workflow Security	56
3.4.8	Privacy	56
3.4.9	Application Data Protection	56
3.4.10	Communication Security	57
3.5	Document Exchange Procedures	58
3.5.1	Scenario definition	58
3.5.2	Security and trust requirements	73
3.5.3	Authorization Policies	78
3.5.4	Accountability	78
3.5.5	Trust Management	79
3.5.6	Workflow Security	79
3.5.7	Application Data Protection	79
3.5.8	Communication Security	79
3.6	Personal Health Information in the Hospital	80
3.6.1	Scenario definition	83
3.6.2	Security and trust requirements	91
3.6.3	Federation	92
3.6.4	Authorization Policies	92
3.6.5	Accountability	93
3.6.6	Privacy	93
3.6.7	Application Data Protection	93
3.6.8	Communication Security	93

List of Figures

1	The loan origination process and its sub-processes	13
2	SDS overall architecture	29
3	SSV application	33
4	Example Application Process of a French Optician in Germany . .	46
5	The Car Registration Process	48
6	Peter Storing a Document in CentrRep	50
7	Architecture of a digital contract signing system featuring Open-trust SPI platform	59
8	Contract signing sequence diagram, part 1: HHH signature	61
9	Contract signing sequence diagram, part 2: WWW signature	62
10	Basic Public Bidding BP - Publication sub-process	66
11	Basic Public Bidding BP - Submission sub-process	67
12	Basic Public Bidding BP - Evaluation sub-process	68
13	Basic Public Bidding BP - Decision sub-process	69
14	Extended Public Bidding BP - Publication sub-process	71
15	Extended Public Bidding BP - Submission sub-process	72
16	Extended Public Bidding BP - Evaluation sub-process	74
17	Logical Access Control Structure for the EHR.	83
18	Basic Data Flow in the EHR Scenario	86

1 Introduction

1.1 Scope and objectives

WP 5 plays a central role within AVANTSSAR since it aims at providing a proof of concept for the entire project outcome. More precisely, its goal is to arrive at a library of SOA problem cases that are validated using the AVANTSSAR Validation Platform provided by WP 4, which in turn adopts the validation techniques developed in WP 3. To this end, the problem cases in the library will be described formally using the formal specification language ASLan introduced in WP 2.

A first step towards the formalization of the library is a detailed description of the relevant problem cases. The second step is more challenging, and consists in the identification and exact presentation for each problem case of the issues that are in the scope of AVANTSSAR, namely trust and security. The project partners have taken and completed these steps, and the present deliverable D5.1 reports on the relevant findings.

The work reported on in this deliverable has allowed us to make considerable progress in identifying and understanding the challenges to be tackled in the course of the AVANTSSAR project. In particular, this work indicates that the trust and security requirements for a SOA are heterogeneous. For example, authorization policies can be tailored towards privacy enforcement by regulating access to private data of individuals; trust management systems can be provided as web services for individuals to calculate their trust on other parties; and inside financial institutions, separation of duty becomes necessary to effectively prevent fraud, because a single actor cannot be trusted to bear no conflict of interest.

1.2 Deliverable structure

The structure of this deliverable reflects *Table 1: Application areas and some related problem cases* of the AVANTSSAR “Annex I — Description of Work”. We give here an updated version of the table since, as already foreseen in the original definition of WP 5.1, we have carried out a number of adaptations and changes to the list of application scenarios and families of problem cases:

- Due to changes in the relevance for the industrial partners, we have replaced two of the scenarios in the area of e-health, namely “Telematics Infrastructure” and “Patient Monitoring”, with a new one, namely the more comprehensive “Personal Health Information”. Accordingly, the Sensor Networks family of problem cases has been removed.
- We have introduced the following new families of problem cases: accountability, application data protection, and communication security.

Areas	Scenarios	Families of Problem Cases							
		Federation	Authorization Policies	Accountability	Trust Management	Workflow Security	Privacy	Application Data Protection	Communication Security
E-Business	in general	×	×	×	×	×	×	×	×
	Banking Services		×	×	×	×	×	×	×
	SW Distribution Services		×	×	×			×	
	Anonymous Shopping	×		×			×		
E-Government	in general	×	×	×	×	×	×	×	×
	Citizen and Service Portals	×	×	×	×	×	×	×	×
	Document Exchange Procedures		×	×	×	×		×	×
E-Health	in general	×	×	×	×		×	×	×
	Personal Health Information	×	×	×			×	×	×

Table 1: Application areas and the related families of problem cases

- The problem cases related to Public Key Infrastructure have been subsumed among the other families because it is more appropriate to see PKI as a mechanism (i.e., a solution) and not actually as a family of problem cases.
- The Digital Contract Signing and Public Bidding families of problem cases have been re-phrased as scenes within the Document Exchange Procedures application scenario. This is more coherent with other scenarios as both the Digital Contract Signing and Public Bidding scenes, as well as the scenes of the Citizen and Service Portals scenario, are concrete business processes that raise problem cases belonging to the remaining families of problem cases.
- The Single Sign-On (SSO) family of problem cases has been generalized to Federation.
- A detailed analysis has revealed that our SW Distribution scenario does not refer to SSO or other federation mechanisms.
- The Citizen and Service Portals scenario has additional problem cases of

the Workflow Security family.

- The more general Privacy family of problem cases subsumes the former Identity Mixer family of problem cases.

In the overall duration of the project, further modifications might be required, in particular due to potential shifts of interest at the industrial partners.

We proceed as follows. We begin by outlining the main application areas in [section 2](#). The various scenarios that are considered within those areas are then described in [section 3](#), grouped by the application area each scenario pertains to. We adopt a uniform structure to present each scenario so to facilitate a natural compare-and-contrast reading style. More specifically, each scenario has its own subsection where the scenario is defined first, optionally divided into several scenes. The relevant trust and security requirements are described next, followed by the description of the contributions to the families of problem cases produced by the requirements. These families of problem cases are described in [subsubsection 3.1.3– subsubsection 3.1.9](#), except for federation, which is described in [subsubsection 3.4.3](#).

2 Application areas

As already indicated in the project proposal and in the Description of Work, Information and Communication Technology (ICT) infrastructures in the areas of e-business, e-government, and e-health are of significant impact for the core business of the AVANTSSAR industrial partners. More generally, these application areas are critical elements of Europe's industrial portfolio. The pivot point for this deliverable turned out to be the following one: to select from those areas a proper assortment of diversified application scenarios with heterogeneous security and trust requirements.

2.1 E-Business

In the emerging global economy, e-business has increasingly become a necessary component of business strategy and a strong catalyst for economic development. The integration of ICT in business has revolutionized relationships within organizations and those between and among organizations and individuals. Specifically, the use of ICT in business has enhanced productivity, encouraged greater customer participation, and enabled mass customization, besides reducing costs.

E-business applications include commercial and administrative processes, but also automation, logistics and others, empowered by information systems. They allow enterprises to link their internal and external processes more efficiently and flexibly, to work more closely with suppliers and partners, and to better satisfy the needs and expectations of their customers. Integration is at the core of e-business: application-to-application (A2A) integration within and between enterprises, integration with database engines, automated interaction with customers and suppliers (e.g., across firewalls) resulting in B2B integration, etc.

It is intuitive that the complexity of this application area brings along a large variety of requirements that in turn will determine various families of problem cases. We have identified a large number of families of problem cases related to e-business: federation, authorization policies, accountability, trust management, workflow security, privacy, application data protection, and communication security.

2.2 E-Government

Today's Internet does not offer simple and cost-effective authentication procedures and security services for e-government purposes, despite very few exceptions in restricted environments or applications. By contrast, electronic transactions over the Web need to have the legal binding, certainty, and liability that is now common for surface mail or, more generally, for other types of paperwork.

Acceptance or trust in the security of remote communications for what concerns the administration of citizens is crucial for the social modernization effect that the Internet can potentially bring forward.

The following families of problem cases belong to this application area: federation, authorization policies, accountability, trust management, workflow security, privacy, application data protection, and communication security.

2.3 E-Health

The e-health area is complex. A typical scenario in this area involves a multitude of actors (patients, doctors, social workers, external service providers, pharmacies, health insurances, etc), of devices (health-care terminals, personal computers, PDAs, phones, VPNs, central servers, etc.) and of databases containing patient data.

There typically also are softwares that cooperate remotely to enhance and help the daily lives of patients as well as the quality, effectiveness and cost-efficiency of health care systems and institutions. One basic and general requirement in this area is that patient health data, their electronic prescriptions, their treatment bills, and other health-related information all need to be protected against disclosure and manipulation.

A number of families of problem cases are relevant to this area: federation, authorization policies, accountability, trust management, privacy, application data protection, and communication security.

3 Application scenarios

This section lists all scenarios studied within AVANTSSAR, along with their security and trust requirements and their corresponding contributions to the families of problem cases for each scenario.

3.1 Banking services

Banking services offer a paradigmatic scenario in the e-business application area as they highlight how the coexistence of flexibility on one hand and security and trust on the other often becomes contradictory. If banking applications are expected to be flexible to fulfill the needs of all involved parties, such as bankers, partners, customers and investors, they are also required to strictly meet a variety of security and trust requirements. Such a variety is in fact very large. For example, particular requirements concerning separation of duties, secure logging of events aimed at auditing, non-repudiable actions, digital signatures, etc, need to be considered and satisfied in order to comply with the current regulations.

One of the core processes of a banking scenario is the loan origination process, which formalizes a bank's evaluation of a customer's request for a loan. Its description is partly borrowed from the European FP6 Project "Serenity" (System Engineering for Security and Dependability, [20]), which aims at enhancing security and dependability for Ambient Intelligence ecosystems by capturing security expertise and making it available for automated processing. As AVANTSSAR concerns the validation of security and trust aspects, the present deliverable required a much deeper investigation about the loan origination process in order to define its security and trust requirements and the related problem cases. Below, the scenario is first described in its participants and their interaction ([subsection 3.1.1](#)). Then, its security and trust requirements are stated ([subsection 3.1.2](#)), and after that several families of problem cases are introduced (starting with [subsection 3.1.3](#)).

3.1.1 Scenario definition

We focus here on a banking scenario and more specifically on a typical loan origination process. The necessary assignments of rights, roles, and tasks will be considered from the security and trust standpoint, which is central to AVANTSSAR. A more detailed explanation of the loan origination process goes beyond the scope of this deliverable and the interested reader should consult [19].

Loan origination is a bank's business process to formalize, evaluate and possibly accept a customer's request for a loan. The bank carries out a careful evaluation of the customer's credit worthiness through internal mechanisms and external

agencies called credit bureaus. A credit bureau is a third party business partner of a financial institution that processes, stores and safeguards credit information of physical individuals or industrial companies. Credit bureaus gather data from various sources and cross-check and match the data for accuracy. Some of these sources include publicly available records (courts and deeds offices) and credit account details (from credit grantors or subscribers).

Credit grantors in turn are companies such as banks, retailers and any other organizations whose business includes credit. They are also called 'subscribers' because they subscribe to the credit bureau in order to collect, submit, use and share the information held in the database management systems. They use the information from the credit bureau to make decisions on whether or not to grant credit, in terms of their own credit granting policies.

Let us now outline a typical story of this scenario. John is a single man 25 years old. Recently, he was appointed as teacher in a primary school on a permanent contract. His gross salary is about 25,000 € per year. Before getting this job, John was a student and he was living at his parents' house. Though he gets on well with his parents, he realizes it is time to move forward and to obtain his own independence. By coincidence, a friend of John's who is about to move abroad advertises his flat for sale and he proposes John to buy it. The price, 100,000 €, and the status of the flat lead John to the conclusion that it would be really an opportunity. Despite the fact that affording this expense will require some efforts, John decides to go for it. Since he was a child John is saving money on a bank deposit account opened by his parents at the BBB bank. At the moment, John's account exhibits a positive balance of 10,000 €. Therefore, John decides to apply to BBB bank for a loan of 90,000 €. On the basis of John's positive records, it seems fair to expect that the bank will grant him the loan.

It is useful to introduce the actors that will participate in the scenario before the detailed description of its scenes.

BBB bank is headquartered in London and is one of the largest banking and financial services organizations in the world. BBB's international network comprises over 9,500 offices in 76 countries and territories in Europe, the Asia-Pacific region, the Americas, the Middle East and Africa. Through an international network linked by advanced technology, including a rapidly growing e-commerce capability, BBB bank provides a comprehensive range of financial services: personal financial services; commercial banking; corporate, investment banking and markets; private banking; and other activities.

John is a single man 25 years old. He has a permanent work contract, earning about 25,000 € per year. He needs a loan to buy his flat. He plays the role of the customer of the BBB bank.

Peter is a rather new BBB bank employee as he was only hired two years ago. He works as a pre-processing clerk. The pre-processing clerk is responsible for receiving and identifying the customer. He has to launch the loan origination process and check data of the customer information file.

Paul is a 10 years experienced BBB bank employee and he plays the role of the post-processing clerk. The post-processing clerk is mainly responsible for performing all banking transaction for the customer and checking the credit worthiness in case of a loan.

Ted is a manager in the BBB bank. He is responsible for leading a local agency of the BBB bank. He can be involved in all the critical steps of the loan origination process but he usually intervenes only when the situation requires his approval or supervision.

William is the general director for the BBB bank. A general director manages the entire bank.

Credit bureau is a third-party business partner that processes, stores and safeguards credit information of physical persons and industrial companies. It provides the BBB bank with information about the credit worthiness of John.

BBB bank Internal Computer System includes the information files for the customers and a software for calculating the price of a bundled product. Each *customer information file* stores all information about a customer, such as saving accounts, current account, shares and obligations, loans, etc.

Greg is the specialized clerk for enterprise account. This clerk is required by the BBB bank to process loans for enterprises.

Jack is the private banking manager. He offers adapted services for rich and important customers.

The loan origination process comprises four scenes, as outlined in [Figure 1](#). Some interactions between the sub-processes are indicated. The figure also depicts where each sub-scene takes place, along with the participating actors and the back-end applications.

Scene 1: John goes to the bank. At the BBB bank, John is introduced to Peter, the bank pre-processing clerk. John explains to Peter his situation and his intention to buy a flat by means of a loan covering 90% of the flat's cost. As Peter needs to retrieve John's data from the customer information file, John authorizes

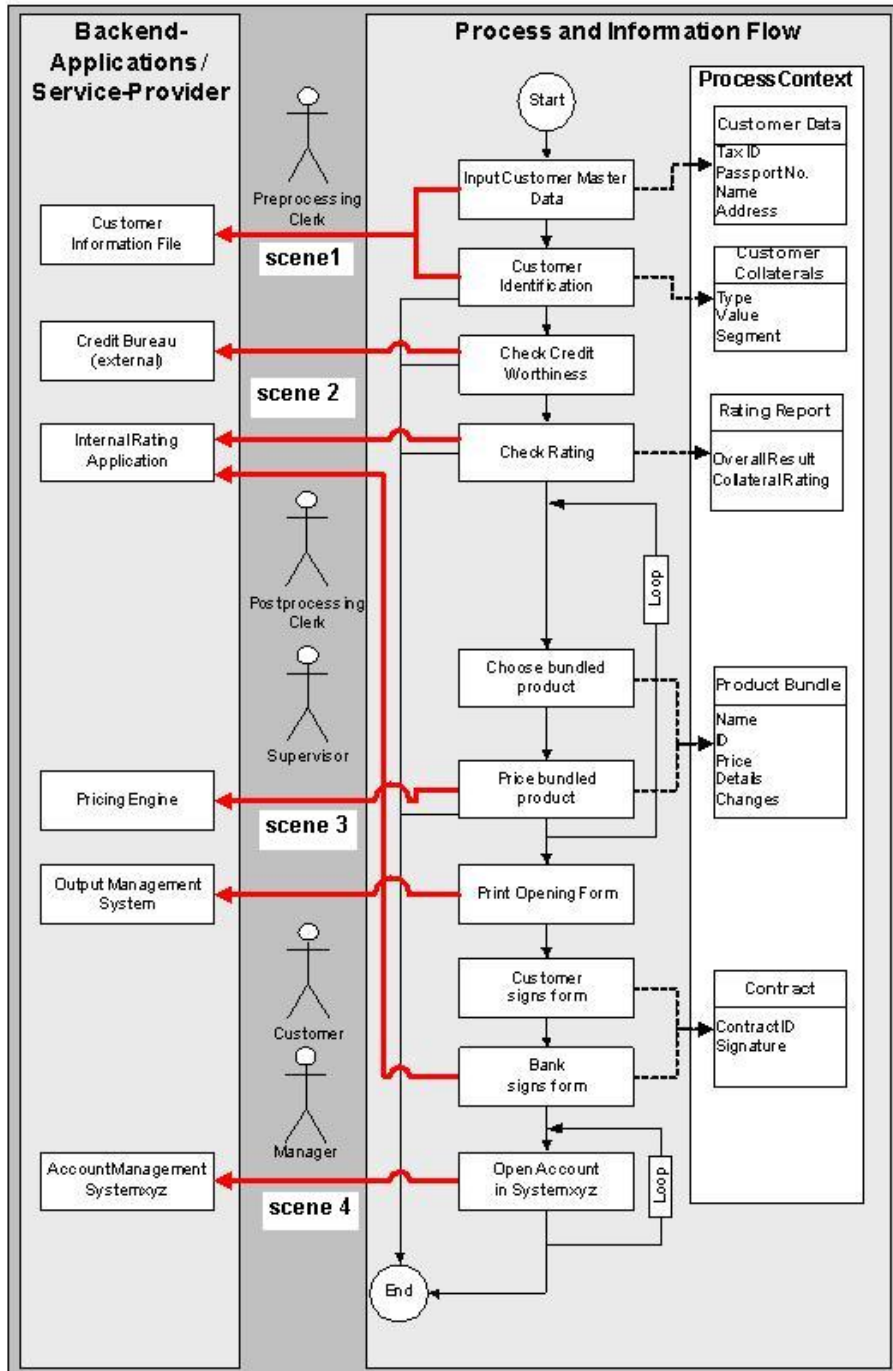


Figure 1: The loan origination process and its sub-processes

him to do so by means of his electronic signature. BBB is subscriber of a credit bureau, and it is therefore likely that this authorization also includes a standard notification and consent clause to notify John that personal information and payment behavior may be accessed by and supplied to a credit bureau for risk management purposes.

Peter processes John's confidential data. The history of John's bank account is definitely regular. Moreover, John has been a BBB's customer for a long time and so have his parents. In addition to this good saving profile, the most important credential for John is the permanent and stable job position he has recently obtained within a public administration. In conclusion, Peter's feedback on John's credentials is extremely positive at this stage. These comments are signed by Peter and reported in both the customer information file and the process log. The loan origination process can move to the next phase.

Scene 2: The bank checks John's credit worthiness through a credit bureau.

However, the inspection carried out by Peter is not enough to provide the level of assurance required by the BBB bank and John still has to interact with the post-processing clerk. This clerk is responsible to double check the credit worthiness of John by means of a more comprehensive risk analysis involving a larger set of data including sums of liabilities, sums of assets, third-party loans, reasons for rating, etc.

It must be noted that the majority of this data is collected and maintained by institutions different from BBB. The access to this information is regulated by appropriate collaborative services. These services behave accordingly to precise security policies defined and enforced through an authorization infrastructure that relies on pre-existing trust relationships between the data owner and the BBB bank.

Paul is chosen for this task and, because John required less than 1 million € as a loan, Paul's manager is not required to take over. Paul proceeds in the post-processing phase by querying the credit bureau first and the internal rating application afterwards. When Paul stores the resulting positive scores, the process can move on to its third phase. In case of negative scores, Paul's manager would complete the request assessment.

Scene 3: The bank calculates the price for the bundled product. Paul has to choose the most appropriate bundled product for John among those available in the product database. He queries the pricing engine service to compute a price for the bundled product (notice that this query does not need the real identity of the customer as an input). The result in some intelligible form, quoting for example original price, customer segment special conditions, customer company special

conditions and asset limit for price, is returned to Paul and then proposed to John.

Scene 4: John and the bank sign the form. John is now introduced to Ted, a manager of the BBB bank, to discuss the bundled product in more detail and to finalize the process. After some negotiations, John and Ted finally come to an agreement on the loan conditions. The contract is digitally signed using the respective secure signature creation device (SSCD) of John and Ted. A copy of the contract is printed for John's own record.

When the contract is signed, Ted updates the BBB information system with the remaining data. Finally, he provides John with a formal document stating that the amount of the loan will be credited to John's bank account in due time.

3.1.2 Security and trust requirements

This section lists the security and trust requirements for the banking services scenario. Therefore, the sub-processes forming the loan origination process that were discussed above will have to be scrutinized exactly in terms of security and trust. It must be remarked how privacy often steps in as an important aspect of security and trust. In the following, the requirements are grouped by sub-process, so it is useful to continue reading with a glance at [Figure 1](#).

The *Personally Identifiable Information (PII)* is particularly important from the trust perspective. PII is "any piece of information which can potentially be used to uniquely identify, contact, or locate a single person" [17] and therefore not everyone should be trusted to handle it. For example, in the banking scenario the primary PII of a customer typically is his passport or identification number. It is important to distinguish the PII from other private information such as gender or political preferences. It is understood that the current social context imposes any banking decision to only be taken upon the PII. This abstract requirement is expressed below by a few requirements concerning privacy.

Input Customer Master Data.

1. The customer shall authorize (some of) the bank representatives to handle his PII privately: they shall not disclose the PII to anyone without the customer's explicit consent. No-one except these bank representatives shall access the customer's information file without requesting and obtaining the customer's explicit authorization.
2. The pre-processing clerk, who assists the customer in this sub-process, shall not note the customer's PII anywhere else than in the customer's information file.

3. The customers' information files shall be managed within the bank's internal computing system, which shall be certified according to established security evaluation standards such as the Common Criteria (CC) [8].
4. Any form of customer's authorization shall be provided by the customer's signature, either in digital or in classical form.

Customer Identification.

5. If the customer is a private citizen, his PII shall be verified by the pre-processing clerk who carried out the previous sub-process.
6. If the customer is an industrial customer, his PII shall be verified by the specialized clerk for enterprise account.
7. If the loan exceeds 1 million €, customer identification and all subsequent sub-processes shall be performed by the private banking manager responsible for the very important customer.

Check Credit Worthiness.

8. There shall be no fraud exploiting combinations of pre-processing and post-processing. This often amounts to requiring that for each process the person acting as post-processing clerk differs from the one acting as pre-processing clerk.
9. The customer shall authorize the bank to transmit his PII and his financial information to credit bureaus and their respective delegates for external evaluation of his request.
10. The credit bureaus shall not be able to process the bank's query about the customer without receiving the customer's authorization from the bank.
11. Information exchanges between the BBB bank and the credit bureaus shall ensure confidentiality, integrity and mutual authentication. (It is worth noticing that in most of cases this information exchange concerns customer's sensible data.)
12. The credit bureaus shall ensure that access to credit profiles is only granted to those authorized by the customer.
13. The credit bureaus' IT infrastructure shall be certified according to established security evaluation standards (e.g., the Common Criteria).

14. The credit bureaus shall retain the customer's PII for no more than a fixed period of time, which is three years for non-fraudulent data.
15. The credit bureaus shall not seek or maintain any private information except the customer's PII. (For example, race or political opinions are not of financial nature and therefore irrelevant to evaluate credit worthiness.)

Check Rating.

16. There shall be no fraud exploiting combinations of internal and external ratings of the customer. For instance, if response from the credit bureau is negative, then the post-processing clerk who performs the internal rating shall differ from the one who contacted the credit bureau.
17. If the internal rating is negative, the post-processing clerk shall inform the manager, who shall validate the outcome.

Choose Bundled Product.

18. The outcomes of the internal and external ratings shall in no circumstance be published.
19. If both internal and external ratings are negative, then, if the manager still wants to grant the loan, he shall seek approval from the general director.

Price Bundled Product.

20. The customer's PII shall not be sent to the pricing engine of the bank's internal computing system.

Print Opening Form.

21. A fair non-repudiation of the signature of the contract shall be ensured to both parties.
22. If the loan exceeds 1 million €, the manager shall sign the form instead of the post-processing clerk.
23. Appropriate cryptographic techniques shall be used to store the contract so that only the bank's representatives that are authorized can access it.
24. Appropriate cryptographic techniques shall be used to store the contract so that it cannot be modified without the customer's authorization.

Some of the requirements listed above are non-technical requirements that are intangible to formal methods. In detail these requirements are:

- Requirement 2 and 18 are non-technical requirements. For example, it is not possible to technically prevent the pre-processing clerk from noting the customer's PII anywhere else but in the customer's information file once the pre-processing clerk has the information in his mind. A possible organizational solution is to put a dedicated instruction into the clerk's contract of employment defining the terms of PII handling the clerk must adhere to. The same holds for Requirement 18. Of course, adequate authorization policies can constrain the access to confidential internal and external ratings, but the main issue of this requirement can only be solved organizationally by, e.g., signing a non-disclosure agreement.
- Requirements 3 and 13 are non-technical as well such that they have to be handled on the organizational layer. Nevertheless, security certification may improve the trust of the customer on the bank.

Though the security certification process itself cannot be formalized, we can model the result of it, e.g., by introducing a trusted third party stating whether a certain system has been certified according to the CC.

It is important to remark that all requirements stated so far are *security* requirements. A broad notion of security is being applied here, including confidentiality, privacy, authentication, non-repudiation and dependability. *Trust* requirements require special consideration, as they can be informally denoted as pertaining to a principal's level of certainty about a certain fact.

For example,

“the customer shall trust that the his/her information file shall be managed within the bank's internal computing system, which shall be certified according to established security evaluation standards such as the CC.”

requires a high level of certainty to the customer about the fact that the bank will not abuse of its role and will properly treat the customer's sensible information. This also requires that the bank's system has been certified to work fine. This trust relation between the customer and the bank is normally established by means of laws, liability contracts, mandatory auditing processes, etc that strongly discourage the bank to behave improperly. Reputation systems and certification authorities can also be used to establish and manage trust relations within the involved entities.

We observe that each security requirement stated above can be lifted over a principal's trust whenever the requirement's subject (in the grammatical sense) differs from the principal and the principal cares for the requirement's object (in the grammatical sense). The example just given demonstrates how to lift requirement 3 over the customer's trust. In general, it is clearly meaningless to lift a requirement having a principal as subject over the trust of the same principal because arguably each principal trusts himself. For example, it would be pointless to lift requirement 3 over the trust of the bank's internal computing system or of the bank itself (which is assumed to trust his internal computer system to work as expected unless the bank opts for outsourcing the management of such a system). By contrast, it is meaningful to lift the same requirement over the customer's trust, producing the requirement in italics. Lifting a requirement is also pointless if the principal has no interest in the objects of the requirement. For example, the credit bureau has no interest in the bank's internal computing system, but the customer has interest in how his/her data are exchanged between the bank and the credit bureau. Requirement 11 can thus be lifted over the customer's trust on both the bank and the credit bureau. According to this requirement, the bank and the credit bureau are basically asked to exchange securely the customer's sensible information. The customer's perception of this security measure can be expressed by means of the lifted trust requirement

“the customer shall trust that information exchanges between the BBB bank and the credit bureaus shall ensure confidentiality, integrity and mutual authentication.”

Interesting enough, a possible way to fulfill this trust requirement would be the AVANTSSAR Tool that, after validating the security requirement 11, acts as a trusted authority and issues a corresponding certificate that the customer can check.

Notice how the lifting approach can be applied on both non-technical (e.g., lifting of requirement 3) and technical (e.g., lifting of requirement 11) security requirements.

Therefore, if we lift the requirements stated above, except those with the customer as subject, over the customer's trust, then we obtain a set expressing the total customer's trust that the whole loan origination process enjoys the necessary security conditions. The same can be done with the bank, producing yet another set of trust requirements. The trust requirements considered in this deliverable can always be obtained by lifting.

The requirements introduced here produce a variety of problem cases. These problem cases are grouped into several families of problem cases and are presented in the next sections. Each problem case in a family is described according to a simple template whereby the problem case and the requirements that produced it are stated first, and then a possible solution to the problem is hinted at.

3.1.3 Authorization Policies

Access Control. In computer security, authorization policies are used to restrict the access to resources, e.g., files, computer programs, computer devices, and functionality provided by computer applications or services, only to those principals permitted to use them. Principals can be human, computer programs, services, or other devices. For example, if the customer formally authorizes some bank representative to enter his PII into the customer's information file, then the authorization policy will allow that representative to access and modify that file. Policies should always be modeled based on the principle of least privilege, i.e., consumers should only be granted permissions they need to accomplish their tasks. Authorization depends on the correct authentication of customers which is a separate task that has to be done before authentication.

The requirements of the banking scenario raising problem cases that can be addressed by authorization policies are originating from the need to handle the customer's PII and private information in accordance with the customer's privacy. The PII cannot be disclosed or generally used without the customer's explicit consent, as stated by requirements 1, 4, 9, 10, 12, and 23.

Solving this problem requires a variety of measures. Because the customer's PII is to be handled by a number of bank employees, credit bureaus and representatives, the customer must sign an agreement to explicitly authorize these information flows (but this is an organizational task that will not be captured by AVANTSSAR, just as the following task.). The bank and the credit bureaus must not operate outside the agreed terms. The access to PII and private data can be restricted using existing authorization mechanisms like access control lists or capabilities. Cryptographic techniques can be used to restrict access to confidential data, too.

3.1.4 Accountability

Non-Repudiation. Requirement 21 produces the problem case of non-repudiation of contract signing. Non-repudiation means that it must be ensured that, towards a third party, none of the signers of a contract is able to repudiate the signature in a dispute later on (non-repudiation of origin and non-repudiation of receipt), or to refuse the validity of the contract (non-repudiation of content).

This kind of problem can be solved using a combination of digital signatures to sign a document (more detail in [subsection 3.5](#)) and fair-exchange protocols.

3.1.5 Trust Management

Service-oriented computing is particularly suited to support collaborative work, with each of the collaborators being represented by a (set of) service(s) that estab-

lish a coalition to achieve a common goal. It is characteristic of such a coalition that not all of the collaborators know each other in advance, so that trust becomes crucial: each of the involved principals has to be convinced that the others behave according to the rules of the coalition, which indicates that they abide by the security policy defined for the coalition. The establishment of such a coalition may be dynamic, meaning that the coalition is born exactly to reach a goal, and dissolves when the goal is reached.

The security policy defined for the coalition is often the result of a negotiation. Upon the formation of the coalition, the services representing the individual entities present proposals and follow a negotiation protocol to achieve a consensus on the security policies to be enforced by the coalition. The negotiation strategies are part of the service specification, and they are guided by the local security policies of the services. Such strategies take the trust level of the services into account: the more trusted a service is, the more willing are the collaborating services to accept a proposal from that service.

The scenario we are currently studying rests on a coalition of principals. The customer, the bank and its representatives, and finally the credit bureau form a coalition whose goal is to decide the granting of a loan. It is clear that a positive decision is a common aim, because it provides value to the customer and profit for the remaining parties. However, a negative decision is equally desirable for everyone when it becomes necessary to impede inconvenient business.

The coalition in this scenario is mostly static, as the bank typically has established collaboration relationships with specific credit bureaus. The dynamic aspect is represented by the customer who joins the coalition to request the loan. Strictly speaking, there ought to be a policy negotiation for the newly-formed coalition (despite its static kernel) but in the real-world this phase is essentially emptied. The customer is typically required to sign a pre-established authorization form whereby he authorizes the bank representatives and a fixed list of credit bureaus to handle his private data appropriately, that is privately.

Because of the simplified negotiation, one wonders whether the loan origination process has significant requirements in terms of trust, producing non-trivial problems that can be addressed by trust management. The answer is still affirmative, due to a number of requirements seen above ([subsubsection 3.1.2](#)), which are pinpointed in the sequel of this subsection. However, before the problem cases in this family can be defined and solved, the general techniques to managing trust must be introduced. Here, we introduce and describe the three main techniques we outlined in the project proposal of AVANTSSAR.

Trusted Third Parties (TTPs) assert security properties (like the ability to enforce a particular policy) to a service. If the TTP is actually trusted by an entity, its assertions are accepted as valid. Attribute certificates are a means

to establish trust through TTPs.

Event histories are evaluations of previous engagements between the involved entities (and the services representing them). Such evaluation results are used, together with the current context, to assess the trust to put on each other.

Reputation systems extend the above to the use of trust evaluation results for a population of services. Evaluation results are collected and published through the reputation system, so that each service can use them to evaluate its own trust assessment.

The natural focus of our project lies exactly upon the techniques that are used for managing trust rather than on the algorithms or logics that provide the actual evaluation results. In consequence, solving a problem case in this family requires choosing and tailoring one or more of the three techniques stated here for trust management.

The customer does not necessarily know the bank or the credit bureaus as secure and trustworthy institutions. Due to more stable collaborations between the bank and the credit bureau, the trust between them should be no issue. Trustworthiness in this context can be seen as honesty in doing business, that is to avoid exaggerate speculation. It is widely acknowledged that trustworthiness can be ensured not only by ethical codes of conduct but also by the open market competition. In brief, the customer would like to be able to believe that the bank is not tremendously speculating on his loan offer. Also, he would like the entire transaction to be secure in the general computer security sense, namely at least authenticated and confidential.

The customer's establishment of the bank's trustworthiness is a problem of trust management, meaning that appropriate trust management techniques can address it successfully. Precisely, it is the problem of meeting the security requirements 3 and 18 together with those obtained by lifting the requirements for the current scenario (subsubsection 3.1.2) over the customer's trust. As stated above (p.18), the requirements that can be meaningfully lifted over the customer's trust are those whose subject is not the customer. For example, lifting requirement 5 produces the meaningful requirement "*A customer who is a private citizen trusts the bank to verify his PII through the pre-processing clerk who carried out the previous sub-process*". Because this problem case is about the customer's trust on properties of the *bank*, also those requirements whose subject is the credit bureau are not to be lifted, as they would not contribute to the definition of the problem. In brief, this problem case originates from the security requirements listed here (i.e., 3 and 18) along with the trust requirements obtained by lifting the scenario

requirements whose subject is not the customer or the credit bureau over the customer's trust.

Even though some of the stated requirements are non-technical—as already mentioned—trust management can be used as an instrument to provide confidence for the customer, that these requirements are fulfilled. For example, a reputation system may offer the customer indications of how the bank performs in the area of loans, as a number of other customers perceive. The techniques of TTP and event histories may be combined into a TTP that exposes event histories as a trusted service. The service might offer the details of a variety of loans previously decided by various banks, which would help the customer in his own trust assessment. Of course, all details would be anonymized.

3.1.6 Workflow Security

A workflow is a collection of tasks that together achieve a particular business goal. More precisely, a workflow defines the individual tasks, the control flow between them, the parameters passed between the tasks upon initiation and completion, and the constraints applying. Tasks are assigned to users being responsible for their execution, and a workflow typically involves the activities of multiple users through appropriate task assignments.

The security requirements of the banking services scenario induce two different problem cases in this family: flow control and separation of duties.

Flow Control. Flow control means the problem of enforcing obligations in a workflow. Those obligations requiring for, e.g., task supervision are clearly related with security and trust. A number of requirements cause this problem case. These are requirements [5](#), [6](#), [7](#), [17](#), [19](#), and [22](#). The enforcement of obligations in the workflow is the problem of ensuring that specific portions of the workflow exist. If specific portions do exist, then they provide a meta-level enforcement of obligations because they rule out all inadmissible portions.

As a hint at the solution, these problems can be addressed by designing the workflow accordingly to what each problem requires. For example, requirements [5](#), [6](#) and [7](#) raise obligations on which principal performs customer identification. These can be enforced by a dedicated workflow portion embedding a case analysis. Likewise, addressing the problem raised by requirement [19](#) for example, demands a flow that involves the general director. On top of this, techniques of workflow simulation would obviously be necessary.

Alternatively, obligations can be expressed as rules and enforced by means of rule engines. This is specially suited for those situations where flexibility in the workflow is required without changing the workflow itself. For instance, many banks might follow the same workflow for the loan origination process, but they

might need to customize it. E.g., different country regulations might demand for some customization, but rather than having a different workflow for each different country, we can have a single workflow and to act on the rules to customize it according to the country (e.g., if the country is Italy, then data retention time at Credit Bureau must be three years; if the country is Germany, then it must be five years; etc).

Separation of Duties. This problem case is raised by requirements 8 and 16. We use requirement 8 in the following to explain the problem, i.e., to prevent a fraud that exploits a combination of the pre-processing and post-processing phases. The preprocessor, who essentially deals with customer identification, may for example have his best friend as customer. If that preprocessor also acts as post-processor for the same customer and the same loan request, then he may decide to act illegally to favor his friend.

Such problems can be faced using separation of duties techniques, i.e., context-dependent limitations of a principal's authority to prevent (in)advertent error and fraud. Separation of duties can be seen as a design principle for the protection of information in computer systems and a mechanism to control fraud and ensure the consistency of the data objects. Separation of duties concerns with dividing the responsibility about sensitive information so that no individual acting alone can compromise the security of a business process. A simple way to enforce a separation control is to prevent a single principal to own all the necessary authorizations for each required step of a process. A more relaxed alternative would be to prohibit him to perform all the steps on his own. This is sometimes referred to as a dual control or four-eye principle, since two or more people are needed for the execution of a critical process. The problems that no-fraud requirements cause are solved by separation of duties by a careful user-to-role assignment strategy of one of the following five kinds.

- I. A principal must not be a member of any two exclusive roles (*static separation of duties*).
- II. If a principal is a member of any two exclusive roles, then he must not activate them at the same time (*dynamic separation of duties*).
- III. If a principal is a member of any two exclusive roles and activates them at the same time, then he must not act upon the same object through both (*object-based separation of duties*).
- IV. If a principal is a member of exclusive roles, then the set of authorizations acquired over these roles must not cover an entire workflow (*operational separation of duties*).

- V. If a principal is a member of exclusive roles, and the set of authorizations acquired over these roles cover an entire workflow, then the principal must not use all authorizations on the same object (*history-based separation of duties*).

Solution I is clearly *static* because it only relies upon the assignment of individuals to roles and the allocation of tasks to roles. All other solutions are *dynamic*, as they can only be enforced during an instance of the workflow, that is at run time. Such a distinction can be easily motivated and demonstrated: “The objective behind dynamic separation of duties is to allow more flexibility in operations. Consider the case of initiating and authorizing payments. A static policy could require that no individual who can serve as payment initiator could also serve as payment approver. This could be implemented by ensuring that no one who can perform the initiator role could also perform the approver role. Such a policy may be too rigid for commercial use, making the cost of security greater than the loss that might be expected without the security. More flexibility could be allowed by a dynamic policy that allows the same individual to take on both initiator and approver roles, with the exception that no one could authorize payments that he or she had initiated”[11]. It can be seen how the problem of preventing fraud is solved dynamically by using both role and principal identifier in authorizing tasks: even if a principal can take both initiator and approver roles, he will not be allowed to both initiate and authorize a payment.

Solutions get more and more flexible from II through to V—and V also is the most recent version introduced in the literature [21]. They can variously address problems deriving from fraud-prevention requirements, as we shall see with the next two problem cases.

3.1.7 Privacy

Although privacy is a concept used in various contexts with often changing meanings, its most general definition appears to be as follows: “the right of an individual to decide when and how sensitive personal information should be revealed” [14]. In consequence, a scenario that sees all principals purposely and gladly publishing their fingerprints etc. does not contain any problem cases in this family, as the principals themselves decided to authorize access to their information.

It is important to remark that private information may either be *static* or *dynamic*. The former is what characterizes a principal since its birth, such as name-surname, gender, hair color, iris layout, fingerprints and so on. As outlined above, static private information may be further split up into PII and non-PII. It is clear that while the iris layout is PII because it uniquely pinpoints the principal, his hair color is not, and his race either.

Dynamic private information of a principal is any private information that the principal is not born with. For example, anything that he chooses to buy or to enjoy or to believe in qualifies as dynamic private information. It is easy to convince that dynamic private information is dramatically important to business in general. One paradigmatic example is the constant monitoring of supermarkets over their sales, which leads to reducing the prices of specific products during certain times of the year or even during certain daily time windows.

It seems interesting to the subsequent development of AVANTSSAR that privacy in abstract terms boils down to *associations*. If a customer sends to a merchant a message stating “Coca-Cola”, then an attacker may violate the customer’s privacy by noting down the association customer/merchant/Coca-Cola. This also demonstrates the subtle difference between confidentiality (secrecy) and privacy. The name of the popular drink is in fact public, so confidentiality is not an issue here. However, its association to the specific pair customer/merchant is the sensitive information, which indeed belongs to the customer’s dynamic private information.

Apparently, authorization policies are one possibility to face privacy requirements, see [subsubsection 3.1.3](#). Requirements being subject to privacy and authorization policies are always classified into the most specific family of problem cases to avoid unnecessary complexity. I.e., in this subsection, only privacy related requirements are handled that are not tackled by authorization policies. We will describe the various problem cases subsequently.

Data Austerity. Data austerity means that when accessing a service, as few personal data and other application-related information (if any at all) may be collected as technically and organizationally possible. In other words, a service is only allowed to collect data that is absolutely necessary to accomplish the service. There are several country-specific legal requirements concerning data austerity that services have to comply to. Since data austerity avoids the acquisition of, e.g., PII, it is the most effective protection of privacy.

In the context of the banking scenario, data austerity is relevant to the loan origination process. To make sure, that an external rating of a credit bureau is calculated without any discrimination based upon private information other than PII, such as gender, race, political opinions and the like, only PII must be used, see requirement [15](#). Another requirement raising a very similar data austerity problem is requirement [20](#).

Incidentally, it seems desirable in the present social setting to state the same requirement about the bank, although most banks still at present do appear to record the customer’s gender.

Solving this problem case requires the bank not to disclose any non-PII to a

credit bureau. Additionally, it requires the credit bureaus to sign a declaration that they will not seek other static private information beyond the PII. This signed agreement must be made available to the customer.

Whenever such information is stored in the system of the bank for plausible reasons such as statistics, the authorization policy shall keep it secret from any principals involved with the customer's rating. Therefore, the policy shall make sure that at least the post-processing clerk is prevented from accessing non-PII static information about the customer.

Data Oblivion. Another privacy-issue for customers is the right of oblivion, i.e., the right to erasure of obsolete data. Just as data austerity, there are several country-specific regulations on data oblivion. Requirement 14 demands that credit bureaus store PII for not longer than three years.

A possible solution to this problem is the definition of appropriate policies stating how long data is retained. Currently, the enforcement of such policies could be done by a legal framework or bureaucratic controls.

3.1.8 Application Data Protection

Data Integrity. Requirement 24 raises the problem of application data protection, more precisely, the protection of the integrity of the signed contract. In contrast to subsection 3.1.9, data must not be protected only a short time during transportation, but as long as the contract is stored in a computer system. For the customer as well as for the bank it is crucial, that neither the bank nor the customer alone is able to modify the contract without mutual agreement.

A threshold cryptography access mechanism can be used in this context so that the secret that is necessary to modify the contract is split into (at least) two portions. One is given to the bank and its employees, the other to the customer. Modification of the contract then requires both portions to be submitted. The customer's portion is only stored in a smart card or smart token that is given to the customer and nowhere else. However, accessing the contract without permission to modify it can be done by supplying the bank's portion alone. Another possibility would be to consider, as in OpenTrust's digital contract signing 3.5.1 a trusted third party that will protect the loan agreement and ensure it is only accessed in ways agreed upon by the customer and the bank.

3.1.9 Communication Security

A communication security problem case is raised by requirement 11. Communication security may mean that during transportation of data, one or more of the following three security needs have to be fulfilled.

Message Confidentiality. No-one except the intended recipient of a message must be able to read the content of the message.

Message Integrity. No-one must be able to modify the contents of a message unnoticed.

Peer Authentication. A principal is assured of the identity of its communication partner.

The problem of communication security has been researched broadly and there are plenty of solutions to the problem. Mutual authentication can be done using a trustworthy certificate authority issuing digital certificates. Data confidentiality and integrity can be assured using cryptographic techniques like encryption and digital signatures. Another simpler possibility would be to rely on secure channels at transport layer—such as those provided by e.g., the TLS/SSL protocol—for communications to and from the BBB bank.

3.2 Software Distribution Services

A very general and widespread ICT infrastructure scenario is secure software distribution (also known as software upgrade). The Siemens project partner currently runs multiple projects that involve secure Software Distribution Services (SDS).

More precisely, we consider security and trust requirements appearing in distributing software (i.e., code and/or data: parameters, configuration data, keys, certificates, policies, multimedia contents, navigation data, etc.) from content providers to a potentially large and widespread set of consumers. Typical target devices include

- safety-critical embedded systems in e.g. cars, engines, airplanes, power plants, industrial production equipment, ...
- security critical ICT infrastructure components like routers, firewalls, mobile phones, ...

3.2.1 Scenario definition

Our scenario is about “numerical control” software for robots (or other production machines) used in industrial automation, but can be transferred easily to e.g. airplane software distribution [18, 15]. Therefore, we describe the software distribution service (SDS) scenario in a rather generic fashion.

Software items are produced by software *suppliers* and are initially integrated with the target device hardware during their production at the Original Equipment Manufacturer (*OEM*). Typical reasons for software upgrades are bug fixes and/or functional extensions. To this end, the OEM collects software upgrades from the software suppliers and ships them to software *distributors*, which in turn make them available to the target *owners*. The owners bear responsibility for the safe and secure operation of their target devices, and ultimately decide which software items get loaded and installed on which device. The software may be handed to a target *operator* who executes load and installation orders on the *target device*. Thus the software distribution process consists of several hops, and the SDS stretches over the IT systems related to the process at each of the roles involved. [Figure 2](#) shows the overall flow of software items.

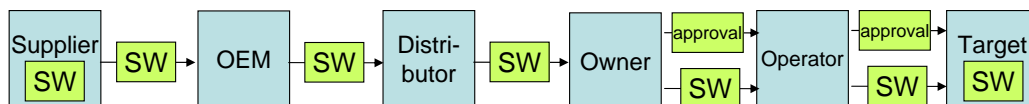


Figure 2: SDS overall architecture

Simpler scenarios are possible, e.g. where the distributors coincide with the OEM or even with the suppliers and the owners coincide with the operators. The transfer of software items may be initiated by basically any role, in particular by a software supplier (pushing upgrades directly or sending notifications on their availability) or a target operator (requesting or pulling upgrades).

3.2.2 Security and trust requirements

The typical security requirements for SDS are as follows.

Integrity of the software items and authenticity of their origin, which may be crucial for safety or security reasons.

1. Any changes to the contents of the software items, all the way from the suppliers to the target devices, must be detected, and manipulated items must not be accepted. This must hold in particular between all suppliers and the OEM, between the OEM and all target owners, and between each target owner and its target devices.
2. Software items must only be accepted from trusted sources. For instance, the OEM maintains a list of trusted suppliers, the target owners trust the OEM and possibly one or more distributors, and the target devices trust only their owner or one or more operators.

Approval of software load and installation, pertaining the soundness of the software configuration of the target device and the freshness of the software.

3. The target owner may specify policies constraining the relation between the type and identity of target devices, their current software configuration, and the types and versions of software installable on them. Any such policies must be enforced by the target operator or the target device itself. For instance, the policy that existing software may be replaced only by newer software (of the same type) aims at preventing version rollback attacks.

Protection of intellectual property for software and the commercial value of software licenses, which implies that protected software items cannot be accessed by parties not having a license for possessing and using them.

4. Confidentiality protection of software items all the way from the suppliers to the target devices, such that in each step the sending node determines which receivers are allowed to get read access to the software. For instance, the distributors determine which software items they make accessible to which target owners. This also involves the authenticity of the receivers.

5. Software licenses may be issued by the distributors and must be enforced at the target devices, such that any software item can be installed and run only if allowed by its distributor. Licenses may have an restricted lifetime.

Accountability for various activities, which may be required for forensic and legal reasons. In general, logs are not sufficient as evidence to a third party. Accountability information may need to be very long-lived (e.g., 50 years).

6. Any local processing of software items at any role must be logged in a secure way, for instance the initial signing of a software package at the supplier and the inspection of the software items by the target owners. Logs are typically stored in the SSVs or in their secure local environment.
7. Non-repudiation of origin for all nodes in the distribution chain and non-repudiation of receipt at the target device.

For this project, payment (if any) for software upgrades is considered out of scope, as well as content availability, which may be required for service continuity.

The distribution process may be seen as an application-level cryptographic protocol, yet with extensions to accommodate, e.g., business process policies. The SDS scenario involves certain dynamics and flexibility: instances of various roles mentioned above, e.g. suppliers, may join and leave the SDS, and target devices and other nodes may be intermittently off-line.

Trust relations can be assumed or established only between certain parties, since for instance the OEM does not even know about all the companies ultimately owning and running its products. Further, even if the distributor knows all its clients (which are the target owners), it cannot trust them to run only the number of software installations for which they have obtained a license. On the other hand, a certain level of trust is required among the nodes in the distribution chain: the suppliers and the OEM trust the distributors to properly enforce their licensing policies. The target owner trusts its target operator. Each node trusts at least one of its predecessors to provide only benevolent and correct software (unless the node is able to independently verify the quality of software items received). In particular, the software supplier needs to be assumed to develop and supply good software. Therefore, we can assume or establish that suppliers can be trusted by all other parties, in particular by the OEM. Distributors and device owners trust the OEM. Further such trust relations are already mentioned for requirement 2.

The integrity, authenticity, non-repudiation, and authorization requirements are typically achieved using digital signatures, whereas confidentiality is typically achieved using encryption. Both mechanisms require public-key cryptography or similar techniques. Key distribution is yet another (higher-level) instance of the

core of the software distribution problem: for each node in the chain, the public keys of trusted predecessors need to be made known in an authenticity and integrity protected way. A trusted public-key infrastructure (PKI) is typically employed to achieve this, which in turn involves a mixture of technical and organizational protection means. Authorization typically involves role-based access control and may involve license servers and Trusted-Platform-Module-like mechanisms built into the target devices or hardware dongles attached to them in order to enforce license restrictions. Challenges include

- management of certificates and secret keys, including their creation, distribution, and revocation (in case of compromise or expiration),
- trust management concerning in particular the receiver-sender relation (as far as approval of upgrades cannot be achieved by inspection of the software items) and the provider-customer relation,
- integration of application-level policies, e.g. version and license constraints.

For enforcing software license restrictions at the target devices, we assume for our scenario that the OEM has equipped them with a Trusted Platform Module (TPM). When a protected software item is installed on a target device, the device owner registers the installation at the distributor, giving both the identity of the software item and of the target device. At determined points in time, the TPM at the target device sends license requests for all installed protected software items to the distributor's license server, and disables those software items for which it does not receive (within a certain period of time) a license acknowledgment signed by the distributor. Technically, this procedure is a challenge-response protocol with replay protection.

Since public-key cryptography plays a vital role for the secure software distribution, crucial security services to be used are software signers and signature verifiers, PKI and related auxiliary services like time stamping. Each of these basic services has its own requirements.

The intermediate nodes in the distribution chain may just store and forward the software items, or in addition perform some local processing, such as including owner specific license keys or setting other target specific software parameters. In any case, each intermediary has to check the signatures applied by the previous node and may add new signatures. If in addition confidentiality is required, the sender encrypts the signed message for the intended receiver.¹

In order to exploit the inherent commonalities, one may structure the SDS into several instances of a generic signature application component called Software Signer Verifier (SSV). [Figure 3](#) shows an SSV instance as a “black box”

¹For large amounts of software, this is usually done in a hybrid way using an intermediate symmetric key that is encrypted with the receiver's public key.

in its application context. The application context consists of the preceding and subsequent SSVs (if any) in the distribution chain and a secure local environment in which software items may be processed locally. Each node in the above distri-

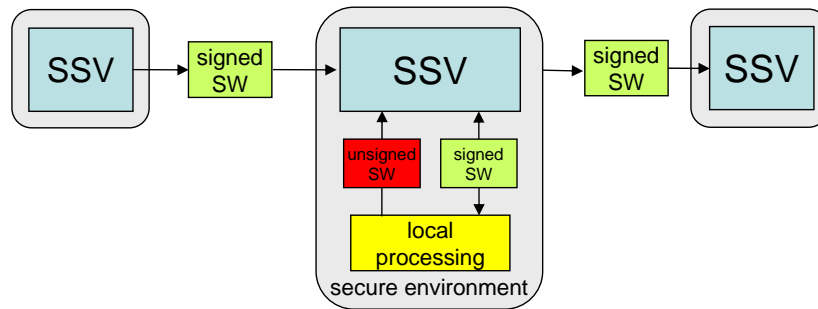


Figure 3: SSV application

buton chain runs an instance of the SSV. Each such SSV can be developed and certified independently, or instances of one and the same SSV product can be used at all nodes.

Each SSV instance is a service that in general offers the following functionality.

- Introduction of unsigned software into the SDS by digitally signing and optionally encrypting it and making it available for other SSV instances. This typically takes place at the supplier, yet may take place also at intermediate nodes.
- Verification of the signature on software received from other SSV instances (after decrypting it if needed) and of the authenticity and authorization of the sender. This takes place at all SSV instances except at suppliers.
- Approval of the software by adding a signature and optionally re-encrypting the software and making it available to further SSV instances. Adding a new signature is usually done at SSV instances located at the OEM and at the target device owner after some local processing of the software, such as adding license information or by performing quality inspection. Such processing is performed within the local secure environment of the SSV.
- Delivery of software out of the SDS after successfully verifying it. This takes place at the software target.

As the target owner is responsible for its target devices, it has the special task of managing the software configurations on the targets, i.e. deciding which software versions may be installed on which targets. This may be implemented by explicit signed installation approvals statements sent by the owner and interpreted

and enforced by the operator or the target device itself. Airplane software distribution typically uses out-of-band delivery for the installation approval: the target owner issues installation orders in the form of a work order on paper, to be executed by a mechanic. Similar processes might apply in software distribution systems if target devices are located in the vicinity of the owner and operator. For our SDS scenario, administration of the target device is done on-line under remote control of the owner sending a signed installation approval along with the distributed software package. We assume a secure local connection between the target operator and the target, such that the operator can enforce the authorization for the installation of the software item at the specified target device instance.

3.2.3 Authorization Policies

Access Control. As described in [subsubsection 3.1.3](#), authorization policies determine the rules for access control, e.g. to data files. In general, they determine which subjects may perform which actions on which objects.

Requirement 3 specifies adherence to installation approvals.

Requirement 4 prevents unauthorized read access to the software items.

Requirement 5 requires any licenses to be checked at the target.

3.2.4 Accountability

Audit Logging. Requirement 6: logging of local processing of software items.

Non-Repudiation. Requirement 7: non-repudiation of origin and receipt.

3.2.5 Trust Management

Trust Policies. Complementing the aspects introduced in [subsubsection 3.1.5](#), the SDS scenario contains trust management in the sense that certain entities maintain a list of other entities (in the form of their names and public keys) they trust. This list may be dynamic as new entities may come into play or trust in certain entities is revoked. Entities may rely on other entities (which they already trust) to assert which further entities may be trusted. In this way, chains of trust can be constructed, like in the PGP trust model [16].

Requirement 2 relies on policies that describe which sources of software items are trusted by a particular node in the distribution chain.

3.2.6 Application Data Protection

Data Integrity. Requirement 1 specifies integrity protection for software items.

3.3 Anonymous Shopping

3.3.1 Scenario definition

Scene 1: Shopping Privacy. We consider a typical online shopping scenario. Alice orders a box of wine bottles at an online winery. Usually, this requires Alice to provide her name, delivery address and credit card details. The winery's warehousing and delivery is outsourced to a logistics provider. Before delivery, the winery also checks that Alice's credit card is valid and good for the amount of money at a credit processor service that also takes care of the money transfer. All these entities—the winery, the logistics provider, and the credit processor—are able to store the data of this transaction. With all the stored data they can build profiles of their users. While there may be legitimate business interest behind this, Alice prefers the (relative) anonymity that she enjoys when shopping in classical off-line stores, paying cash, and taking the shoppings home by herself.² In order to obtain this privacy in the online world as well, Alice uses an anonymous credential system as described in the following.

In the anonymous online shopping scenario, Alice routes all communication, e.g. browsing the shop's website, through an anonymity network such as TOR [10]. This is needed to prevent that the shop or an intermediate node between Alice and the shop can link her actions by her IP-address. The technical details of this anonymization is not our concern here, however. Alice chooses a box of wine on the website and clicks on the check-out button. Alice may either create a fresh pseudonym for the transaction or choose a pseudonym created in a previous purchase at the store; in the former case, her actions are unlinkable, in the latter case she may benefit from a frequent customer bargain that the shop offers (but we will see below that such a bonus system does not necessarily imply linkability). When she uses a pseudonym for the first time, she must first prove that she is over 18 years old in order to purchase the alcoholic beverages.

The policy of the winery says that the legal age can be proved by showing an electronic passport that was issued by any OECD country, where the date of birth is at least 18 years ago. Showing, however, does not mean here that Alice transmits a copy of this credential to the winery, but rather she *proves the possession* of such a credential. In particular, she does not reveal her name or her date of birth.

In fact, the winery needs to tell Alice in a first step that they require a proof that she is over 18 according to an OECD passport. Let us assume that Alice owns such a credential. Still, it is her choice whether she agrees to give out the information the shop asks for, or rather cancel the transaction. The fact that Alice is over 18 and is, say, a Swiss citizen is probably not very sensitive information.

²Of course, privacy and security are not perfect in an off-line store either, but automated surveillance and profiling is more difficult.

However, in some cases the set of attributes that an organization asks for may be excessive and thus the users should be in full control of which information they reveal about themselves.

Due to the anonymity, the shop does not know the real name and address of Alice and thus they have no means to contact Alice if something goes wrong, e.g., if Alice does not pay later. In order to make actions accountable, Alice is further required to produce a *verifiable encryption* of her name and the transaction data (such as date, shop, and order ID). This message is encrypted for a trusted third party which regulates in the case of a dispute. Alice proves to the winery that the message is indeed encrypted for this trusted third party and contains the same name as in her passport as well as the correct order information. Along with this, Alice also sends a data handling policy that declares under which circumstances the encrypted information may be revealed. (This policy is usually proposed by the shop first and Alice accepts it hereby.)

For shipping and payment there are of course several alternatives in the system design. In our case, for instance, using credit cards for payment allows (at least) the credit processor to see that Alice bought goods of a certain amount from the winery. Alternatives would be e-Cash systems that do not allow tracking of the money spent [4]. A similar problem appears on the shipment side as an address must be provided here. For privacy, Alice may decide to have the goods sent to a post office near her and fetch them there using a code.

Scene 2: Credential Federation. The online winery of the previous scene is a member of an association of online stores that have a privacy-friendly frequent customer service (FCS). Using this FCS, one does not need to show several credentials (such as a passport for an age check and a credit card for payment) at each store for each purchase. Rather, Alice first anonymously registers at the FCS and obtains a credential from the FCS for shopping at any associated online store. Alice proves to the FCS that she owns a Swiss passport according to which she is over 18. Next, she proves that she owns a credit card issued to the same name as in the passport without revealing this information. Further, she also proves that the expiry date of her credit card is at least one year in the future (without revealing the expiry date). The FCS issues a credential with a unique customer number that certifies that Alice is over 18 years old and that she owns a (non-expired) credit card. This credential has an expiry date of its own that is set to one year from the current date—thus it is guaranteed to expire before the credit card does.

It is even possible to include information such as Alice's name, credit card number and expiry date in the FCS credential without the FCS learning these values. This allows that Alice later proves properties related to this credential, e.g. when putting her name into a verifiable encryption. To that end, one can use

techniques similar to *blind signatures* [9]. Here, Alice transmits to the FCS some values (e.g. her name) in a blinded form (such that the FCS cannot see the true values). Next, the FCS creates a credential with the blinded values. Alice can now unblind the credential such that it contains all values in clear and that it still has a valid signature from the FCS. Note that there are several differences to existing blind signature schemes. First, the blinding applies only to certain attributes rather than to the content as a whole. Second, Alice also proves in zero knowledge that the blinded values she sends correspond (modulo blinding) to values in credentials she owns.

The FCS credential can now be used as if it were both a passport and a credit card at all shops that accept the credentials from the FCS (i.e. that trust statements from the FCS). As described in the requirements below, multiple uses of the same credential cannot be linked even when the FCS and all the shops cooperate and share their information.³

Observe that using the passport in scene 1, the winery could necessarily see that Alice is a Swiss citizen as the Swiss government is the issuer of her passport. The FCS credential does either not contain this information at all, or it is an attribute that is not necessarily revealed when showing the credential. The federation of credentials as it was performed here can thus help to make the system more privacy-friendly.

Scene 3: Anonymous Bonus System. Our online winery also wants to offer a little bonus to the returning customer worth a certain percentage of all purchases. Once more this shall be implemented in a privacy-friendly way. In a conventional store, such a bonus system can be implemented by a paper card which has, say, 20 fields for stamping; for every 20€ worth of purchases, the customer gets one field stamped; once all fields are stamped, the customer gets 20€ off its next purchase in exchange for the stamped card.

Similarly, the online wine shop additionally issues special credentials to Alice when she makes a purchase. Each of these credentials represents one stamp from the off-line world, i.e. 20€ in the example. Note that the credentials must not contain the exact amount of the purchase, as otherwise such a credential can easily be linked to a particular purchase. Rather, each credential represents a fixed amount (such as 20€ in the example).

Once Alice has collected enough “electronic stamps” she can use them in the next purchase to get the bonus; to that end she must prove the possession of 20 such credentials. The requirements in this case (as described below) are that the winery cannot link the stamps to particular previous purchases and that Alice can

³Note however that the number of users who obtain an FCS credential may be quite small (e.g. when only Alice uses this service), giving the online shops a better chance to link actions.

“spend” every stamp-credential only once, as the name one-show credential suggests.

3.3.2 Security and trust requirements

1. **Correct presentation of credentials.** When a user convinces a server that it possesses a credential with particular properties, then this user indeed possesses credentials with these properties. This entails that also a dishonest server cannot “forward” such a proof to another party, impersonating the user. It further includes that one cannot forge credentials that the apparent issuer has never issued. Similarly, also proofs about verifiable encryptions shall be correct.
2. **Privacy/Unlinkability.** When a user proves the possession of a credential to a server, then the server does not learn more about the credential than those facts that the user deliberately chose to prove. More generally speaking, the user maintains the control over the information it reveals and to whom.

Further, dishonest servers cannot tell whether particular actions have been performed by the same or by different users. This holds even when several dishonest servers collaborate. For instance, consider two dishonest servers S_1 and S_2 that collaborate and where S_1 has issued a credential for some user U (acting under some pseudonym) with a property P (e.g. that the owner is over 18). U (acting under a different pseudonym) proves to S_2 that it possesses a credential from S_1 with property P . We require that, even combining the knowledge of S_1 and S_2 , each of the certificates with property P that S_1 has issued is equally likely to be the one U has shown to S_2 .

3. **Accountability.** In case of criminal behavior or violation of regulations, it shall be possible to revoke the anonymity of the user performing the transaction in question. This can only be done under well-defined circumstances by a particular trusted authority. Such circumstances could be the violation of regulations that the user had accepted, or a court order in case of criminal investigations.
4. **Blind Signatures.** A user can transmit attributes of a credential in a blinded way to the issuer of a credential and later unblind the obtained credential. The result should be the same as if the respected values were transmitted without blinding. Similar to the correctness requirement (1), we further require that the user can only prove that the blinded value corresponds to a particular attribute in a particular credential if the correspondence actually holds.

5. **One-Show Credentials.** While normal credentials can be used any number of times, a one-show credential can only be spent once.⁴

Note that every party of this system may potentially be dishonest (i.e., in general no party needs to trust any other one) with one exception: for accountability, we need a trusted third party that must be honest (since it has the power to revoke the anonymity of users); in particular, it is part of the requirements that even several dishonest collaborating entities (e.g. the issuer and the verifier of a credential) cannot break the anonymity.

On the other hand, there are trust relationships between issuing and verifying organizations. For instance, all scenes silently assume that all OECD governments are trustworthy for issuing a passport, while some other countries' governments may not be trusted, and thus are not accepted for this. Similarly, in the frequent customer scene, the wine shop needs to trust the FCS to give out credentials only after being shown an OECD passport and a valid credit card. While these trust relationships are important in the design of such a system, they are not directly related to the requirements of the credential system. It is a requirement, for instance, that one cannot forge credentials (that appear as being issued by an organization that never did), but it is not a requirement that a dishonest issuer may only sign correct statements.

Solution: Identity Mixer. A system that is designed to achieve all these requirements is the Identity Mixer anonymous credential system of IBM [5, 7, 2, 6] that we will consider in the following. The Identity Mixer system is based on non-interactive zero-knowledge proofs [3]. This means that one party P (the prover) shows to another party V (the verifier) that it knows a secret with a particular property without revealing the secret. In particular, this is used to “show” credentials: for instance, a user proves to a server that (s)he owns a passport with the property to be over 18 years old, but without transmitting a copy of the passport or even just revealing the precise date of birth.

A subtle problem arises out of the combination of correctness and privacy requirements: we want to ensure that only the legitimate owner of a credential can use it (in particular preventing proof forwarding), but privacy dictates that the true identity of the owner is not revealed and moreover the owner should be able to act under different pseudonyms in each transaction in order to prevent linkability. The solution of Identity Mixer to this problem is that every user has a secret value that is never revealed, called the *master secret*. All pseudonyms and credentials are based on this master secret in the sense that it is input to the construction of

⁴One may relax this requirement to not *preventing* but just *detecting* double-spending [5]; thanks to accountability, one can then reveal the misbehaving user.

pseudonyms and credentials. Whenever a user wants to show a credential, (s)he must also prove to know the master secret behind the credential — again in zero knowledge, i.e. without revealing the master secret.

Zero-knowledge proofs are also used to verify the already mentioned verifiable encryptions. Namely for accountability we need an escrow of core information (such as a user's real name) encrypted for a trusted third party. The zero-knowledge proof convinces the verifier that the encrypted text indeed contains the required information and is encrypted for said trusted third party — without revealing the encrypted information.

3.3.3 Federation

Credential Federation. Scene 2 requires the federation of anonymous credentials: organizations can issue new credentials based on other credentials that the user has shown. These newly issued credentials can contain blinded attributes, i.e. ones that the issuer has not seen in clear. This refers to the requirements of privacy (2) and blind signatures (4).

3.3.4 Accountability

Privacy-Friendly Logging. This problem case addresses the requirement (3) to prevent the abuse of privacy for criminal actions. Again, the privacy requirement demands that we implement accountability, i.e. “logging transactions”, in a privacy-friendly way using verifiable encryptions for a trusted third party.

3.3.5 Privacy

Anonymous Credential System. This problem case addresses the core requirements of this scenario. The first, on which AVANTSSAR focuses, is the main requirement that any credential system must satisfy (even those that are not designed to protect the users' privacy): correctness (1). The second core requirement is privacy (2). The combination of these two requirements can only be fulfilled using advanced techniques such as the ones deployed in Identity Mixer.

One-Show Credentials. For credentials that represent a value to be spent only once, as used in scene 3, we want to prevent (or detect) double-spending as an additional requirement (5).

3.4 Citizen and Service portals

E-Government (from electronic government) refers to the use of Internet technology as a platform for exchanging information, providing governmental services and transacting with citizens and businesses, but also between different arms of government. One delivery model is government-to-citizen (G2C), which is considered primarily in this document. Example services in the area of G2C range from very security-sensitive, e.g., e-voting to more security-insensitive ones like a bulletin board for the exchange of ideas between citizens and government.

Other models are government-to-business (G2B) comprising services like e-procurement of public authorities and electronic filing of trademark right requests and government-to-government (G2G) for services between different arms of government or even between governments of different countries. G2G is not considered in this deliverable.

Within these interaction domains, typically, four kinds of activities take place:

- pushing information over the Internet, e.g.: regulatory services, general holidays, public hearing schedules, etc.
- two-way communications between a government agency and a citizen, a business, or another government agency. In this model, users can engage in dialog with agencies and post problems, comments, or requests.
- conducting transactions, e.g.: lodging tax returns, applying for services and grants.
- governance, e.g.: on-line polling, voting, and campaigning.

The most important anticipated benefits of e-government include improved efficiency, convenience, and better accessibility of public services.

The development and implementation of e-government involves consideration of its effects on the organization of the public sector and on the nature of the services provided by the state including environmental, social, cultural, educational, and consumer issues, among others.

An indispensable prerequisite for the success of e-government is trust, where trust mainly refers to security and privacy. If citizens and businesses do not trust the e-government services they will not make use of them.

In general and in particular for purposes of e-government, but also for e-health or e-commerce, today's Internet does not offer appropriate (simple and cost-effective) authentication procedures and security services, except for very limited and restricted environments or applications. Furthermore, electronic transactions over the Web do not have the legal binding, certainty, and liability that have been common for surface mail or, more generally, other types of paperwork.

In order to overcome the existing problems, several countries of the European Union, and in particular Germany and Austria, announced to create and maintain certified Citizen Portals that will support a secure communication interface within the Internet. With them, every citizen will have a secure, personalized access point in Internet, enabling communication with government offices, e-health providers, and other service providers in a straight-forward, easy, and yet secure way. From this portal citizens may access a great variety of services with different authentication, authorization, and protection requirements. The portal is responsible for providing the corresponding security mechanisms and protecting the privacy of the citizen. In particular, the person-related data (more precisely: Personally Identifiable Information, PII) must be adequately protected and the informational self-determination ensured. In other words, every individual has the right to decide what information about her or him may be communicated to the individual service providers and under what circumstances.

Similar portals, i.e., service portals, are expected to be provided for businesses, to support the specific public services they need to interface with. In the European Union, the European Directive on services in the Internal Market plays an important role in the area of G2B. This Directive sets out an ambitious program of administrative simplification and modernization. It aims to break down barriers to cross border trade in services between EU Member States, making it easier for service providers, particularly small and medium sized enterprises, to set up business and offer services in other Member States and to provide services temporarily and/or at a distance in other Member States. Probably the most striking achievement and challenge of the Services Directive is the introduction of the so-called Points of Single Contact: Member States shall ensure that it is possible for service providers to complete all procedures and formalities through points of single contact in the administration instead of having to deal with multiple authorities, which can be quite burdensome especially for foreign services providers. Additionally, the Directive requires to make information on national requirements and procedures easily accessible. The Services Directive Member States have made binding commitments to deploy e-government applications by the end of 2009.

The Services Directive will lead to services related Internet portals which serve as intermediaries in a way that a service provider's activity will not be broken down into partial components. The portal as single point of contact must provide comprehensive coverage of all relevant administrative procedures.

In the following section a couple of different scenes will be sketched in different level of detail where Mike is using the citizen portal and a service portal to carry out some recurring or non-recurring striking tasks most people know from daily life.

3.4.1 Scenario definition

Mike is a middle-aged single, living in a smaller suburb of a medium-sized town in Germany close to the French frontier. He is a hard-working employee of a medium sized French enterprise running 10 optical stores spread over France, and he often feels that in his private life he is running out of time.

Scene 1: Electronic Mail – Secure and Legally Binding. Mike wants to change the provider for his mobile telephone service. In the past he would have had to write a letter on his computer, and then to print it. He then had to drive to the post office in next local center, a place where parking is difficult, in order to have the letter registered. The whole procedure would have wasted two hours of his precious time.

Mike remembers that he has signed in at the citizen portal of his town for a new countrywide mail service. This service offers the possibility to send and receive electronic letters in a secure way (protection of integrity and confidentiality), and furthermore legally binding.

Mike visits the home page of the mobile network provider and he is pleased that the provider supports the new way of electronic mailing. The provider even offers an appropriate form, being downloaded by Mike. He fills in the form and stores it on his computer. He then accesses his citizen portal and logs into the new browser based e-mail service. He chooses the service “certified mail” and sends the form to the mobile network provider. Within a few seconds he obtains confirmations on his mail: one from the citizen portal, the other one from the network provider.

The whole procedure took only about 15 minutes of time. If everything during his past relation as a customer of the network provider had worked as the described act of service cancellation, Mike surely would have preferred to remain a customer.

Scene 2: Annual Tax Declaration. Mike opens his mailbox at his citizen portal. He finds an e-mail from his tax authority. He opens it, although he knows in advance what the content of this mail is. Mike is reminded to hand in his tax declaration for the last year.

Mike uses the service of the citizen portal to log directly into the server of the tax authority. Thus, no additional authentication is needed. The tax service automatically shows Mike’s tax number, and it offers a selection of forms suitable for Mike’s needs. He starts to fill in the forms. Some time later his brother rings up and reminds him of an appointment with some friends. Mike stores the forms on the tax server and decides to continue the work at a later time.

The next day, Mike is on a short business trip. Unfortunately, the meeting took much more time than expected, and Mike misses his plane in the late afternoon. Thus, he has to wait for two hours at the airport for the next plane. He decides to use the dead time for continuing his tax declaration. He books a WLAN based Internet connection and connects with his notebook to his citizen portal. The communication channel offered by the portal provides end-to-end security. Mike reopens the tax forms and starts to work. Although there was plenty of time, he is not able to bring the document into its final form, as he misses a few supporting documents. So a little work remains which is to be done at home again.

The next evening Mike does the final work on his tax declaration. He signs the document electronically and adds any receipts, vouchers etc. which are available in electronic form. He is aware of the fact that the tax authority may ask for further supporting documents which are only given as hard copies. It will take years until this process will be available in pure on-line form, but the on-line tax service provided by the citizen portal is a major improvement, anyway.

Scene 3: Street Party. Mike meets a couple of neighbors. During their conversation they come to the conclusion that it would be nice to have a street party with all neighbors around. They decide to ask around if there exist objections, and if a street party would meet the interest of the majority. Mike is asked to take care for the administrative decision by the municipality.

Mike remembers that in the past this job has not been easy, as the following public offices and institutions are involved:

- The office for urban planning and traffic, as public streets are to be used and will be blocked during the party
- The police and the fire department which support the office for urban planning and traffic
- The office for restaurants and public houses, as food and drinks are to be offered
- The office for waste management
- The office for public order, as music is planned to be played

The last time, when the neighbors had planned a similar event, they had to run from office to office in order to get the final decision.

Mike logs into his citizen portal and finds a button “services for citizens”, leading to another one “events”. Here, several standard events like street processions, sporting events, markets, as well as street parties are listed. Mike chooses

street parties. He is offered a form where he can enter information on his event, date, location description, further information on music, catering etc. He learns that his application will have to run through a first check, and that he will receive further demands automatically, depending on the decisions of the various offices which are involved. These demands will be sent to him using the secure e-mail provided by the portal. All communication will run through the citizen portal acting as kind of trustee. In order to achieve data austerity, the involved offices will only get a minimum of personal identifiable information (if it is needed at all) and other application-related information.

Mike is glad to have this new way of event application, because the whole procedure is much easier than in the past, while achieving a higher degree of data protection.

Scene 4: Taking Benefit of Business Opportunities. The next day, back in his office again, Mike is asked to take care of the company's future business opportunities in Germany. As a first step it is planned to open a new office in Munich.

It only happened recently that Mike obtained his European eID card which has been endowed with attributes which enable Mike to act as a company representative. Mike logs into the German e-government portal. Mike is recognized a representative of his company, and it is checked that he has the rights to open a new business in Munich.

Mike fills in an application form and signs it with his eID card. A few minutes later Mike receives a signed e-mail from the German portal which acknowledges the receipt of the application. He is informed that he will get further information on the next steps and on the possible time schedule within the next seven days. The first step towards a new business has been done.

In the background, the (simplified) example application process depicted in [Figure 4](#) is executed, showing the involved authorities (and therefore the need for a single point of contact):

- Registration at the respective, local registry office

As the first step the applicant has to be registered at the respective, local registry office, which will send the certified document of the applicant to the immigration office requesting his freedom of movement document. Since the applicant is a craftsman, the District Government is asked if the applicant owns a master craftsman's certificate or has an equivalent standard (master craftsman's certificate is not known in all Member States). After that, the craftsman has to be registered at the Chamber of Trade.

- Registration at Chamber of Trade

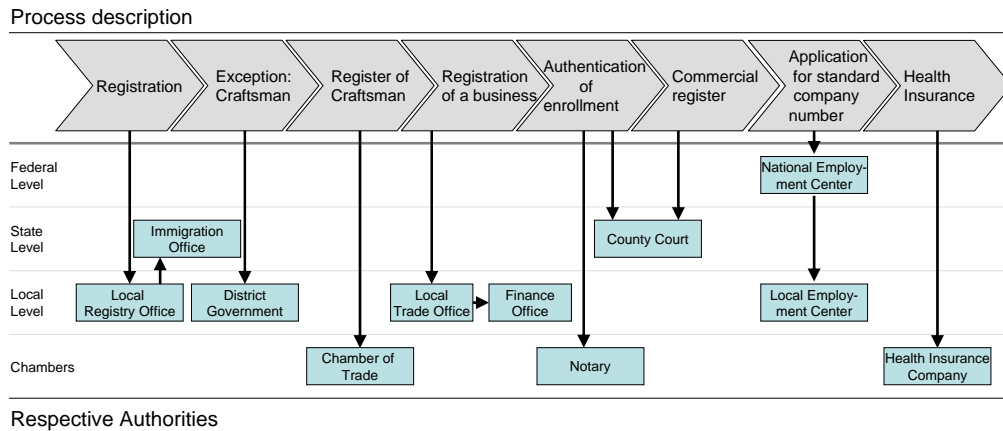


Figure 4: Example Application Process of a French Optician in Germany

The applicant must be registered at the respective chamber of trade in order to offer his service in Germany

- Registration of the business at the respective, local Trade office. The local Trade office will contact three more authorities:
 - Notification of Finance Office to certificate a tax number for the applicant
 - Notification of Employers Mutual Insurance Association in order to get workers compensation insurance in Germany for the applicant (not shown in the Figure for reason of clarity)
 - Certification of national insurance pass (not shown in the Figure for reason of clarity)
- Authentication of enrollment claim by county court or notary

The applicant needs to authenticate his enrollment by the county court or an independent certified notary
- Commercial register by county court

The business of the service provider has to be enrolled in the national commercial register
- Application for standard company number at National Employment Center

Every Service provider needs a standard company number for his business in Germany. The National Employment Center delegates to the respective Local Employment Center

- Declaration of employees regarding health insurance

With the beginning of the service offering, all employees need to have a health insurance

Aside from the presented authorities, there are some infrastructure services needed to be able to execute the task:

- Forms Management: The single point of contact provides forms which have to be filled in with all necessary data of the demanded service.
- Document Safe: The document safe is a functionality to securely upload and download relevant documents necessary in communication with the public authority.
- Identity Management: Identity Management is necessary to handle the access of different users and services.

Scene 5: Car Registration. The next weekend, Mike is looking for a new car. After consultation of several car dealers, he chooses a two-year old mid-size car and pays using his credit card. Of course, before he is allowed to drive the car, he has to register his new car at the car registration office, which – of course – is closed during weekend. Thus, again Mike takes advantage of the citizens portal where he can lodge a car registration anytime.

Before describing the workflow which has to be initiated to accomplish this task, the actors and roles of the scene – apart from Mike, who obviously is the customer of the car dealer and the one who wants to register his new car – are introduced.

CentrRep is a central repository where car registration documents are stored among other things. Additionally, empty forms for several purposes are stored and available to everyone. The central repository is not purely passive: it checks digital signatures of documents and authorization of users retrieving and storing documents. The documents themselves are not inspected by the central repository but treated as black box.

CarRegOffice is the car registration office where requests for car-registration have to be submitted. There might be several CarRegOffices, e.g., one per district. All CarRegOffices use the same CentrRep to access and store documents, therefore CarRegOffice in this example stands for the office which is in charge for Mike.

RegOffEmpl is an employee of the car registration office. He is processing car registration requests. Employees of the car registration office are allowed to

- read documents from the local repository,
- add any number of comments to them and
- store documents in the local repository.

Additionally, a RegOffEmpl is allowed to store fully processed car registration requests to CentrRep if his or her RegOffHead legitimates him/her to do so. In our example the RegOffEmpl Peter is processing Mike's request.

RegOffHead is the head of the car registration office. The head is responsible for leading a car registration office and is the only person which is primarily permitted to write documents to the central repository. Of course, s/he can delegate this privilege to trustworthy employees to support him/her in his/her hard work. Please note, that RegOffHead is always also a RegOffEmpl. Melinda is the head of the CarRegOffice in charge for Mike.

RegOffCA is the certificate authority of the car registration office. Its task is to generate trustworthy certificates on employees. A "certificate" in this context is an authentic assertion, such that the consumer of the assertion is sure that the author of the assertion is the "signer", in this case the RegOffCA.

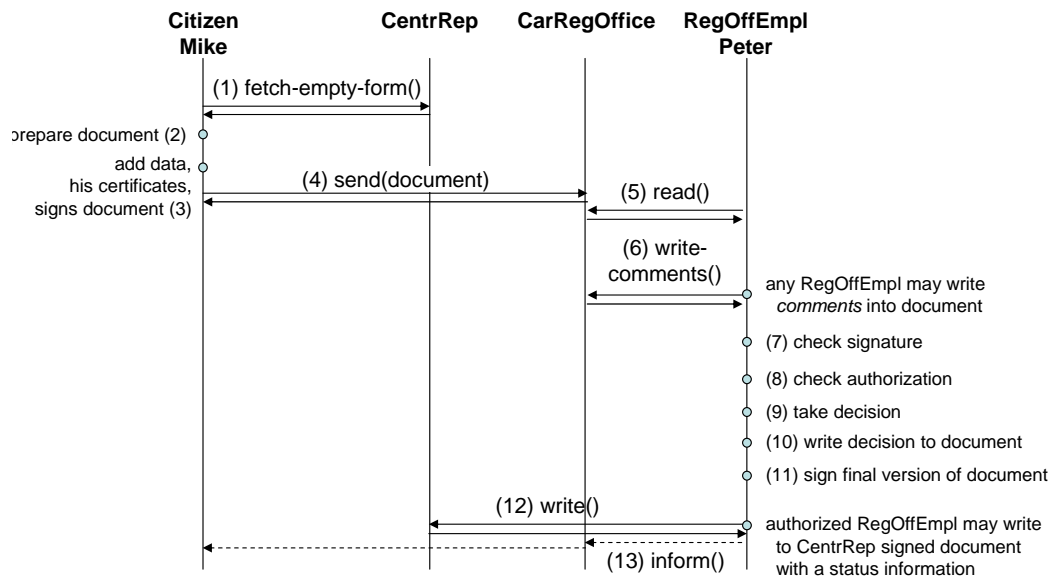


Figure 5: The Car Registration Process

The car registration process is shown in Figure 5. Of course, Mike does not know about this process running in the background. He logs into the citizens portal, navigates to the car registration office and initiates the car registration process

by fetching an empty form (1). He fills in his PII and data about his new car (2). After that, he adds any required and optional certificates to the form, e.g. a proof that he possesses a valid car insurance, a certificate that the car is roadworthy, etc (3). After that, he signs the document and sends it to the CarRegOffice (4). Thus, the document is stored in the local repository of the car registration office.

On Monday, Peter fetches Mike's document, reads it (5) and adds some comments to the document (6). These steps may be repeated by any number of RegOffEmpl. After that, Peter performs the following actions in an atomic way:

7. checking Mike's signature,
8. checking the authorization of Mike, based on the certificates provided by Mike (e.g. car insurance, etc),
9. taking a decision,
10. writing the decision in the same document,
11. signing the final version of the document, and
12. writing it into CentrRep. Note that RegOffEmpl needs special permissions for this step.
13. Finally, the RegOffEmpl informs the CarRegOffice that he has completed his task.

Thereupon the CarRegOffice informs Mike about the decision taken upon his request. With this last action, the car registration process terminates.

The system constraints of this scene are given in [Figure 6](#):

- Peter holds three certificates:
 - The RegOffCA confirms, that Peter is a RegOffEmpl and
 - that Melinda is RegOffHead.
 - Melinda permits Peter to write documents to CentrRep.
- The access control list (ACL) of CentrRep states, that
 - Anybody can get empty forms
 - RegOffEmpl can read documents from the repository, but can only write documents if his/her RegOffHead permits it.
 - RegOffHead can write documents.

The local (trust) policy states that RegOffCA can say

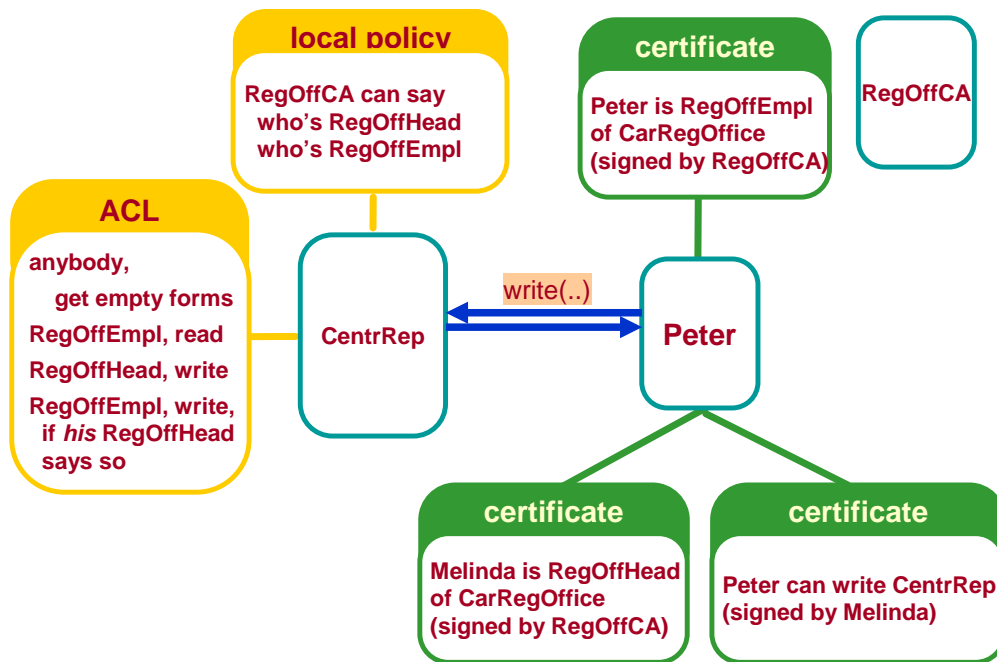


Figure 6: Peter Storing a Document in CentrRep

- who is RegOffHead and
- who is RegOffEmpl.

With these system constraints in mind, it is easy to see that Peter actually is authorized to process Mike's request and is also authorized to write the final document back to the CentrRep.

3.4.2 Security and trust requirements

Complying with the uniform structure described in [subsection 1.2](#), we are now specifying the security and trust requirements of a citizen or service portal focusing on the presented scenes. Of course, the statements about privacy and PII made in the context of banking services in [subsection 3.1](#) are valid in the context of citizen portals as well.

Overall requirements.

1. The operator of the portal, i.e., the government or a representative, has to verify the PII of each citizen registering to the citizen portal, if an explicit registration to the portal is necessary. In this case, the representative of the

portal must not store the citizen's PII anywhere else than in the citizen's information file. Additionally, a citizen must authorize the portal and its representatives to handle her/his PII. This implies that a representative must not disclose PII to anyone without the citizen's explicit consent.

2. If every citizen has access to the citizen portal without explicit registration, e.g., by data stored on the customer's identity card, government must ensure that only the rightful owner of an identity card can access the stored information or use it to log into the service portal.
3. Information files must be managed within a governmental computing system, which must comply with international security evaluation standards like Common Criteria [8].
4. No-one shall access or modify an information file without the authorization of the owning citizen, who must give permissions according to the needs of services he wants to use.
5. Any form of citizen's authorization (see 3.4.4) must be provided by the citizen's signature, typically in digital form.
6. The portal operator or its representatives must not be able to act on behalf of a citizen (i.e., use provided services) without the citizen's authorization.
7. In order to achieve data austerity, services must be designed in a way to limit the amount of personal data necessary to a strict minimum. This applies not only to PII, but also to other private information. For example, race or political opinions are not of interest for the approval of a street party application.
8. Using the portal, a citizen should be able to securely log into other related services, possibly belonging to different security domains, with the same credentials as for the portal itself (Single Sign-On).

Electronic Mail – Secure and Legally Binding.

9. Information exchange between a citizen and any recipient must ensure integrity, confidentiality and mutual authentication on the demand of sender or receiver.
10. Non-repudiation of origin and non-repudiation of receipt must be ensured if demanded by sender or receiver.

11. The receiver of a mail must not use data contained in a mail in any way that is not associated with fulfilling the requested service. For example, the mail address, PII or dynamic private data of the sender must not be sold, used for spam or used for advertising if the citizen does not give explicit permission.

Annual Tax Declaration.

12. No-one except the owning citizen or his/her tax counselor shall access or modify partially filled in forms stored on the tax server. A citizen must explicitly declare his/her tax counselor as authorized delegate.
13. No-one must be able to modify the finalized tax declaration without the citizen's authorization.
14. Information exchange between a citizen and the tax authority must ensure integrity, confidentiality and mutual authentication.
15. Both non-repudiation of origin and non-repudiation of receipt must be ensured. Also, non-repudiation of content must be ensured. For example, a citizen must not be able to fill a deficit field and claim having filled a benefit field at a later date.
16. The tax authority must not seek or maintain any private information except the citizen's PII and facts crucial to tax declaration, e.g., annual income.
17. Only the representative of the tax authority who has to audit a specific tax declaration, his boss and possibly controllers must be able to access a tax declaration and the corresponding tax audit result.
18. No-one must be able to modify a tax audit result which is sent back to a citizen in response to his/her tax declaration.
19. The decision on a tax declaration must be made within three months.

Street Party.

20. No-one must be able to modify the street party application without the citizen's authorization.
21. Information exchange between a citizen, the portal and all involved offices must ensure integrity, confidentiality and mutual authentication.
22. Non-repudiation of origin and non-repudiation of receipt of the street party application must be ensured.

23. In order to achieve data austerity, the involved public offices and institutions must only be able to access a minimum of information and the street party application must contain as few information as possible.
24. Only the representatives of the involved offices which have to audit a specific street party application, their bosses and possibly some controllers must be able to access a street party application and the corresponding resulting permission or refusal.
25. No-one must be able to modify a street party permission or refusal which is sent back to a citizen in response to his/her application.
26. The decision on an application must be made within six weeks.

Taking Benefit of Business Opportunities.

In the following, only security requirements that are new to a scene are enumerated. Most of the requirements listed in the scenes presented up to now can be adjusted to the subsequent scenes.

27. Only employees who are accredited by their companies as company representatives with the necessary privileges must be able to open a new office.

Car Registration.

28. Documents must be secret for anyone who can not read CentrRep.
29. Final documents stored in the CentrRep must be consistent, i.e., signatures are correct, etc.
30. The authentication and trust policies must be adhered to.

Some of the requirements listed above are non-technical and thus intangible to formal methods, others are out-of-scope of the AVANTSSAR project. In detail these requirements are:

- Requirement 1 is partly a non-technical requirement: It is not possible to technically prevent the representative of the portal from noting the customer's PII anywhere else but in the customer's Information File once the representative has the information in his mind. A possible organizational solution is to put a dedicated instruction into the representative's contract of employment defining the terms of PII-handling the representative must adhere to.

A very similar problem is raised by requirement 11 which demands restrictions on secondary use of data. This problem must be addressed on the organizational layer as well, e.g., by appropriate contracts.

- Requirements 3 is as well a non-technical requirement that has to be handled on the organizational layer. Nevertheless, a Common Criteria certification might improve the trust of the customer on the portal.

Only security requirements have been stated so far. The corresponding trust requirements can be deduced from the security requirements (cf. [subsection 3.1.2](#)).

Complying with the uniform structure described in [subsection 1.2](#), we are now enumerating the problem cases which are produced by the various security and trust requirements, grouped into several families.

3.4.3 Federation

In general, the 'federation' of identity, credentials, or attributes enables the portability of such information across otherwise autonomous security domains. The ultimate goal of identity federation is to enable users of one domain to securely access data or systems of another domain seamlessly, and without the need for completely redundant user administration. Federation comes in many flavors, including user-controlled, enterprise controlled, and B2B scenarios. Typical use-cases involve things such as cross-domain, Web-based single sign-on, cross-domain user account provisioning, cross-domain entitlement management and cross-domain user attribute exchange.

Single Sign-On. A fundamental problem when using diverse protected applications and resources belonging to diverse security domains is that users have to log in for every single application and/or resource they want to use by providing adequate credentials, e.g., username and password. This often leads to the situation that the same username/password-pair is used for several applications and resources, which is not desirable from the security point of view. Another problem is the time spent re-entering passwords for the same identity.

To overcome this problem, Single Sign-On (SSO) techniques are used. SSO is a method of access control that enables a user to log in once and gain access to many applications and resources of different systems without further authentication. As different applications and resources support different authentication mechanisms, a single sign-on system has to internally translate and store different credentials, compared to what is used for initial authentication. SSO uses one or more centralized authentication servers that all other applications and systems

utilize for authentication purposes, and combines this with techniques to ensure that users do not actively have to enter their credentials more than once.

Requirement 8 raises the problem case of Single Sign On for Web portals.

3.4.4 Authorization Policies

Access Control. Analogous to banking services (cf. [subsection 3.1.3](#)), the security requirements of citizen and service portals raise problem cases, that can be addressed by authorization policies. Because the problem cases here are in the same family as those of [subsection 3.1.3](#) and almost all of them can be solved in a similar way, we merely list the corresponding security requirements: 1, 2, 4, 5, 6, 12, 13, 17, 20, 24, 27, 28, and 30.

The requirements 12, 27, and 30 must be pointed out, because they add a new problem case to the authorization-family of problem cases, i.e., delegation.

Delegation. In Scene 5, the RegOffHead can delegate the permission to write to the central repository to trustworthy employees. The word “trustworthy” could mean that an employee has signed a contract defining all rights and liabilities bound to this permission. A hint to a technical solution is the usage of certificates, as already demonstrated in the car registration scene.

3.4.5 Accountability

Non-Repudiation. Analogous to banking services (cf. [subsection 3.1.4](#)), some security requirements of citizen and service portals raise the problem case of non-repudiation. Because of the similarity of the problem case, we merely list the corresponding security requirements: 10, 15, and 22.

3.4.6 Trust Management

General Trust Management. As already stated in [subsection 3.1.5](#), even though the requirements 1, 3, and 11 are non-technical or out-of-scope - as already mentioned - trust management techniques can be used as an instrument to provide confidence for the customer, that these requirements are fulfilled.

Trust Policies. Analogous to the already presented application scenarios, the security requirement 30 of citizen and service portals raises a problem case that can be addressed by trust policies. The problem case here is in the same family as those already presented in [subsection 3.2.5](#) and can be solved in a similar way.

3.4.7 Workflow Security

Flow Control. Security requirement 29 of the citizen and service portal scenes is a flow requirement, as already introduced in [subsection 3.1.6](#).⁵ If the presented work flow is adhered to, the inner signature of a citizen is always verified by a RegOffEmpl before it is written to the CentRep and thus, the documents in CentRep are always consistent.

Timeliness. A new class of workflow security requirement introduced by the presented scenario are timeliness requirements, see requirements 19 and 26. These requirements are necessary to assure fair processing of applications. Without such requirements, e.g., a car registration application might be delayed for a very long time, e.g. because a RegOffEmployee dislikes the applicant, which could be sensed as denial of service.

3.4.8 Privacy

Security requirements 7, 16, and 23 of citizen and service portals raise the problem case of data austerity.

Data Austerity. Requirement 7 splits up into two different problems which most easily can be demonstrated using Scene 3 as an example. The initial form presented to the citizen collects all data necessary for all further steps. In order to achieve data austerity, this form must be designed in a way that only data is collected that is absolutely necessary for at least one authority involved in the approval process. After that, the form is given to several different authorities. An adequate authorization policy (see [subsection 3.4.4](#)) is needed, such that every authority can only access data contained in the form that is absolutely necessary for the authority to accomplish its task.

3.4.9 Application Data Protection

Data Integrity. Requirements 13, 18, 20, 25, and 29 raise the problem case of data integrity, that already has been introduced in the banking services scenario (cf. [subsection 3.1.8](#)) and can be solved in a similar way.

⁵Workflows executed internally by the different authorities involved in the processing of a request to a citizen or service portal may require flow requirements and separation-of-duty requirements as well, but these workflows are out of the scope of this deliverable.

3.4.10 Communication Security

Requirements 9, 14, and 21 raise the problem case family of communication security. This problem case family was already introduced in the banking services scenario (cf. [subsubsection 3.1.9](#)).

Message Confidentiality. No-one except the intended recipient of a message must be able to read the content of the message.

Message Integrity. No-one must be able to modify the contents of a message unnoticed.

Peer Authentication. A principal is assured of the identity of its communication partner.

3.5 Document Exchange Procedures

On the Internet and within the enterprise, there is a growing demand for dematerialized procedures, that is to say for online electronic services that may replace paper-based business and administrative procedures. One critical concern for such service portals is guaranteeing the legal value of the produced electronic documents such as business contracts or public service forms. The European Commission and European countries have foreseen this need by producing directives and national regulations for giving a legal base to digitally signed documents. Thus, it is possible nowadays to build secured document exchange Web Service platforms which produce legally binding documents by means of digital signatures. The security requirements imposed on such platforms are highly critical since the probative value of digitally signed documents relies on the conditions under which they have been produced and validated.

To illustrate this security aspect, this section describes and analyzes examples of signed document exchange procedures based on Web Service infrastructure.

3.5.1 Scenario definition

Scene 1 - Business Digital Contract Signing. In this example, two parties (employer/employee, supplier/vendor, ...) have secure access to a trusted third party Web site, a business portal, in order to digitally sign a contract.

First, the business portal application generates an electronic document corresponding to the terms of agreement between the two parties. Then, the first party accesses the business portal using a Web browser, views the contract and signs it using a digital certificate. The business portal verifies the generated signature and stores it.

The second party, in turn, connects to the Web site, checks the status of the existing signature and then co-signs the contract after viewing it. Once the signatures have been completely verified by the business portal, the signers are notified. Then, the contract is archived for long-term conservation.

The business portal's internal system is Web service enabled. It delegates the processing of proof elements (signatures, signed documents, timestamps) to a *Signature and Proof management Infrastructure platform* (SPI platform) using SOAP messages.

To provide its services, the SPI platform relies on the following trusted third parties:

- PKI: Public Key Infrastructure issuing signing certificates and maintaining certificate revocation lists.
- Timestamping: timestamping server synchronized with a reliable time source.

- Archiver: proprietary archiving provider. It guarantees long-term safekeeping for proof elements.

An instance of the SPI platform potentially supports interfacing with several business portals running simultaneously. In this case, the deployment architecture is illustrated by Figure 7. Each business portal is associated to a specific account and cannot access other business portals' data.

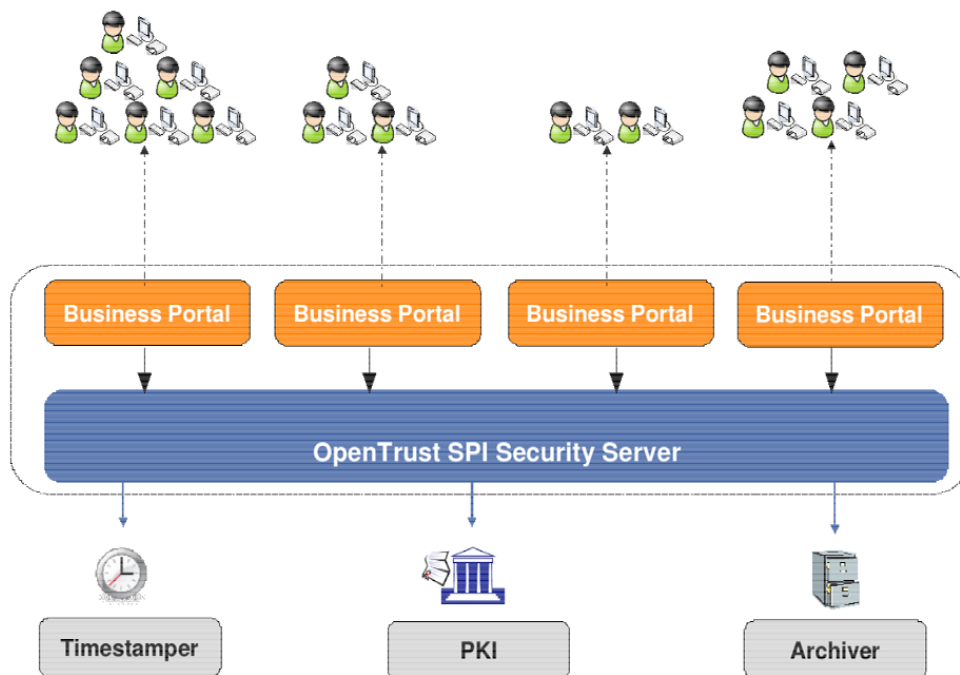


Figure 7: Architecture of a digital contract signing system featuring Opentrust SPI platform

The section below describes the interactions between the actors involved in a typical contract signing scenario. We first introduce these actors:

- PPP is a company which manages a web portal where temporary work agencies and temporary work employers meet to sign contracts of employment. PPP has chosen to run *Opentrust SPI Security Server* as SPI platform for all signature and proof-related tasks necessary in their business workflow. We will call this platform *Security Server* in the scenario.
- HHH is another company, which would like to hire a personal assistant for a 3-month contract.

- WWW is a temporary work agency which happens to be able to provide the personal assistant that HHH needs.

The first steps of the workflow are shown in [Figure 8](#).

1. We assume at this point that the contract has already been generated by the business portal application. An agreement has been found between HHH and WWW, and they have provided enough information for the business portal to create the employment contract. The scenario starts with the creation of a digital case, called proof record under the Security Server's context. It will later gather the contract, the signatures and any complementary proof elements used for their validation (such as the Certificate Revocation Lists). It is stored on the Security Server's file system.
2. A copy of the contract is transferred to the Security Server. It is deposited into the newly created record.
3. This step is reached when HHH (aka signer1 in the diagram since it is the first signer of the contract) requests the contract from the business portal as well as all the rules and parameters applicable for generating the signature. These rules are defined in a signature policy managed by the SPI Security Server. They are retrieved at step 4.
4. The business portal requests the contract signature preparation from the SPI Security Server. This preparation consists in extracting a set of parameters from the signature policy. They indicate the mandatory properties that must be embedded in the signature (such as the signing time, the signature policy identifier, etc.).
5. First, HHH views the contract and makes sure that it corresponds to the terms agreed upon. Then, HHH selects his signing certificate and uses the associated private key to sign the contract and the mandatory properties. After this step, signature1 is created.
6. signature1 is submitted to the business portal.
7. The business portal forwards the signature to the Security Server in order to add it to the corresponding proof record.
8. This is a critical step in the record's life-cycle. Before adding any signature to the record, the SPI Security Server immediately verifies the following:
 - The signing certificate's validity period has not elapsed.
 - The signing certificate is issued by a trusted certificate authority.

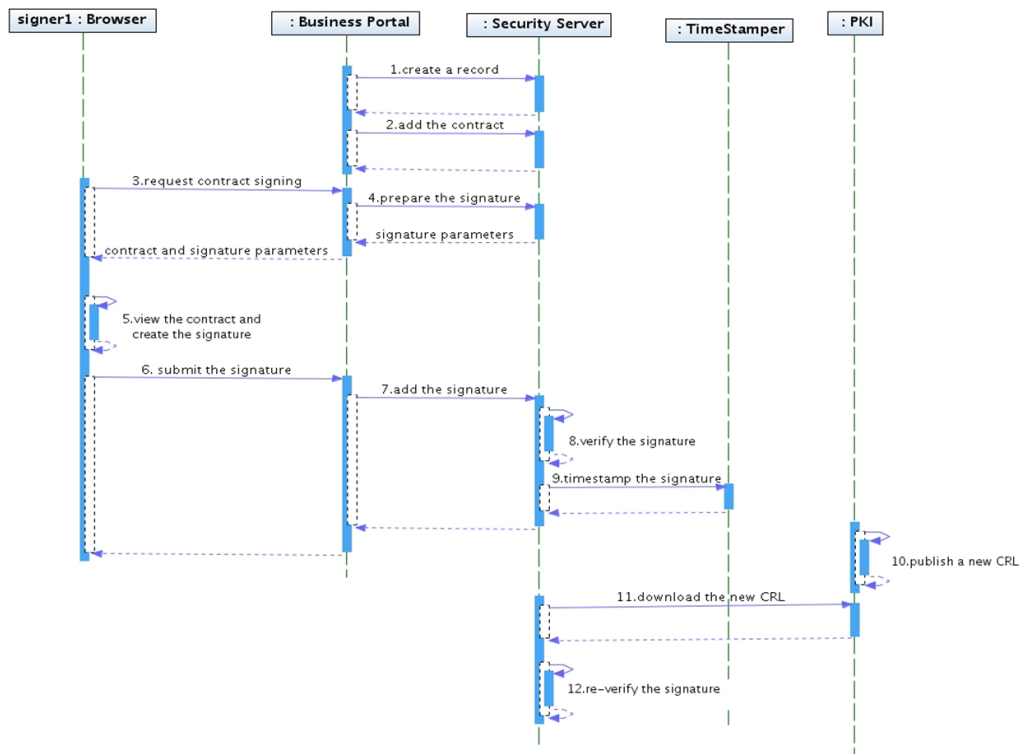


Figure 8: Contract signing sequence diagram, part 1: HHH signature

- The signing certificate is not revoked.
- The signature is syntactically and cryptographically valid and the signed information corresponds to the contract.
- The signature respects all the rules defined by the corresponding signature policy.

Some verifications cannot be performed immediately. For example, certificate revocation lists are published periodically by the PKI. As a certificate revocation status cannot be reported before the date of CRL publishing, the Security Server must wait for the next CRL to be published before completing the validation of the signature. This control is postponed to step 11.

9. The Security Server sends a timestamping request to the timestamper to seal the signature's reception date.
10. The PKI publishes an up-to-date CRL.

11. The Security Server has an internal CRL cache which is refreshed periodically. At this step, the Security Server downloads the new CRL, verifies it and then loads it into the cache.
12. The Security Server starts a background verification of the signatures in pending status. With the new CRL, the Security Server is able to confirm that the certificate of signer1 was not revoked when signature1 was created.

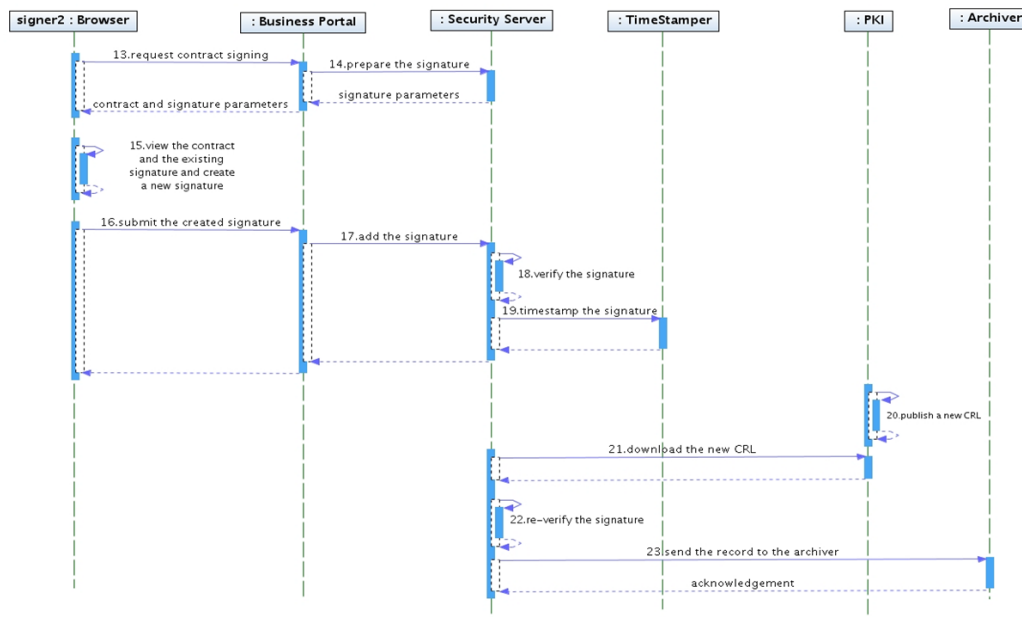


Figure 9: Contract signing sequence diagram, part 2: WWW signature

13 to 22 As illustrated in [Figure 9](#), WWW (signer2) similarly creates a signature of the contract. The Security Server performs the same processing described above.

23 The proof record contains fully verified signatures. The Security Server sends it to the archiver which guarantees it can be restored in case it is removed from the Security Server's file system intentionally or accidentally.

Scene 2 - Public Bidding. The public bidding example illustrates a secure document exchange application used for bid management in the context of public procurement. The goal is to provide a Web application platform offering online services for electronic publication of public calls for tender and bidders' proposal submissions.

Such a platform can be operated by national government entities or by local public entities as well in the context of new e-government services. Compared to the traditional paper public bidding process, an electronic bidding system will raise efficiency in terms of human cost and document processing time and protection. Moreover, in case of an Internet public bidding system, it will increase the *visibility* and *transparency* of public procurements, especially for small and medium-sized enterprises which can more easily access the available tenders and participate in the bidding.

A typical public bidding system would provide the following services:

- Call for tender publication performed by public entity representatives
- On-line subscription of potential bidders
- Tender time-line management (beginning date and end date of each sub-process of the public bidding)
- On-line submission of bidders' proposals
- Retrieval of submissions for public entities representatives
- Publication of the decision of the public entities for tender awarding.

A fundamental requirement of such a system is to comply with applicable legislations and regulations of public procurement related to the protection of exchanged documents and the compliance to the time-line of the bidding process. To ensure equity, bidders' proposals must usually remain confidential until the end of submission date. It is also critical to ensure document integrity during the whole decision process so that no one, including the bid managers or platform IT operators, can tamper with submitted documents.

Let PBS be a *Public Bidding System* that offers a web-based public bidding service. Such a service can be technically viewed as a secure document exchange platform for managing the life cycle of calls for tender. On one side, bidders access the PBS to consult available calls for tender and to submit their offers. On the other side, different persons in charge of conducting the bidding process, access the PBS to publish the calls for tender, consult the submitted offers and validate the workflow steps.

The public bidding Business Process (BP) relies on digital signatures among other security mechanisms for securing the process. It has the following specific interests compared to the digital contract signing BP:

- The public bidding business process makes a special focus on confidentiality issues as documents submitted by bidders must be reviewed only by the right persons and for the right part.

- This scenario introduces time-related security requirements. The PBS must guarantee that some activities cannot be performed after a deadline (e.g. new bidding submission) whereas some others can only be performed until a defined date (e.g. tender openings).

The principals of the public bidding scenario are divided into 2 groups: the PBS users and the PBS itself. The PBS users are the end users that will remotely access the PBS through a web browser for performing tasks according to their role in the public bidding BP.

- *Bid Manager (BM)*: the bid manager is the person in charge of conducting one specific call for tender. He is responsible for ensuring the smooth execution of the whole bidding process with respect to legal constraints. He is in charge of the contacts with the bidders and of making the final decision about the selected contractor.
- *Project Committee (PC)*: the project committee members (represented by the committee chairman) are in charge of the technical evaluation of the bidders' offers. The financial aspects of the offers are not known to the project committee. The committee must designate the most suited offer based on its technical content only. The committee evaluation report is used by the Bid Manager for awarding the market based on the best technical offer with regards to the best price.
- *Bidder (B)*: each bidder can look at the available calls for tender and download them on the PBS. After that, they submit their offer with respect to the bidding process rules on document confidentiality and bidding time-line. They connect to the PBS during the evaluation process to check whether their application has been declared acceptable, and at the end of the bidding process to know if they are awarded the contract.

The PBS infrastructure is composed of the following service components:

- *Bidding Portal Application (BPA)*: The BPA implements the bidding process and manages all interactions with the bidding BP actors through its web portal. It is responsible for authenticating portal users. It delegates all server side management of digital proof elements (signatures, validation, signed documents, encryption and decryption, timestamping, ...) to a dedicated security service provider, the Security Server.
- *Security Server (SS)*: The Security Server is an infrastructure that is independent from business applications. It provides security services based on digital signature and data encryption. Business applications call the Security Server interface for document signing, signature validation, document

encryption and decryption. The Security Server relies on trusted third parties for performing specific operations:

- PKI: Public Key Infrastructure which issues certificates used for signatures and publishes certificate revocation lists.
- Time Stamp Authority: timestamping server synchronized with a reliable time source. The TSA delivers a signed timestamp on any data digest submitted by service clients.
- Archiver: proprietary archiving provider. It guarantees long-term safe-keeping for proof elements.

The following paragraphs describe the tasks that compose the public bidding BP. We assume that all users are authenticated by the BPA by means of individual cryptographic tokens (e.g. smart card containing a user authentication certificate and associated private key) prior to any operation on the portal. For newcomers (e.g. potential bidders), we assume that they perform a subscription process (not described here) in order to obtain an authentication certificate from the BPA.

Publication sub-process. The bidding process starts with the call for tender publication step which includes the security context preparation (encryption keys creation). The publication process tasks are as follows:

1. The BM locally generates an encryption key pair (or he may reuse one that he has previously registered). He uploads the public part $k_p(BM)$ to the BPA as the key to be used by potential bidders for ciphering submitted documents.
2. The BPA automatically generates a receipt form stating the registration of the encryption key. It calls the Security Server for signing the receipt with a server signing certificate on behalf of the PBS.
3. Security Server calls the timestamp server for timestamping the receipt.
4. The signed and timestamped receipt is returned by the BPA to the Bid Manager.
5. The Project Committee represented by its chairman also generates an encryption key pair and uploads the public key $k_p(PC)$ to the BPA. This key will be used to encrypt bidders' documents that are to be read by the PC only (technical proposals).
6. A signed and timestamped receipt is returned to the PC chairman.

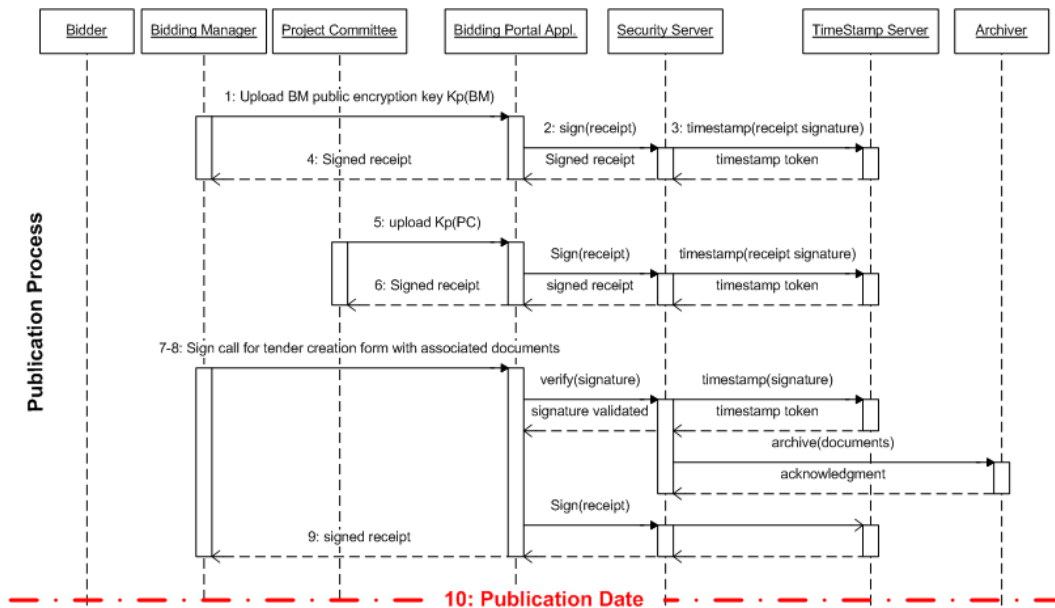


Figure 10: Basic Public Bidding BP - Publication sub-process

7. The BM creates a new call for tender by filling the creation form on the BPA web site. He indicates the timetable of the bidding process. He joins the call for tender documents. The BPA makes him sign the creation form along with the attached documents.
8. The BPA submits the signed creation form and call for tender documents to the Security Server for validating the BM's signature and for initiating a new proof record with this data. From this point, a copy of all submitted signed documents related to the bidding will be kept by the Security Server and put into the associated proof record.
9. The BPA generates a signed receipt to be returned to the BM to acknowledge the registration of the new public bidding request.
10. When the publication date is reached, the BPA makes the call for tender visible to the public.

Submission sub-process. During the submission process, potential bidders check out the call for tender documents and produce their proposals. In order to submit their offers, each bidder performs the following tasks:

1. The Bidder B_i fills in the application form, ciphers it with $k_p(BM)$ provided by the BPA and signs it with its private key $k_s(B_i)$. The resulting signed

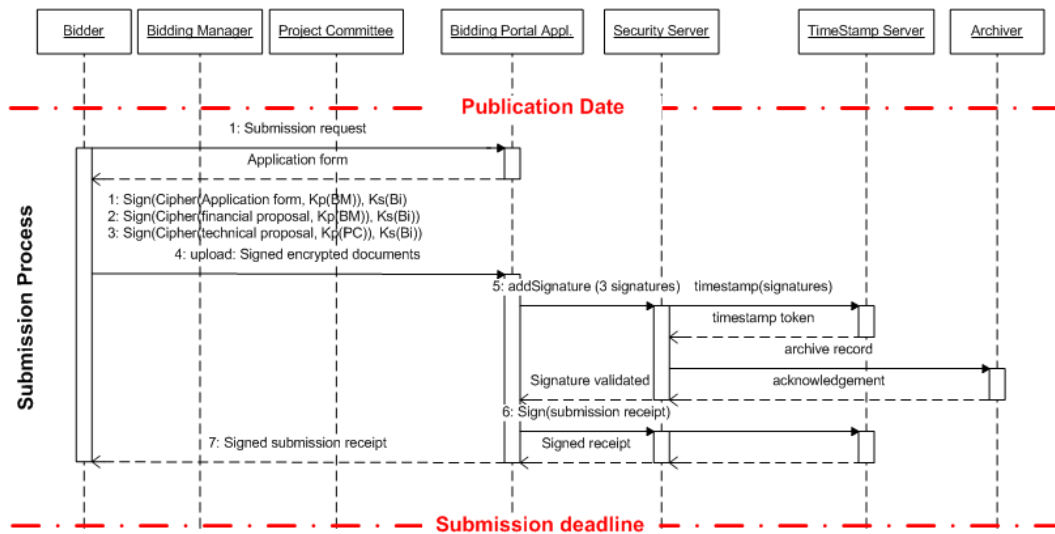


Figure 11: Basic Public Bidding BP - Submission sub-process

document is uploaded to the BPA:

$$\text{sign}(\text{cipher}(\text{application form}, k_p(BM)), k_s(B_i))$$

2. B_i performs the same task for the financial proposal. The resulting signed document is uploaded to the BPA:

$$\text{sign}(\text{cipher}(\text{financial proposal}, k_p(BM)), k_s(B_i))$$

3. B_i uses $k_p(PC)$ instead for encrypting the technical proposal to be reviewed by the PC only. The resulting signed document is uploaded to the BPA:

$$\text{sign}(\text{cipher}(\text{technical proposal}, k_p(PC)), k_s(B_i))$$

4. The BPA submits all 3 signed encrypted documents with their signatures to the Security Server.
5. The Security Server validates bidder signatures on encrypted documents, generates signature timestamps and sends a copy of the data to the Archiver.
6. The BPA generates a submission receipt document stating the deposit of the application form and associated proposals. The BPA calls the Security Server for signing and timestamping the receipt.
7. The submission receipt is returned by the BPA to the bidder B_i , it serves as proof of deposit stating that the submission has been done before the submission deadline as defined by the BM.

6. The PC chairman retrieves all submitted technical proposals belonging to the eligible candidate list.
7. The PC chairman decrypts the technical proposals and reviews them (with all PC members).
8. At the end of the document review, the PC chairman fills in the evaluation form for each proposal indicating its compliance, and if so, gives a score according to the quality of the offer. The evaluation form is signed by the PC Chairman.
9. For each signed evaluation form, the BPA transmits it to the Security Server for validation and generates a signed receipt to be returned to the PC Chairman.

Decision sub-process. This is a straightforward sub-process aimed at confirming the contract awarding.

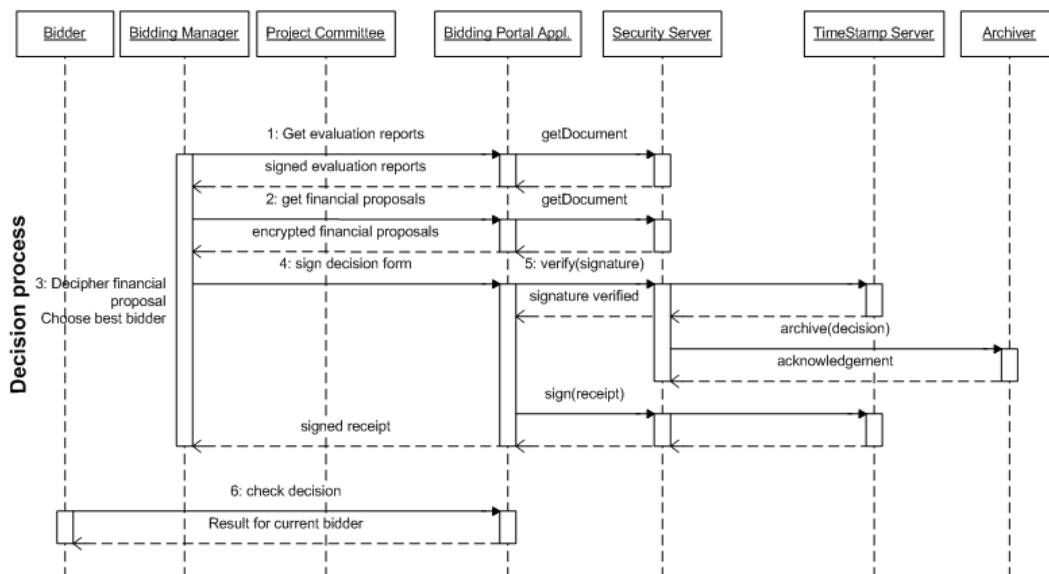


Figure 13: Basic Public Bidding BP - Decision sub-process

1. The BM retrieves all evaluation reports from the BPA.
2. Offers stated as non-compliant by the PC are rejected. For the others, the BM retrieves the encrypted financial proposals.

3. The BM decrypts the financial proposals and calculates the final score for each bidder with regard to the price indicated in the financial proposal. ⁶
4. The BM chooses the best bidder based on its technical and financial scores, fills in the decision form on the BPA and signs it.
5. The BPA transmits the signed decision form to Security Server for validation and generates a decision receipt to be returned to the BM.
6. Each Bidder checks the result of the bidding process on the BPA. It can see if its offer is rejected and the identity of the winning bidder.

Scene 3 - Extended Public Bidding. The extended public bidding example is a variant of the previous public bidding example. The extended variant introduces some new constraints, and consequently, new tasks and mechanisms to address those constraints:

- *Tender upload grace period:* it is a common situation for bidders to submit their application at the very last moment. This may lead to technical issues if the number of bidders is significant and/or if there are huge amounts of documents to upload. To fulfill legal requirements on the fact that all bidders must be able to register their submission up to the last moment, the PBS can implement a pre-deposit option associated with a grace period for uploading documents.
- *Unbiased third-party control:* we have previously assumed that the BM is a fully trusted principal. In the extended variant, this assumption is loosened. A new role is added, called Bailiff (BL), who acts as an unbiased party which can enforce the control for date constraints. This is achieved by an additional encryption with the BL key to ensure the confidentiality of submitted documents until the evaluation step.

The public bidding BP sub-processes are modified as described below (we omit the Archiver role to simplify the process description).

Publication sub-process. The extended scenario for the publication sub-process contains the following specific tasks compared to the basic variant:

⁶We assume that the financial evaluation is a pure calculation and that the 'subjective' part of the scoring relies on the technical evaluation. This is only an example, different scoring strategies are of course possible in real life.

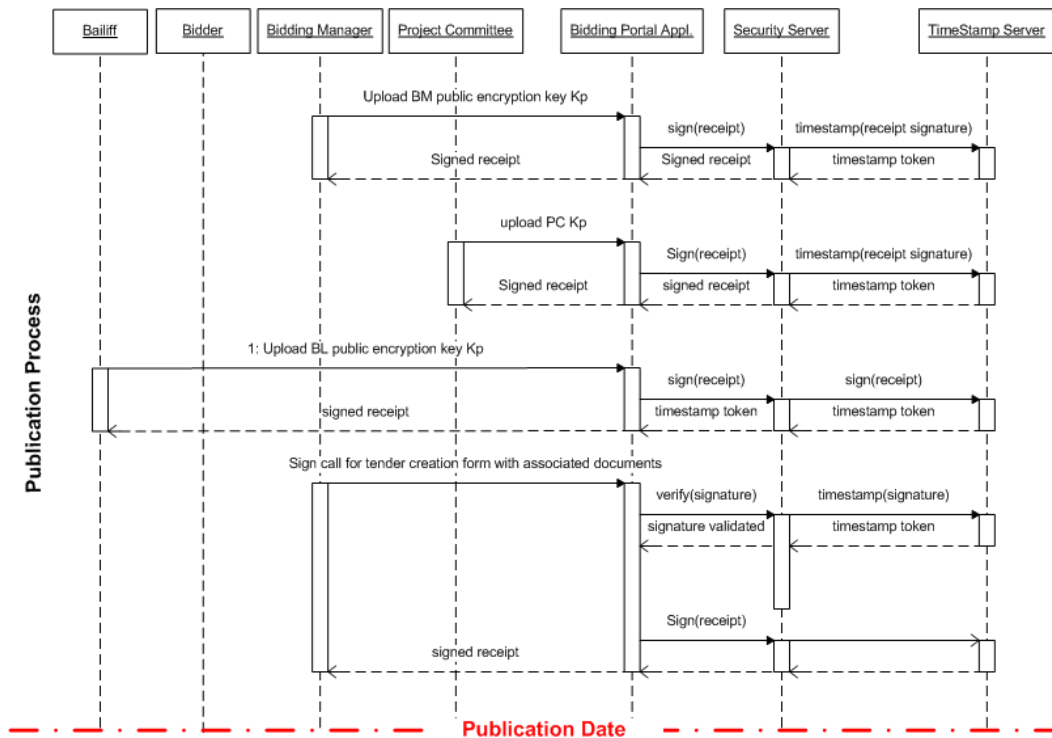


Figure 14: Extended Public Bidding BP - Publication sub-process

1. The BL locally generates a new encryption key pair to be used specifically for the designated bidding process (since the secret key is revealed during the BP, this key pair is not re-usable). He uploads the public part $k_p(BL)$ to the BPA as the key to be used by potential bidders for double ciphering of submitted documents.

Submission sub-process. The extended scenario for the submission sub-process contains the following specific tasks compared to the basic variant:

1. The Bidder B_i fills the application form, successively ciphers it with $k_p(BM)$ and $k_p(BL)$, both provided by the BPA, and signs it with its private key $k_s(B_i)$. The resulting signed document is uploaded to the BPA. There is no problem for uploading the application form at this step since the application form is usually a small statement. Even if there are many bidders connected to the BPA, the application form file upload overhead is not significant and should not prevent the bidder from completing the submission transaction in a reasonable delay.

$$sign(cipher(cipher(application\ form, k_p(BM)), k_p(BL)), k_s(B_i))$$

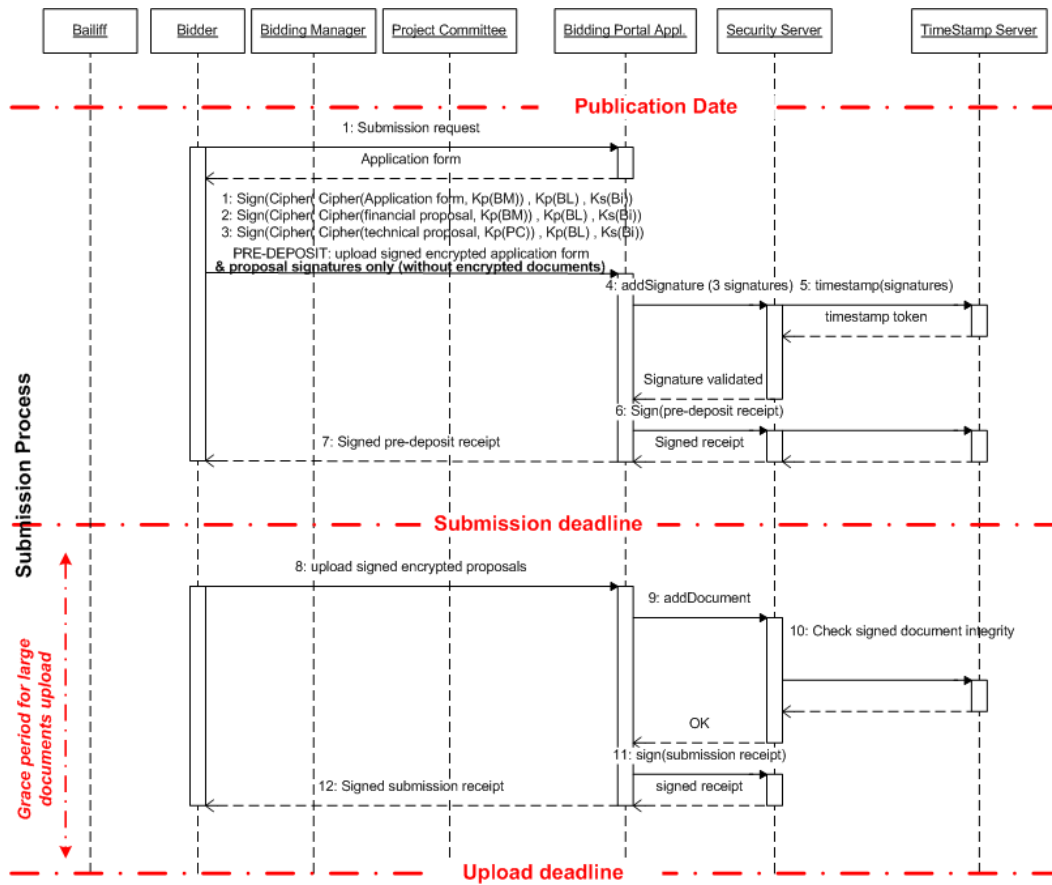


Figure 15: Extended Public Bidding BP - Submission sub-process

2. B_i performs the same task for the financial proposal. But he only uploads the digital signature block extract only without the document itself:
 $extract(sign(cipher(cipher(financial\ proposal, k_p(BM)), k_p(BL)), k_s(B_i)))$
3. B_i uses $k_p(PC)$ for encrypting the technical proposal to be reviewed by the PC only. He uploads the resulting signature block extract only without the document itself:
 $extract(sign(cipher(cipher(technical\ proposal, k_p(PC)), k_p(BL)), k_s(B_i)))$
4. The BPA submits the uploaded data with their signature to the Security Server.
5. The Security Server validates bidder signatures on encrypted documents. For the proposals, the Security Server validates the integrity of the signature and the validity of the signing certificate but not the integrity of the signed

document as it is not uploaded. The Security Server generates signature timestamps and sends a copy of the data to the Archiver.

6. The BPA generates a pre-deposit receipt document stating the deposit of the application form and proposal signatures. The BPA calls the Security Server for signing and timestamping the receipt.
7. The pre-deposit receipt is returned by the BPA to the bidder B_i , it serves as proof of deposit stating that the pre-submission has been done before the submission deadline as defined by the BM.
8. At any time later, whether the submission deadline is over or not, B_i can upload the previously encrypted proposals to the BPA until the upload end date. This is the grace period that runs from the submission end date to the upload end date.
9. The BPA transmits the encrypted documents to the Security Server.
10. The Security Server checks the integrity of the documents based on previously submitted signatures.
11. The BPA then generates the final submission receipt and calls the Security Server for signing and timestamping the receipt.
12. The final submission receipt is returned by the BPA to the bidder B_i .

Evaluation sub-process. The extended scenario for the evaluation sub-process contains the following specific tasks compared to the basic variant:

1. When the submission date and upload date are officially over, the Bailiff (BL) uploads its private key $k_s(BL)$ to the BPA so that the BM and PC can retrieve it and decrypt the submitted proposals.

3.5.2 Security and trust requirements

Common requirements. The previous document exchange procedure scenes have a common set of basic security requirements since they share the same service components architecture. These requirements apply to the global platform made of the front end (the portal in all cases) and the back ends (the different mutualized services).

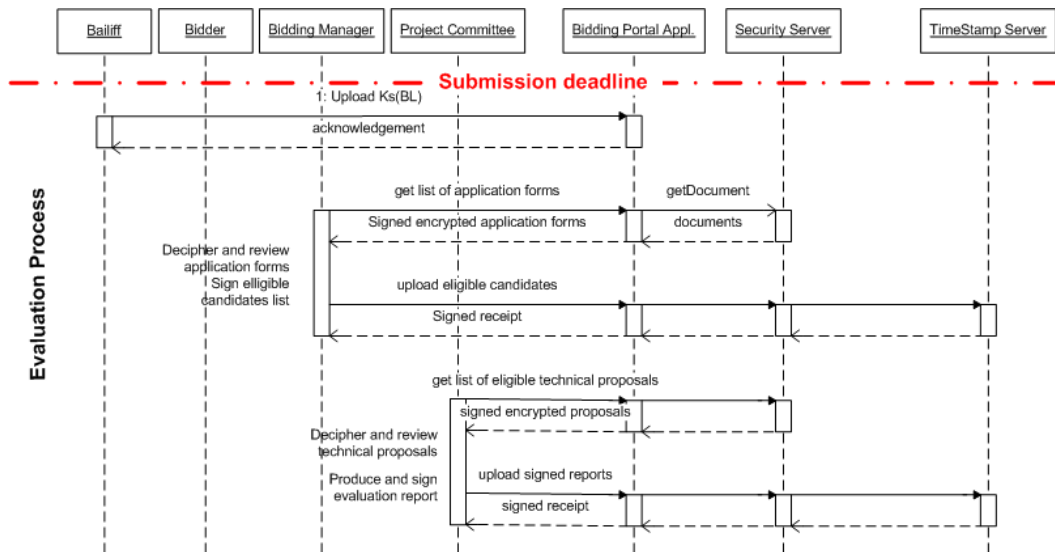


Figure 16: Extended Public Bidding BP - Evaluation sub-process

Authentication. There are several security relationships between the human principals and the service components which require authenticated communications. In the particular examples of document exchange platform described here, each service must authenticate the peers it depends on:

1. All end users must authenticate themselves when connecting to the business portal.
2. All back end service components (BPA, Security Server, Timestamp server, archiver) must authenticate themselves to each other because we made the assumption that there are shared components which must identify their service caller in order to apply specific security restrictions (see the authorization requirements below).
3. In our scenes, the back end services (the Security Server, timestamp server, etc.) do not need and should not need to know the identity of the end-users. They provide service to the business portal and their security policies are based on what one particular business portal instance is entitled to do.
4. Signers must be able to authenticate the origin of the data they received (application form to sign, signed submission receipt, signed declaration of the bid winner, etc).

Authorization.

5. The business portal is in charge of enforcing end user authorization control based on business workflow specifics (see individual security requirements for each scene).
6. Considering the fact that there are several business portals accessing the service of the Security Server, records created and managed on behalf of a given portal must not be accessed by other portals. Likewise, records co-signed by two parties must not be accessed by another party.

Integrity.

7. The Security Server must be able to detect any modification of the signatures or the signed information.
8. The Security Server must ensure that the document that has been signed and uploaded by the user is the same as the one generated by the portal and submitted to the user for signature. This can be achieved by making the Security Server sign the document right after the document generation and before submission to the user for signature. This requirement applies to server side generated document, not document provided by user.
9. The Security Server must not be able to forge false documents, that is to say, the IT administrator who has root access to the Security Server should not be able to modify documents submitted by end-users and make the Security Server re-sign the modified documents. This requirement leads to have the document signed by the different parties (user-side signature along with server-side signature) so that one party alone cannot change the document once it is viewed and validated by all actors.

Traceability.

10. Signatures and timestamps provide relevant information when re-enacting a scenario for security or business reasons. Additionally, the Security Server must also keep audit trails supporting real-time or post supervision.

Timeliness.

11. The Security Server uses timestamps to associate reliable dates to the signatures. The Security Server must be able to verify the freshness of timestamping responses.

Non-repudiation.

12. Non-repudiation of origin and content: Signers, in a legal dispute, must not be able to refute the validity of their signatures or the content of the signed contract. The Security Server must have enough proof elements to verify a signature even in a far future.
13. Signing certificate validity: the signer certificates must be valid and not revoked at the time the signature is claimed to have been generated. Thus, the Security Server must check the revocation status of any signing certificate when validating the signature with regards to the claimed signing time (time explicitly indicated in the signature or by the portal)

Specific security requirements for business digital contract signing Here is a list of security and trust requirements specific to the contract signing scene:

14. Integrity: Signer1 and signer2 must not be able to modify the presented contract before signing. This leads to the requirement that the business portal must check that the uploaded signed contract is the same as the initial contract submitted to signer1 for reviewing and signing. This requirement also applies to subsequent signatures.
15. Proof of origin: The contract is submitted to Signer2 only when the Security Server has fully validated Signer1 signature (including Signer1 certificate revocation status) and ascertained Signer1 is a valid first signer in the workflow.
16. Timeliness: the timestamping delay (delay between signature submission and signature timestamp generation by the Security Server) should be sufficiently small (less than a few seconds) so that the signed timestamp can be considered later as an approximation of the signature time.
17. If the signature/co-signature of the contract must be performed within a limited time span, the Security Server must be able to download CRLs and to fully verify the signatures before a specified expiration date. Otherwise, the contract is null and void.
18. Non-repudiation: Signer2 signature must be fully validated by the Security Server before archiving the signed contract.
19. Trust policies: Signer1 and Signer2 certificates must be issued by a trusted certification authority.

Specific security requirements for public bidding

Authorization constraints.

20. During the submission process, no one except the Bid Manager or his representatives can access any information related to the identity of the bidders.
21. A bidder can only view the decision related to himself and to the bid winner (not for any other bidders).
22. No user can have more than one role within the same bidding process.

Fairness constraints.

23. The BM must not omit a tender when uploading the list of eligible tenders for Committee evaluation. Any tender is either eligible or not, but the status has to be declared by the BM on the BPA for all submitted tenders, otherwise the committee evaluation process should not start.
24. The Committee must not omit to evaluate an eligible tender. The list of evaluated tenders must match the eligible list submitted by the BM.
25. The bidders must all have a proof of receipt of their tenders. This allows them to have fair defense in the case of a conflict.

Integrity constraints.

26. Since one bid is made of several documents (application form, financial proposal, technical proposal), it is important that the system enforces the integrity of the entire bid, that is to say, no-one can remove, substitute nor add documents once the bid is submitted by the bidder.

Security requirements for extended public bidding The security and trust requirements specific to this extended scene are:

27. BL's encryption key must not be known to the PBS or any other party except the BL itself until the end of the submission deadline. Once the deadline is over, the key is made public.
28. The BL's certificate must be recognized as a trusted third party by the bidders. Thus it must not be issued by the same CA as the certificates given to the other users by the PBS in case this CA is operated by the same party as the one operating the PBS.

29. The integrity of the documents submitted during the upload grace period must be verified by the PBS and must match the previously submitted signatures.

3.5.3 Authorization Policies

Access control. This problem case deals with sensitive user data protection. As a shared infrastructure, potentially used by different e-business or e-government portals, the Security Server must enforce access rights to contracts or tenders. This is usually done by means of access controls based on authorization policies.

For instance, in the advanced public bidding scene, the BM is not fully trusted for guaranteeing that the tenders are not opened before the end of the submission process. He may collaborate with the IT operator for retrieving the encrypted proposals when the submission process is still pending. The BM and PC must not be able to view the proposals until the BL gives it authorization through the publication of its secret key. Thus, a role based access control is applied to bidding process flow management and to user documents.

This problem case is related to the common security requirements: 5 and 6. It is particularly crucial for the Public Bidding requirements 20, 21, 22 and 27.

3.5.4 Accountability

Non-repudiation. There are three sub-problems related to non-repudiation:

- The first concerns the proof of origin: principals have to verify the authenticity of the documents as originated from the real authors. For example bidders need to verify the authenticity of the downloaded call for tender documents as originated from the public entities.
- The second concerns the content: authors should not disclaim their responsibility for the data they submitted. For example, public entities need to compel the bidders to provide the service described in their submitted proposals once the bidding decision process has started.
- Finally, the third is proof of receipt. For public bidding, bidders may require getting a proof of submission so that the BM cannot remove arbitrarily tenders and deny having received them.

This problem case focuses on the techniques used to prevent repudiation in conformity with requirements: 4, 12, 13 and 18.

3.5.5 Trust Management

Digital contract signing implies the agreement of parties on terms that usually have critical impacts. This family is concerned with problems that deal with trust policies applied at the moment they electronically sign their agreement. As they make use of digital signatures, the same problems apply to the public bidding scenes.

Trust policies. The document exchange platform is designed to support trust establishment via TTPs. This problem case focuses on the mechanisms implementing it as specified by the requirements: 19 and 28.

3.5.6 Workflow Security

Timeliness. A fundamental issue about documents exchange workflow security is enforcing of the timeline of the different sub-processes. In the context of public bidding, requirements addressing this issue (11, 16 and 17) ensure that beginning date and end date of each step of the workflow are to be checked by the PBS to authorize actions specific to the corresponding steps in such a way that it cannot be challenged by any party (especially a losing bidder) to invalidate the bidding process. Since bidders are able to upload their proposal after the submission deadline, the upload grace period must not be used as an additional delay for completing the proposals. So the BPS must ensure that the uploaded documents are the ones signed at the pre-deposit step.

Traceability. This problem case deals with issues related to requirement 10.

3.5.7 Application Data Protection

Data Integrity. Another typical digital signature problem is integrity: undetected alteration of the signed data must not be possible. In the public bidding context, document integrity is to be enforced by the whole PBS in such a way that it cannot be challenged by any party to invalidate the bidding process. Requirements concerned by this problem case are: 7, 8, 9, 14, 15, 29.

3.5.8 Communication Security

Peer Authentication. In this problem case, we focus on the authentication of communication between the involved peers/components as specified by 1, 2 and 3.

3.6 Personal Health Information in the Hospital

Over the past years there have been several proposals for standardizing format, exchange protocols, and access control to electronic personal health information (PHI), but there is still no global consensus to be expected in the near future. In Germany, for instance, there is a large controversy about the pros and cons of different types of electronic repositories for health data. The *mobile Patient-EHR file* model proposes that the patient himself will keep and will be able to carry with him, in a properly backed-up secure portable repository, like a smart card, all his Personal Health Information. In a similar vein, the *web-based Patient-EHR file* proposal (Elektronische Patientenakte), being part of the future German Telematics Infrastructure, suggests that the user should keep his medical data in the Internet, in a page that he can presumably control, see [13]. The German project eFA [12], a project run by six big German clinics and four big hospital operators, advocates the *Electronic Case Record (eCR)*, a possibly distributed, logical “file” owned and managed by one hospital or a network of hospitals. The eCR replicates more closely the way hospitals manage Personal Health Information today. The hospitals create, collect, and manage electronic health records (EHRs) for the purposes of treating a particular case.

It is indeed possible that several of those models will co-exist in the future. The family doctor or the patient himself may have a Patient-EHR file, while the hospitals will create Electronic Case Records that will be eventually merged into a Patient-EHR file after the patient is released from the hospital, and deleted from the hospital when the Personal Health Information is not needed anymore in the hospital.

The paradigm where there is *only* one Patient EHR file, in any of its forms, has the disadvantage that the doctor or clinic will have to search through all information, to find the relevant parts for the case he is treating. In the Electronic Case Record, the doctor will ask for specific EHRs to other hospitals or the family doctor of the patient. There are also privacy concerns with this paradigm as it will be very difficult to grant access to EHRs via the “need to know” principle: The patient can give consent to a doctor to access the EHR as a whole, but it is very difficult for the patient to give him consent to access only a subset or a view of his EHRs. It is even more difficult if not impossible to decide which parts of the EHR file should the doctor be able to access. Patients who wish to exercise control over access to their data seldom understand the implications of their decisions. It is much easier for the doctor to ask if the patient has had, say, a certain type of examination, and if yes, ask consent to the patient to access it. This process is much more transparent to the patient.

Since there is not even a coherent terminology in the different countries and projects, it will not be surprising that there are no well-established standards de-

scribing the internal format or structure of EHR files, and no defined coherent sets of valid procedures to store and manage them, and no access control rules to access or process them. The regulations of the different countries are incomparable and sometimes even contradictory, and the mechanisms and guidelines of care delivery organizations (clinics, hospitals, hospital networks, medical practices, etc.) differ extensively, even within the same country.

Thus, we have several choices for the definition of the EHR, the policies that should control it, the requirements to be fulfilled, and the problem cases associated to it. For the purposes of the AVANTSSAR project, we have chosen a rather generic scenario, which exemplifies typically difficult problems to be encountered in the different environments. One aspect that is important to have is related to the dynamics of policy management: Our scenario will allow authorized users to create, view and withdraw privacy policies established by patients, care delivery organizations and jurisdictions⁷. The policies will restrict the access to the EHRs at various levels of granularity depending on several factors: the class of information (e.g. lab report, diagnostic imaging study), the type of patients and their consent, the medical practitioners trying to access the data and their relation to the patient.

Our approach is close to the so-called BMA-Model [1], which has been proposed to regulate the access to health records. In that model each health record has its own access control list, an approach which we consider to lack practicality. We regard our approach as an implementation or refinement of the BMA-Model. In ours, each record has an implicitly associated AC list, via some indirections which facilitate the understanding for patients and clinicians. There are also differences in the treatment of the sensitivity of EHRs, ours allowing for more flexibility. For other related work, see [22] and [23].

We propose to use the following terminology:

Electronic Health Record (EHR) File is a logical⁸ “File” that contains a set of patient’s medical records (EHRs) in digital format, all belonging to one patient. The EHR file may contain a large variety of healthcare-related information. It may be made up of many different medical records at different locations and in different formats. An EHR file is in general not intended to contain the full medical history of a patient, but some types of EHR files are. Information may pass between the different EHR files in a controlled

⁷*Policies* regulate the access to EHRs, while the access to those policies are regulated by so-called *meta-policies*. In this way, authorized users are allowed to view or modify privacy policies within well-defined limits and rules.

⁸We use the term logical or virtual file, since we do not want to restrict the way the EHR File is implemented. It may be implemented as one or several XML files, as a directory, as a set of independent linked single files, as part of a database, etc.

manner, in the form of summaries (such as referrals or discharge letters to GPs). The mobile Patient-EHR file, the web-based Patient-EHR file, and the Electronic Case Record (eCR) are all particular cases of EHR files. Within the hospital scenario, we assume that each medical practitioner who treats the patient has controlled access to the relevant parts of the EHR file in a computer system of the hospital or hospital network. Whether the requested access is permitted or not depends of course on the type of medical data, the requester, the patient, their relationship, etc. From the point of view of the policy controlling the access to the EHR file, the hospital or hospital network is the “administration domain” of the policies and thus is responsible for implementing and managing the policy enforcement mechanisms and placing the correct enforceable policies and meta-policies in the system.

Thus, in order for an access request to the EHR to be honored, the policy must be evaluated. The policy itself will have two parts, a static part and a dynamic one. While both can be read, only the dynamical part can be written. Who is authorized to do which actions on the policy is determined by the static meta-policy. [Figure 17](#) shows the two levels of access. Clinician Y requests for an EHR, while clinician X makes an attempt to change the dynamic part of the access control policy. Acronyms used inside the figure: PDP - Policy Decision Point, PEP - Policy Enforcement Point.

Electronic Health Record (EHR) is a single part, a record, in an EHR file. The EHR is the atomic unit of access control (a user is authorized to read or write records, not parts of records). Each single record is of a particular type, which can be of a wide variety of types, for instance,

- Patient demographics, including name, address, date of birth, health insurance number and the name of the treating doctor, and the details of the family doctor,
- Medical history, examination reports of health and illnesses,
- Medications, including side-effects and interactions, allergies, vaccination, and immunization status,
- Laboratory test results, radiology images or photographs,
- Record of appointments and billing records,
- Patients’ directives, living wills, and health powers of attorney,
- Emergency contact info, etc.

Electronic Personal Health Information (PHI) is any information that is contained in an EHR file or have been deduced from it. Thus, a concrete piece

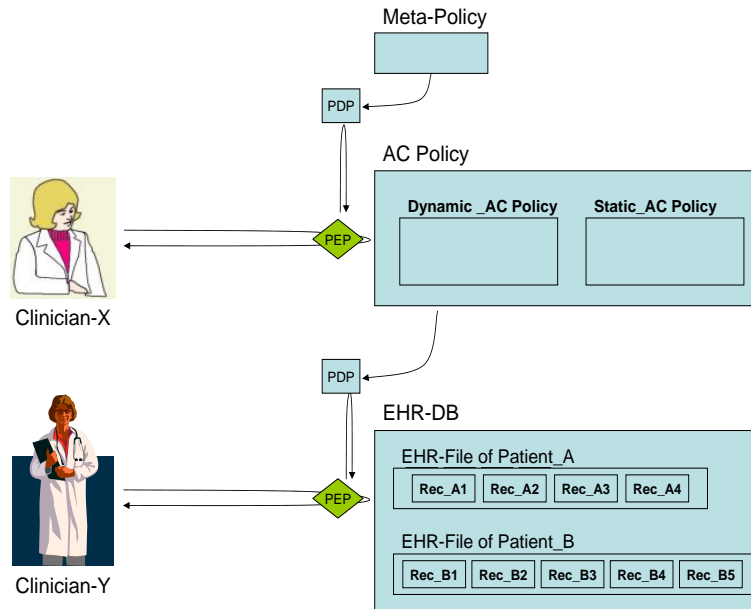


Figure 17: Logical Access Control Structure for the EHR.

of information like demographics, a living will, etc., is PHI if this specific instantiation of the information is included or has been retrieved or deduced from an EHR file. The rules defining the access control criteria for an EHR will depend, as was mentioned above, on the type of data. For instance, a highly sensitive record of a treatment for depression might be only available to his treating doctor (and perhaps a few others, on certain conditions), while a record of heart disease can be open to all staff, for the case of an emergency.

3.6.1 Scenario definition

We focus here on a rather prototypical EHR scenario: registering new patients in a clinic including the assignment of clinicians (doctors, nurses, radiologists, dentists, etc. depending on the system) to a patient, reading and updating a record, retrieving patient information from external sources, and providing the results of examinations and treatment to authorized external clinical entities.

For the sake of diversity, the access to an EHR will depend on several factors:

- The type (sensitivity group) of the record, which will itself depend on the

type of patient, say if the patient is a celebrity (VIP)⁹ or not, and on the patient having opted-in for a special more restrictive access control management of the EHR. Thus, if the patient is VIP or has opted for this restrictive policy, his default records will be of restricted type. If not, the default access control for the typical record is that all doctors of the practice or ward have access to it.

- The relation of the patient to the requester of the record (for instance, if the requester is the treating doctor or is in the same practice or clinic ward as the treating doctor).
- The consent that the patient has given.
- The access purpose, in particular, if the request is made in an emergency case.
- Regulation.

We will use in our scenario the following four sensitivity groups (but not “sensitivity levels” in the sense that they are not necessarily ordered in any particular way) for EHR records:

- *admin-EHR*,
- *emergency-EHR*,
- *normal-EHR*, and
- *restricted-EHR*.

Those sensitivity groups can be considered as attributes (more precisely, attribute values) of the EHRs; alternatively we may think of them as 4 different types of records. Together with the other attributes on patients, doctors, etc., they determine the access conditions on EHRs. Ideally, policies should be written in such a way that the different policy conditions (EHR sensitivity groups, clinician roles, local regulations, etc.) can be written independently from each other and composed when the system is initialized. We will follow this modular approach here.

EHRs that contain administration data are assumed to be of type *admin-EHR*, information that must be known by all staff for the case of an emergency are of type *emergency-EHR*, typical records are of type *normal-EHR*, and particular sensitive information (like a record of a treatment for depression) are of type *restricted-EHR*.

⁹In some countries the privacy of VIPs is specially protected, while in others this would be a very awkward assumption. Nevertheless, our choice is without any loss of generality: if there is no difference between patient we may simply assume that all are VIPs (or none).

We can assume that most of the EHR for the patient is of type *normal-EHR*, but there may occur situations where this is not the case.

The precise rules describing the permissions of users to access EHRs that are assumed as follows:

- The patient may read or append to any of his records, except in the case of statutory exemptions (court decisions, etc.).
- The treating doctor (note that a whole EHR has one and only one treating doctor) may read or append to any record in the EHR.
- Any clinician of the same ward or practice may read or append to record *emergency-EHR* and *normal-EHR* records.
- Any clinician in same network or system may read *emergency-EHR* records.
- For any other doctor the following condition holds: If he is able to connect and strongly authenticate to the Network or System, he may read or append to any record if the patient (and in some systems if the treating doctor) has consented this action.
- In case of a referral, the referring doctor may read the records of the doctor who made the corresponding examination.

Actions on the EHR are restricted further with the following obligations and constraints:

- For each read and write action, a log must be written (for audit).
- Each write action can only append or attach comments, information, etc. Existing entries can never be deleted.
- Each information appended must be time-stamped and signed at the moment of appending.

Note: The obligations are considered to be independent of the permissions and of the PEP. This means that obligations must be fulfilled in any case of access, via the PEP (which should always happen), but also in cases where the PEP was somehow bypassed (which should never happen, but sometimes does). That is, we want the obligations to be fulfilled, even under abnormal behavior.

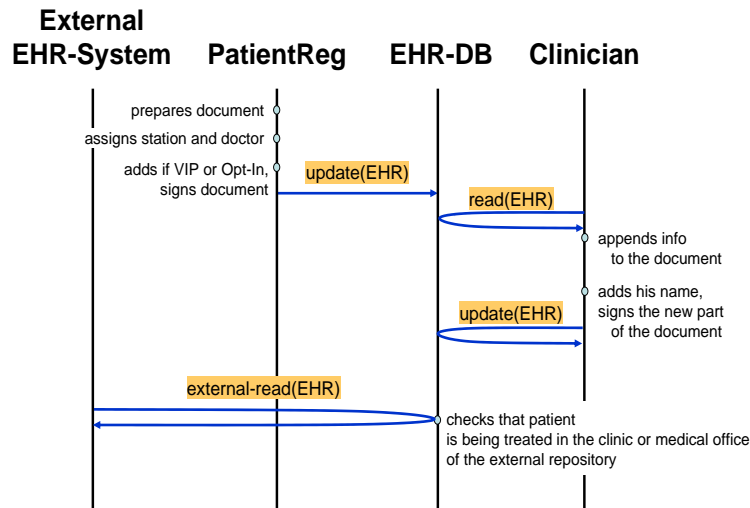


Figure 18: Basic Data Flow in the EHR Scenario

The typical workflow in the EHR scenario is shown in [Figure 18](#). It includes patient registration, and internal and external access to the EHR database.

In the case that the patient is a VIP or has opted for this choice, all EHRs except administration data and information that must be known by all staff, are under *restricted-EHR*. This means that a clinician will be able to access the information only if he is the treating doctor of the patient, or with consent of the patient.

When a new patient registers at the clinic, if the patient has no EHR in the clinic, a new EHR file is created by the front-desk (registration desk) of the clinic. This initial EHR normally contains only the patient demographics. On the other hand the receptionist may also access the policy governing the EHR access: he can assign the patient to a ward and to a doctor and may determine the type of patient (VIP or not, or if the patient opts for a more restricted access control than the default one). These parts of the initial EHR and the initial policy can be incomplete and can be completed or corrected later on by the same receptionist or by any authorized clinician. Note that corrections to old data does not delete the data, but will mark it as obsolete. If the patient has an EHR at this clinic, the receptionist may see and update the patient demographics, but not the rest of the EHR. More precisely, the patient demographics is controlled by *admin-EHR*. A record is under *admin-EHR* control rules, if it is available to the administration

staff (in the registration and accounting departments) and to the treating doctor.

The patient may opt that some or most of his records are *restricted-EHR*, meaning that stricter access control rules (to be discussed below) will apply to access his EHR. This decision may be delegated, if he is unable to take such a decision, or if he prefers, to a close relative or a representative of him. Also, if the patient has been announced as a VIP (or the receptionist considers so), the data will be of type *restricted-EHR*. In both cases, the patient must produce a verifiable, time-stamped “PatientCertificate” that can not be repudiated and certifies that he is being treated in this clinic.

The clinic may probably wish to retrieve information from external EHR files; in that case the patient will authorize the treating doctor to do so by means of his electronic signature on this “PatientCertificate”.

This type of certificates will be necessary to collect information from his family doctor or from other clinics and will be required from other institutions if the EHR is to be sent to them. These certificates have only a short validity period (weeks or months), and may be renewed by the patient. If not, the patient data becomes inaccessible to the clinic except in emergency case. An EHR is accessible to other clinics or family doctors if they process a valid certificate.

Now, or later, the treating doctor is assigned. This action must be logged.

In this scenario we look at City Clinic, three patients, three doctors, and a nurse who are involved in four different scenes viz. registration, requesting external EHR, reading of/writing in EHR and emergency. Let us go through each actor before we can go through the actual scenes so that we can understand their roles in each scene better.

City Clinic is one of the largest clinics in the city offering multi-specialty medical services to a large number of patients coming from different financial backgrounds. It comprises several medical doctors and nurses and SOA-enabled medical equipment. The City Clinic has three main logical IT Entities for our purposes:

EHR Database Server: Here the EHRs of all patients of the City Clinic are stored. It also includes a policy enforcement functionality which controls the access to the database.

Policy Administration Server (also including policy enforcement functionality). We assume that the *meta-policies*, i.e. the access control policies needed to read or write the EHR access policies, are static or their management is out of the scope of this case study. In our example, the following are dynamic policy aspects:

- Clinicians (doctors and nurses), as well as patients have assignments: There is a unique treating doctor for each patient. Each

doctor, nurse, and patient is assigned to a ward of the clinic. Clinicians can be assigned to clinical networks.

- Patients can be classified as VIPs, which means that all EHRs allow only restricted access. Typically the records are in the normal category, but for particular ones the patient can choose restricted access (opt-in).
- The dynamic part of the policy contains information about records which were written as consequences of referrals, as well as
- information derived from patient declarations on their consent that particular records may be read by particular internal or external clinicians.

Auditing Facility: In order to safeguard the patients' privacy according to regulations and consent directives, an audit trail of access to personal health information must be generated.

Dr. Adams, Dr. Baker and Dr. Collins are medical doctors of City Clinic.

Dr. Allen is the family doctor of Alice, the actress.

Nina is a nurse who belongs to Dr. Adams' ward at City Clinic.

Robert is the receptionist who takes care of the Registration Process.

Alice is an actress who wants to register at City Clinic for treatment.

Bob is a normal patient but has some sensitive information which he has the right to be kept secure.

Charlie is a normal patient who has a cardiac problem.

Now the scenes are described where the actors introduced above play their parts.

Scene 1: Registration Process. For each patient who wants to be registered in City Clinic, Robert takes his/her demographics and creates an EHR for them. At this stage the patient may opt for restricted-EHR to his patient data. Here Robert knows that Alice is a VIP and he proposes Restrictive Access to all of her EHRs, which she accepts. Since Bob believes that some of his EHRs will contain some sensitive information, he opts that part of his records should only be opened under *restricted-EHR*. In the case of Charlie, neither Charlie nor Robert believes that there is any sensitive information in Charlie's EHRs, but he has a Cardiac

problem which needs immediate attention in cases of emergency. Hence, all of his EHRs are kept under *normal-EHR*.

After filling the patient's demographics in the EHR, Robert assigns a treating doctor to each patient according to the patient's needs and some local policies (which are out of our scope). Alice, being a VIP, is assigned to Dr. Adams, who is assisted by Dr. Baker. Bob is assigned to Dr. Baker and Charlie is assigned to Dr. Collins.

Then Robert enters the assignments into the City Clinic information system, and they are automatically included into the database on the policy server. The system logs the assignments. This process provides the doctors with appropriate rights to access the patients' EHRs. Dr. Adams (being the treating clinician of Alice) will be in the position to access all records of Alice, while Dr. Baker by default will not, due to the restrictions on Alice's records. In order to change this situation, Alice signs a patient certificate which gives Baker the same access right as Dr. Adams. This declaration of user consent will also be included into the database on the policy server.

Scene 2: Providing User Consent for Requesting External EHRs. Rather often, clinics that are treating a patient need to retrieve part of or the whole EHR file from the patient's family doctor or from other clinics where the patient had earlier been treated. In our scenario, City Clinic's Dr. Adams needs to know some information about allergic reactions to substances from the family doctor of Alice, Dr. Allen.

Dr. Allen will not automatically send a requested record to a colleague. He belongs to a medical network that has a similar access control policy as the City Clinic. Dr. Adams explains the situation to Alice and points out she has to give her consent to the transfer of information. He then prepares a Patient Certificate (PCert) including the information that Alice is a patient of Dr. Adams at City Clinic and that she agrees the data to be sent. She signs the PCert electronically. Dr. Adams sends his request, the signed PCert along with his own public key to Dr. Allen. After verifying the PCert and the requestor's identity, Dr. Allen encrypts the required part or whole of the corresponding EHR in his system with using Dr. Adams' public key and sends the encrypted EHR to Dr. Adams, who can now read the needed information after decrypting it with his private key.

While Alice is staying in the clinic, her family doctor (Dr. Allen) is thinking about the best therapy to follow when Alice will be back from hospital. Now it is he who wants to look at Alice's EHRs stored at City Clinic. A similar procedure as described above has to take place. Allen manages to get the desired information, as he can show an appropriate PCert which was signed by Alice before she went to City Clinic.

Scene 3: Reading and Writing into EHR. After having a medical check of Alice, Dr. Adams and Dr. Baker want to update her EHRs by including the treatment they wish to give to her. Either of the two clinicians can make the corresponding entries into the database. Dr. Adams, being the treating doctor, has the access right per definition, while Dr. Baker has been authorized by Alice herself during the patient registration process, where Alice signed an appropriate declaration. During the night, Alice calls for the doctor on duty, in this case Dr. Collins. As Collins is not assigned to Alice, he is not in the position to access Alice's EHRs, which are not emergency relevant, as these records all belong to the *restricted-EHR* group. Now, Collins has to decide either to restrict himself to the emergency data, or to contact Adams or Baker for obtaining further information.

Some time later, a similar situation occurs to Dr. Collins after a medication of Bob, as in his case some important EHRs are in the *restricted-EHR* group, as well. Here we have to keep in mind that Dr. Baker (and not Collins) is Bob's treating doctor. So, only Dr. Baker is allowed to access these EHRs.

Early in the morning Nina (a nurse from the same ward) measures the blood pressure of Bob and writes the measurement in his EHR. She retrieves Bob's EHR, and as the corresponding parts are classified as *normal-EHR*, she is able to write the measurements into the EHR and signs the new entries before sending the record back to the repository. The signature helps to achieve non-repudiation and is mandatory according to the security policy.

The handling of Charlie's EHRs is much easier for the clinicians of the ward he is assigned to, as all entries are in the *normal-EHR* group which means free access to all data (for this particular group of clinicians).

Scene 4: Emergency. Charlie suddenly has a shortness of breath and requires Emergency treatment. Yep Charlie's treating doctor, Dr. Collins, is in on a business trip. Dr. Baker is contacted immediately. But since he does not have any information neither on Charlie's present illness nor on general health parameters, he feels somewhat uneasy. In order to find out the best treatment for Charlie, he uses the health information system. He indicates that an emergency condition has occurred. He signs this indication which then is logged by the system. Now, Baker can access the part of Charlie's EHR which is in *emergency-EHR* group. He immediately learns what has to be considered in emergency cases, and he can go ahead with an appropriate treatment.

Scene 5: Anonymization. Robert receives a request from the local university for data on particular cardiovascular diseases. This data should serve several medical students working on their theses. Robert knows that data of that kind may be given to universities and research institutes, but only in an anonymized form.

This means that Personally Identifiable Information (PII)¹⁰ like names have to be deleted, addresses have to be replaced by regional information, birth dates have to be replaced by time ranges, etc. The EHR database of the City Clinic supports the export of anonymized data for scientific application. Robert sends the desired data to the local university.

3.6.2 Security and trust requirements

In this section we summarize the description of the EHR scenario in the form of security and trust requirements.

Security Requirements.

1. All EHRs shall be kept secret (requirement of confidentiality). This means that access to EHRs has to be ruled in an appropriate way and that EHRs have to be protected while being transmitted via networks (e.g. using encryption).
2. Each access to the EHR database as well as to the policy repository shall be logged.
3. The EHR system shall realize non-repudiation of EHR access (typically using digital signatures).
4. Access to the EHRs or policies shall only be granted after appropriate identification and authentication. This may involve the federation of the identity of clinicians among diverse clinics.
5. Write access to EHRs shall be restricted: It shall only be allowed to append data; deletion is prohibited.
6. The EHR system shall realize integrity and authenticity of EHRs.
7. The access to EHRs shall be ruled by an Access Control Policy (ACP) depending on assignments, declarations of consent, etc. In our example above, the ACP reflects the following ideas:
 - Each patient is assigned to exactly one treating doctor.
 - Clinicians are assigned to wards, clinics, networks.
 - EHRs are classified.
 - Patients can opt in for stronger classifications.

¹⁰See also [subsubsection 3.1.2](#)

- Patient can declare their consent on access with respect to particular doctors.
 - Clinicians are granted access depending on their assignment, the EHR classification, or given patient consent.
8. Metapolicies shall rule the access to the dynamic part of the ACP.
 9. Data derived from EHRs are only allowed to be exported (e.g. for reasons of medical research) in an anonymized form.

Trust Requirements.

There are no particular trust requirements besides the following obvious one:

All clinicians (doctors, nurses, and receptionists) are assumed to handle the patients' personal information with confidentiality.

This requirement is a non-technical one, and it will not be regarded within AVANTSSAR.

3.6.3 Federation

Authorization in the EHR scenario heavily depends on user identification and authentication. Requirement 4 leads to the following problem cases:

Single Sign-On. Typically EHRs are distributed over various servers, and it is unacceptable for users to run through the authentication process again and again in order to access the EHRs. This gives rise to a demand for single sign-on.

Identity Federation. In cases of clinical networks we not only have to face the problem of distributed data, but also the fact that the users belong to different organizations. This gives rise to a demand for identity federation.

3.6.4 Authorization Policies

As described in [subsection 3.1.3](#), authorization policies determine the rules for access control, e.g. to data files. In general, they determine which subjects may perform which actions on which objects.

Access Control. In the context of EHRs, requirement 7 relies on policies that describe the rules for users (patients, doctors, clinicians) to read and write EHRs. Part of the access control policies is dynamic which gives rise to a further requirement (8) for metapolicies which rule the access to this dynamic part. Access control is addressed in 1 and 4, as well.

3.6.5 Accountability

The security requirements 2, 3, 5 for electronic health records rise problem cases that belong to the accountability family of problem cases.

Non-Repudiation. Requirements 3 and 5 deal with non-repudiation. A clinician should not be in the position to repudiate having made a particular entry to an EHR. Furthermore he/she shall not be able to modify an entry in a way that its former content is deleted.

Audit. Requirement 2 is quite typical for a situation as it is given with the EHRs. All attempts to access the EHR database have to be logged.

3.6.6 Privacy

Data Anonymization. EHRs contain highly sensitive data. In order to preserve the patients' privacy, excerpts of the EHRs are only allowed to be published in anonymized form. Requirement 9 leads to this problem case.

3.6.7 Application Data Protection

Data Integrity and Data Authenticity. For EHRs it is important that they are authentic and that they are protected against unauthorized modification. Requirement 6 leads to these two problem cases.

3.6.8 Communication Security

Message Confidentiality. EHRs require a high level of confidentiality. EHRs are transmitted over networks within the area of clinics (possibly LANs), but also (using public networks) between different clinics or doctors with own practices. Confidential communication is addressed in requirement 1.

References

- [1] R. J. Anderson. A Security Policy Model for Clinical Information Systems. In *1996 IEEE Symposium on Security and Privacy*, pages 30–42. IEEE Computer Society Press, 1996.
- [2] Endre Bangerter, Jan Camenisch, and Anna Lysyanskaya. A Cryptographic Framework for the Controlled Release Of Certified Data. In *Twelfth International Workshop on Security Protocols*. Springer-Verlag, 2004.
- [3] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Proceedings of the twentieth annual ACM symposium on Theory of computing (STOC 1988)*, pages 103–112, 1988.
- [4] J. Camenisch, A. Lysyanskaya, and M. Meyerovich. Endorsed E-cash. In *IEEE Symposium on Security and Privacy*, pages 101–115. IEEE Computer Society, 2007.
- [5] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Advances in Cryptology — Eurocrypt 2001*, LNCS 2045, pages 93–118. Springer Verlag, 2001.
- [6] Jan Camenisch, Dieter Sommer, and Roger Zimmermann. A general certification framework with application to privacy-enhancing certificate infrastructures. In *SEC*. Springer-Verlag, 2006.
- [7] Jan Camenisch and Els van Herreweghen. Design and implementation of the idemix anonymous credential system. In *ACM Computer and Communication Security*, 2002.
- [8] Common Criteria for Information Technology Security Evaluation (CC). ISO/IEC 15408, <http://www.commoncriteriaportal.org/>.
- [9] David Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology - Crypto '82*, pages 199–203. Springer-Verlag, 1983.
- [10] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The Second-Generation Onion Router. In *13th USENIX Security Symposium*, 2004.
- [11] D. Ferraiolo and R. Kuhn. Role-based access controls. In *15th NIST-NCSC National Computer Security Conference*, pages 554–563, 1992. <http://citeseer.ist.psu.edu/ferraiolo92rolebased.html>.

- [12] Fraunhofer ISST. Umsetzungsprojekt elektronische Fallakte (eFA). <http://www.fallakte.de> (in German).
- [13] German Ministry for Health (Bundesgesundheitsministerium). Die Gesundheitskarte. http://www.die-gesundheitskarte.de/gesundheitskarte_aktuell/elektronische_patientenakte/pdf/gesundheitskarte_aktuell_elektronische_patientenakte.pdf (in German).
- [14] Kjell J. Hole, Thomas Tjøstheim, and Vebjørn Moen. Next generation internet banking in norway. Technical Report 371, Department of Informatics, University of Bergen, 2008. <http://www.ii.uib.no/publikasjoner/texrap/pdf/2008-371.pdf>.
- [15] Monika Maidl, David von Oheimb, Peter Hartmann, and Richard Robinson. Formal security analysis of electronic software distribution systems. In *Proc. 27th International Conference on Computer Safety, Reliability and Security (SAFECOMP)*, LNCS. Springer, 2008. To appear, <http://ddvo.net/papers/SAFECOMP08.html>.
- [16] Web of trust. http://en.wikipedia.org/wiki/Web_of_trust.
- [17] Personally identifiable information. http://en.wikipedia.org/wiki/Personally_identifiable_information.
- [18] Richard Robinson, Mingyan Li, Scott Lintelman, Krishna Sampigethaya, Radha Poovendran, David von Oheimb, Jens-Uwe Buser, and Jorge Cuellar. Electronic distribution of airplane software and the impact of information security on airplane safety. In *Proc. 26th International Conference on Computer Safety, Reliability and Security (SAFECOMP)*, volume 4680 of LNCS. Springer, 2007. <http://ddvo.net/papers/SAFECOMP07.html>.
- [19] Andreas Schaad, Volkmar Lotz, and Karsten Sohr. A model-checking approach to analysing organisational controls in a loan origination process. In *SACMAT'06*, pages 139–149, 2006. <http://doi.acm.org/10.1145/1133058.1133079>.
- [20] System Engineering for Security and Dependability. <http://www.serenity-project.org/>.
- [21] Richard T. Simon and Mary Ellen Zurko. Separation of duty in role-based environments. In *IEEE Computer Security Foundations Workshop*, pages 183–194, 1997. <http://citeseer.ist.psu.edu/simon97separation.html>.

- [22] A. C. Simpson, D. J. Power, M. A. Slaymaker, D. Russell, and M. Katarova. On the development of secure service-oriented architectures to support medical research. *International Journal of Healthcare Information Systems and Informatics*, 2(2):75–89, 2007.
- [23] Jenny Ure, John Geddes, Clare Mackay, Sharon Lloyd, Andrew Simpson, David Power, Douglas Russell, Marina Jirotko, Mila Katarova, Martin Rossor, Nick Fox, Jonathon Fletcher, Derek Hill, Kate McLeish, Yu Chen, Joseph V Hajnal, Stephen Lawrie, Dominic Job, Andrew McIntosh, Joanna Wardlaw, Peter Sandercock, Jeb Palmer, Dave Perry, Robert Procter, Mark Hartswood, Roger Slack, Alexander Voss, Kate Ho, Philip Bath, Wim Clarke, and Graham Watson. Designing for e-health : Recurring scenarios in developing grid-based medical imaging systems. In *HealthGrid*, June 2006.