predict
prioritise
prevent

TRESPASS

Technology-supported Risk Estimation by Predictive Assessment of Socio-technical Security

Deliverable 5.3.1

Abstraction levels for model sharing

Project:	TRE$_S$PASS
Project Number:	ICT-318003
Deliverable:	D5.3.1
Title:	Abstraction levels for model sharing
Version:	1.0
Confidentiality:	Public
Editor:	Margaret Ford
Cont. Authors:	Barbara Kordy, Aleksandr Lenin, Wolter Pieters, Rolando Trujillo, Jan Willemson
Date:	2013-10-31

Part of the Seventh Framework Programme
Funded by the EC-DG CONNECT

Members of the TRESPASS Consortium

1.	University of Twente	UT	The Netherlands
2.	Technical University of Denmark	DTU	Denmark
3.	Cybernetica	CYB	Estonia
4.	GMV Portugal	GMVP	Portugal
5.	GMV Spain	GMVS	Spain
6.	Royal Holloway University of London	RHUL	United Kingdom
7.	itrust consulting	ITR	Luxembourg
8.	Goethe University Frankfurt	GUF	Germany
9.	IBM Research	IBM	Switzerland
10.	Delft University of Technology	TUD	The Netherlands
11.	Hamburg University of Technology	TUHH	Germany
12.	University of Luxembourg	UL	Luxembourg
13.	Aalborg University	AAU	Denmark
14.	Consult Hyperion	CHYP	United Kingdom
15.	BizzDesign	BD	The Netherlands
16.	Deloitte	DELO	The Netherlands
17.	Lust	LUST	The Netherlands

Document History

Authors		
Partner	Name	Chapters
CHYP	Margaret Ford	1, 2, 3, 4 A
CYB	Aleksandr Lenin	A
CYB	Jan Willemson	1, 2, 3
UL	Barbara Kordy	3
UL	Rolando Trujillo	1, 3
TUD	Wolter Pieters	General comments, consistency with other deliverables
UT		Contributions at a later stage
DTU		Contributions at a later stage
ITR		Contributions at a later stage
TUHH		Contributions at a later stage
AAU		Contributions at a later stage
BD		Contributions at a later stage

Quality assurance		
Role	Name	Date
Editor	Margaret Ford	2013-10-31
Reviewer	Lorena Montoya	2013-10-14
Reviewer	Jan-Willem Bullée	2013-10-15
WP leader	Jan Willemson	2013-10-30
Coordinator	Pieter Hartel	2013-10-31

Circulation	
Recipient	Date of submission
Project Partners	2013-10-31
European Commission	2013-10-31

Acknowledgement: *The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 318003 (TRESPASS). This publication reflects only the authors' views and the Union is not liable for any use that may be made of the information contained herein.*

Contents

List of Figures

Management Summary

Key takeaways:

- This deliverable contains a review of different use cases for sharing security information: sharing within a single organisation, sharing between organisations and sharing in a cross-border context.
- We propose a design for the standard library service, instantiating it with the Model Pattern Library (MPL) and Attack Pattern Library (APL). This is intended to promote the reuse of modular elements to improve the process of model development.
- Current best practices for sharing sensitive information between organisations are described in an extensive appendix incorporating two case studies (one on e-health and the other on federated database management).

The issues and opportunities arising from sharing security information in different contexts are explored in considerable detail in this document. In particular, we have considered the complex practical, commercial, legal and privacy issues which may influence the choices an organisation makes regarding the sharing of data.

This deliverable contains a review of different use cases for sharing security information: sharing within a single organisation, sharing between organisations and sharing in a cross-border context. Requirements occurring in each use case have been identified and a suitable architecture for meeting these requirements has been developed.

It has been identified that to some extent, the appropriate levels of abstraction may become evident from frequent repetition of particular items within a model (for instance commonly repeating sub-trees within an attrack tree may relate to theft or impersonation). This particular tendency was noted during the practical development of an attack tree to model the IPTV case study. It was found that particular elements, such as 'stealing a payments card', were common to a number of different scenarios.

A design has been developed for a standard library service, composed of the Model Pattern Library (MPL) and Attack Pattern Library (APL). This design takes a deliberately modular approach to creating the library, in order to encourage the reuse of modular elements and contribute to the process of model development.

The best practices described in relation to the case studies in the appendix will form the basis for ongoing development of the essentially decentralised approach to model sharing currently proposed by TRE$_S$PASS.

1. Introduction

1.1. Motivation and challenges

Cyber security breaches continue to increase along with the fast-growing adoption of IT technologies by small, medium, and large companies. A recent survey performed in the UK (*The 2013 Information Security Breaches Survey*, 2013) suggests that such security breaches cause losses in the order of billions of euros per year. According to the same study, companies have on average raised their investment in security to 10% of their IT budget in 2013. The main drivers for information security expenditure include: preventing downtime and outages, complying with laws/regulations, protecting the reputation of the organisation, and protecting intellectual property.

Both the government and the private sector have recognised the need to protect against cyber security attacks, in particular, to tackle the security of critical and strategic infrastructure assets. A key factor in this respect is the gathering, analysis, and sharing of cyber security data, *e.g.* successful and unsuccessful attacks, known vulnerabilities and threats, attacker profiles and motivations.

Cooperation and sharing have been acknowledged as a breakthrough in the fight against cyber attacks. Some examples can be found in (Narasimhan, Varadarajan, & Chandrasekaran, 2010; Frincke, Tobin, Mcconnell, Marconi, & Polla, 1998; Cuppens & Miège, 2002; Nojiri, Rowe, & Levitt, 2003; Koutepas, Stamatelopoulos, & Maglaris, 2004) for Spam and DoS attacks detection and mitigation, and in (Huang & Lee, 2003; Locasto, Parekh, Keromytis, & Stolfo, 2005; Gkantsidis & Rodriguez, 2006) for securing peer-to-peer and *ad hoc* networks. Sharing security related data allows the dissemination of system vulnerabilities, threats, and incidents within an integrated view. It also provides best security practices and solutions for continuous improvement of IT security products, which increases their commercial appeal.

However, the sensitive nature of information security data makes companies reluctant to share something so critical. Most cyber attacks have a discovery phase as a first step, and this type of information is very valuable for attackers who could identify weak points in the network, critical authentication servers that are not replicated, potential bottlenecks, machines infected with worms or other malware, etc. This vital information is normally invisible to outsiders and it is unlikely that conducting scans from outside the network would reveal it. Consequently, sharing or publishing such privileged information could potentially have devastating results for an organisation which is under attack.

Recognising both the benefits and the risks inherent in sharing information security data is the first step towards delivering more effective security solutions. Organisations face

the challenge of defining what data to share, how, and to whom, in such a way that the economic profit is maximized. Several studies (Gal-Or & Ghose, 2005; Gao, Zhong, & Mei, 2013; Fang, Parameswaran, Zhao, & Whinston, 2012) have been published on these economic incentives in recent years, many of them based on Game Theory.

1.2. Goals

The modelling process used by today's risk assessment frameworks must support adaptivity of the models to a fast-changing threat environment. It has become impossible for one person to keep up with all these changes. Hence, the modelling process must provide a convenient way of distributing descriptions of new attacks and their properties to communities of people using risk assessment tools.

This issue was partly addressed by the FP7 project SHIELDS[1] that introduced a special online service called SVRS[2] for sharing model components, more specifically attack trees, describing various technical attacks. However, this service does not currently appear to be maintained or used on any scale and historically its focus has tended to be on issues relating specifically to software development.

Despite these issues, SVRS provides a useful prototype for the TRE$_S$PASS project to build upon. In particular, there is an opportunity to continue where the SHIELDS project left off, and implement a *standard library* of elementary attacks, to be integrated with the TRE$_S$PASS tools.

More specifically, the goals of Task 5.3 are the following.

- Defining technical specifications for the Model Pattern Library (MPL) and Attack Pattern Library (APL).
- Proposing and developing best practices for sharing security information.
- Providing support for the risk assessment workflows, including sharing the models within different kinds of communities (in-house, inter-organisational, cross-border).

The focus of this deliverable is on different use cases for information sharing, collecting the implied requirements and making architectural decisions on how to meet these requirements.

1.3. Structure of the document

The Deliverable consists of three chapters and an appendix. In Chapter 2, we discuss different general approaches to model sharing based on the organisational requirements.

[1] http://www.shields-project.eu/
[2] https://svrs.shields-project.eu/SVRS/

We consider the requirements arising from the needs of one organisation for their internal security assessment, inter-organisational aspects and requirements set by the cross-border sharing of information and models. Chapter 3 covers several technical solutions for achieving model sharing and seamless tool integration.

Finally, Appendix A presents two case studies of sensitive information sharing between different organisations, one from the e-health domain and another concerning federated database management.

1.4. Foreground and background

In the current deliverable, the content items containing various reviews of existing sharing frameworks and security patterns are considered as background. Foreground developed during the project consists of the definition of pattern libraries, the associated workflows and the sharing routines in various settings.

2. Approaches to security information sharing

Several contexts exist that put different restrictions on sharing of models and attack descriptions. For the purposes of the TRESPASS project, we will consider sharing within an organisation, sharing between organisations, cross-border sharing, and sharing with the general public (essentially publishing). For information sharing there may be some methodologies in common between these contexts, but each is inherently different in nature.

2.1. Sharing within an organisation

The most significant challenge faced in intra-organisational sharing is how to minimize the effects of a security breach. In particular, it is necessary to deal with perhaps the most complex security threat: the Insiders (Gordon & Loeb, 2006). Therefore, companies need to rely on frameworks for measuring knowledge-sharing risks such as the one proposed by Trkman and Desouza (Trkman & Desouza, 2012). A different approach is proposed in (Soper, Demirkan, & Goul, 2007) where a single security hub aimed at notifying all other hubs in the event of a breach is considered.

Under usual circumstances, the group of people who need detailed and up-to date information about the potential vulnerabilities and possible related attacks will be reasonably well defined within an organisation. This group may include the Risk Manager, Chief System Administrator (for IT risks) and Human Resources Manager (for risks originating in human factors, including social engineering risks). Mutual trust within this group of people is essential for secure operations within the organisation. However, it is still necessary to identify the information which they require to achieve their operational goals and limit their access accordingly.

The TRESPASS tools have a significant role to play in supporting information sharing within an organisation. For the purposes of keeping the organisational models and risk assessment up-to-date, TRESPASS tools may be installed and used locally, within the organisation's network. There may also exist localised versions of the attack and model pattern libraries (see Chapter 3 for more discussion on these libraries). It will of course be possible to update the libraries with content from similar libraries held in trusted organisations (see Section 2.2), or published libraries (see Section 2.4).

Dissemination and updating of security procedures is handled via policies and regulations that all employees of the organisation are expected to follow. Decisions about whether to

inform all employees regarding specific breaches are taken on a case by case basis, and these decisions are governed by the organisation's policies and culture.

2.2. Sharing between organisations

Either because organisations engage in joint development projects or because they create strategic alliances, they may need to share knowledge and information within a closed group. This inter-organisational sharing is usually based on trust and ruled by company-specific policies. Two main examples are:

- IT-ISAC (Information Technology Information Sharing and Analysis Center)[1] founded in 2001, which serves as a central repository for security-related information. It includes IT industry companies such as Oracle, IBM, EDS, and Computer Sciences. IT-ISAC distributes much of its information anonymously, which helps members to feel more comfortable when sharing organisation-critical information.
- FIRST (Forum for Incident Response and Security Teams)[2] encompasses several security incident response teams from government, commercial, and educational organisations. It aims to promote cooperation, coordination, and information sharing among its own members, as well as the global security community.

In order to support inter-organisational sharing of the models and attacks, synchronisation between the respective libraries is necessary. There are several possible approaches to implementing this:

- If the organisations form a strategic alliance, they may establish a separate entity responsible for keeping the library available and managing the access control credentials.
- If some of the organisations have a greater level of knowledge and/or capabilities than the others in the group, they may take the lead in maintaining the library services.
- If the organisations all have a sufficient and similar level of capability, each of them may choose to maintain their own library service. In this case periodic synchronisations between the libraries are required. Although this is organisationally the most complicated setup, it does allow for the greatest flexibility in the control of each organisation's security information.

In order to facilitate such usage scenarios, essentially a revision control system is needed. The requirements arising from the scenarios above are the following:

- The revision control system should allow for distributed data management so that each organisation may run a master copy of the database for itself.

[1] http://www.it-isac.org/
[2] http://www.first.org/

- The system should also allow for some (public) central servers distributing some common knowledge patterns. In principle, these central servers can also be logically equivalent to the other servers in the distributed network.
- The system should allow fine-grained access control.

Such systems are well known in the software development industry. Best known examples of distributed or mixed information sharing solutions include Git[3], Mercurial[4] and Bazaar[5]. At this point further research is required to clarify which would best suit the needs of the TRESPASS project.

2.3. Sharing in a cross-border environment

Information sharing in the cross-border environment is conceptually similar to the inter-organisational scenario. Where the international parties are business organisations, the requirements are very similar to those described in Section 2.2, although the legal environment may be more complex.

However, the situation is potentially more serious where state actors are concerned. In this case, information regarding security breaches not only potentially threatens individual organisations, but the very existence of whole nation states may be at stake. For this reason, decisions regarding the types of information to be shared with designed recipients must be taken by officials with an appropriate level of authority.

However, from a technical point of view, the solutions described in Section 2.2 can also be applied in this context.

2.4. Publication

It is a widely accepted position in the security community that it is in the public interest to make security related data freely and publicly available. In a networked world, every weak and insecure node potentially affects the rest, so that security is nowadays a global interest. Secrecy can not be used as a cover for bad design, with flaws that are sure to be exploited. In addition, vulnerabilities and threats can be better managed if they are notified as soon as they are detected. Below, we provide some examples of efforts for publishing security data.

The European project SHIELDS aimed at increasing software security. Together with security guidelines, SHIELDS provides a Security Vulnerability Repository Service (SVRS) containing security models, provided by security experts, independent of the applied development process. A similar idea is the Open Risk Model Repository (ORI-MOR)[6]. How-

[3] http://git-scm.com/
[4] http://mercurial.selenic.com/
[5] http://bazaar.canonical.com/en/
[6] http://www.somap.org/orimor/

ever, both SVRS and ORI-MOR are currently unmaintained, which makes them interesting only from the viewpoint of a historic data source for TRESPASS.

SHIELDS relies on two other projects: Common Attack Pattern Enumeration and Classification (CAPEC)[7] and Common Weakness Enumeration (CWE)[8]. CAPEC provides a taxonomy of attack patterns, which makes it easier to find relevant information related to known attacks. CWE provides a unified and measurable set of weaknesses enabling better description of software security tools.

Even though not originally designed to be used with the TRESPASS tools, ORI-MOR, CAPEC and CWE provide useful attack descriptions to seed the APL. We performed preliminary experiments with CAPEC and it is possible to convert its content to a format convenient for APL. For public attack descriptions we plan to set up one public repository based on the distributed revision control technology described in Section 2.2.

Security Content Automation Protocol (SCAP) (Waltermire, Quinn, Scarfone, & Halbardier, 2011) is aimed at organising, expressing, and measuring security-related data. SCAP can also be used to increase the security of enterprise systems by automatically verifying the installation of patches, checking system security configuration settings, and examining systems for signs of compromise. The standards in SCAP have been designed in such a way that security configuration guidance can be both human and machine-readable. This allows automated verification of security configuration settings.

Vulnerabilities have also been modelled with a directed acyclic graph representing the causes of a vulnerability and their relationships (Ardi, Byers, Meland, Tondel, & Shahmehri, 2007). Others repositories and mailing lists are X-Force[9] and the Open Web Application Security Project[10]. Unfortunately, these projects lack a common format and, individually, do not cover all aspects of a vulnerability.

Vulnerabilities and incidents can also be shared in the form of logs, which form the core source of evidence for computer forensic investigations. A multi-layered approach to publishing them in a secure and private way can be found in (Slagell & Yurcik, 2005).

[7] http://capec.mitre.org/
[8] http://cwe.mitre.org/
[9] http://xforce.iss.net/
[10] https://www.owasp.org

3. Content distribution in TRESPASS

3.1. Pattern-based architecture

When building the models for risk assessment, one does not expect the whole model to be 100% unique. In fact, quite the opposite is true. Even though each organisation operates differently, the socio-technical description of its operational environment will have components which are practically identical to other organisations. Most of the companies have some office space consisting of rooms with rather standard features (doors, windows, network connections, etc.), they have employees with standard sets of credentials, etc. It would be very time-consuming to create these parts of the models by hand over and over again for each organisation. Furthermore, when it comes to estimating the parameters of the attacks against these standard components, possible inconsistencies between these estimations may cause some risk assessments to become invalid. It therefore makes a lot of sense to include reliable parameter estimations as a part of the commonly shared body of knowledge.

Frequently occurring modules in the models will be proposed as *patterns* which will serve as default, reusable components. Patterns are especially interesting in the case of large-scale models whose treatment poses numerous practical challenges. Creating large models is very time consuming and should therefore be modularized and, if possible, automated. Management of such models is often performed by several independent parties. For instance, security analysis of the physical infrastructure of a company and of its computer network might be delegated to two different groups of specialists.

There is no unanimity concerning reuse of security patterns in the security modelling community. Two main attitudes can be distinguished: the *favourable* and *unfavourable* approach.

- Researchers representing the *favourable* approach believe that reusing security patterns makes threat analysis more efficient and accurate. According to this approach, generating a general model from existing libraries constitutes a good starting point for further model refinement and analysis. Furthermore, although new technological opportunities arise every day, empirical studies show that most attackers reuse the same attack vectors with little or no modification (Dulaunoy, 2012). Often the same company is attacked several times by an intruder exploiting the same, already known vulnerability.
- The representatives of the *unfavourable* approach affirm that using predefined model components blocks the modellers' creativity (which is especially important in creating models such as attack trees) and results in unusable and impractical models.

Followers of this unfavourable approach claim that each system, organisation and related security scenarios are unique and, as such, they deserve customized models.

Interestingly, there exists strong scientific evidence supporting the favourable approach with respect to the reuse of security patterns (Schumacher, Fernandez-Buglioni, Hybertson, Buschmann, & Sommerlad, 2006; Meland, Tøndel, & Jensen, 2010; Sindre, Firesmith, & Opdahl, 2003; Jensen, Tøndel, & Meland, 2010). The opposite, unfavourable approach is often advocated in informal discussions or meetings, but, to the best of our knowledge, there are no scientific or experimental results proving its validity.

Our initial experiments with the IPTV case study (see Deliverables 2.2.1 (The TRE$_S$PASS Project, D2.2.1, 2013) and 7.1.1 (The TRE$_S$PASS Project, D7.1.1, 2013)) seem to imply that the favourable approach is applicable for the TRE$_S$PASS project. The low-level attack trees emerging from the case study look generic enough to be useful also in other scenarios, but in order to establish this with greater certainty, further research is needed.

The idea of reusing security patterns is not new. For instance, in the attack tree field, reuse of attack patterns was already mentioned in 2001 by Moore et al. (Moore, Ellison, & Linger, 2001; Linger & Moore, 2001). Subsequently, a positive initiative was taken by the FP7 project SHIELDS, in which the Security Vulnerability Repository Service (SVRS) was developed. SVRS is an on-line library of various security models. Currently, it contains around 35 generic attack trees and 10 domain specific misuse case diagrams. SVRS can be accessed by registered users and the regular download rate is about 20 models each week. For now, the repository is mostly used for teaching purposes, its active maintenance has been stopped.

By setting up the SVRS repository, an initial effort was made to allow the reuse of security patterns. However, further research is necessary in order to develop methodologies and tools for a consistent and sound use of libraries and composition of larger models from pre-existing templates.

TRE$_S$PASS considers two different security patterns: *TRE$_S$PASS Model Patterns* and *Attack Patterns* provided respectively by the Model Pattern Library (MPL) and Attack Pattern Library (APL). TRE$_S$PASS Model patterns basically contain templates of the TRE$_S$PASS model that might be applicable to several organisations. In addition to the model, each TRE$_S$PASS model pattern is enriched with attributes describing the type of organisation it models (*e.g.* estimated number of employees, high level description of business processes, critical assets, and geographical area.) These patterns are aimed at being reused in large-scale modelling of organisations.

In contrast, attack patterns are re-usable components describing known attacks rather than security models. They have already been used on previous projects such as CAPEC. However, CAPEC is limited to securing software development and does not provide support for detailed and quantifiable attack trees.

Our approach is to extend the fundamentally sound CAPEC taxonomy in order to support different kinds of socio-technical attacks. Therefore, each attack pattern will contain several attributes such as description, attack prerequisites, examples or instances, solutions

and mitigations, related vulnerabilities, etc. In addition, a field named *Attack tree* is added so as to describe the attack through a well known and widely supported formalism.

3.1.1. Initial development of pattern libraries

As stated above, we foresee two kinds of modules to be provided by the libraries – TRE$_S$-PASS model components (that are themselves valid TRE$_S$PASS models for simple standard components) and attack trees.

At the time of writing (month 10 of the TRE$_S$PASS project), the specific class of models to be used by the TRE$_S$PASS tools has yet to be confirmed within WP1. However, it has been decided that at least in the first phase of the project, the *lingua franca* for the modelling tools and computational routines (as defined by WP3) will be attack trees. This decision is justified by the consideration that attack trees currently provide a relatively well-understood formalism for attack representation. Equally, several computational tools are available for automated reasoning about them (see Deliverable 3.1.1 of TRE$_S$PASS (The TRE$_S$PASS Project, D3.1.1, 2013)).

Hence, for the first iteration of Task 5.3 we will also use the formalism of attack trees for distribution purposes. As a consequence, for the time being we will concentrate on the development of APL and consider the development of MPL in a later stage of the project.

A separate question to address both in the initial development phase and future iterations is the quality of the patterns and parameter estimations to be distributed. Since the outcome of the entire analysis phase will depend on these values, they may become mission critical for effective risk management.

When considering only the intra-organisational use case, the organisation is free to choose any quality assurance mechanism it feels most comfortable with. But for the other scenarios (inter-organisational, cross-border and public), common standards must be agreed upon.

The quality of both TRE$_S$PASS model and attack patterns will be assured by following basic guidelines:

- Validation of patterns and parameter estimations by both internal and external reviewers.
- Existence verification to check whether the pattern can be easily abstracted to an already existing pattern.
- Abstraction verification to ensure that the pattern is not too specific.

External reviewers come into play when security patterns get shared between different organisations, countries or even published. In the latter case, essentially the whole international community can participate in the validation efforts.

3.2. APL workflow

In general, the workflow of building and using the TRESPASS models in the first iteration will be the following.

1. A socio-technical system model will be built using the TRESPASS tools.
2. An attack tree is generated from the socio-technical model.
3. The attack tree is analysed to determine the best attack vectors.

However, the problem is that the attack trees that can in principle be generated from the socio-technical system models are not detailed enough to enable the analysis of Step 3 above. While working through the IPTV case study attack tree we discovered that the system model provides input to the attack trees up to the level where the attacks become technical (see Deliverables 2.2.1 (The TRESPASS Project, D2.2.1, 2013) and 7.1.1 (The TRESPASS Project, D7.1.1, 2013) for more detailed discussions on the IPTV case study).

For example, the system model can describe the payment card and state that the card has an access code. Hence it is possible to generate the nodes "Get payment card" and "Get access codes" from the model. However, the model has no deep knowledge about all the possible techniques that can be utilized for getting the access codes (e.g. skimming or shoulder-surfing). This also implies that the attack tree that can be generated from the model is incapable of providing the computation engine with sufficient details to allow the actual computations to be performed.

This is where the APL comes into the play. We see that skimming and shoulder-surfing are actually generic techniques not really specific to the IPTV case study, but can be applied to different passwords, PINs and other kinds of access tokens. Hence, the system model should be able to state that the card access code is a kind of password (e.g. by some sort of an ontology), and using this statement the corresponding subtree can be fetched from the APL.

Hence, the workflow above can be extended as follows.

1. The socio-technical system model will be built using the TRESPASS tools.
2. An attack tree is generated from the socio-technical model.
3. Subtrees corresponding to the specific technical attacks are fetched from the APL and added to the tree.
4. The attack tree is analyzed to determine the best attack vectors.

3.2.1. The levels of abstraction for the APL

As discussed above, our first iteration of the APL will be able to distribute attack trees to be used as components in construction of larger trees. Determining their abstraction level, we actually have to talk about two different aspects of abstraction, namely the levels of the root nodes and leaf nodes of the trees.

The level of abstraction of the root nodes of the attack trees provided by the APL is pretty much determined by the role of these trees in the entire TRE$_S$PASS workflow. Considering that the socio-technical TRE$_S$PASS model describes the system in terms of standard components, the APL must provide parameters and attacks against these components. Hence we can say that the APL must provide the trees with the root nodes that fit into the high-level attack trees produced by the tools of WP1.

This level is also appropriate from the privacy point of view – the standard components can be assumed to be well-known anyway and there is no particular reason to limit access to them in the APL.

However, the level of abstraction of the leaf nodes of the trees in the APL is a much more sensitive issue. Since these trees are generally developed with a specific organisation in mind, care must be taken not to leak any of the organisation-specific private information through the sharing of the resulting tree.

The level of acceptable leakage is actually determined by the potential user base of the developed attack tree component. A certain level of specificity may be acceptable for sharing within the same organisation, but not between organisations or cross-border. A more detailed discussion of these issues is included in Chapter 2.

3.3. Unified Glossary of Security Terms

Besides the above-described concept of patterns, APL also relies on the concept of a Unified Glossary of Security Terms and Security Patterns.

A glossary of security terms is aimed at minimising the misunderstanding and confusion of security terminologies. Each individual may have different experiences and training regarding security, which makes the sharing and dissemination of security patterns particularly challenging. These security terms and concepts might also be different depending on the type of organisation, cultural aspects, country regulations, amongst other factors. Therefore, it is of paramount importance to define security patterns which are compliant with a unified glossary of security terms.

The Unified Glossary will be built based on different sources. First, we will use the glossary developed by Deliverable 6.2.1 (The TRE$_S$PASS Project, D6.2.1, 2013), which will then be extended using several existing comprehensive glossaries. The SANS Institute[1], established in 1989, publishes an information security glossary with hundreds of words

[1] http://www.sans.org/

and acronyms. More recently, in 2013, the National Institute of Standards and Technology updated its glossary of information security terms with more than 200 pages of definitions. A draft can be found in pdf format in the latest revision of the Interagency Report 7298[2].

To improve searching, each security pattern should be marked up with one or more keywords taken from the glossary. These keywords can then be used by a search engine to assist analysts during the development process. Analysts might also search existing TRE$_S$PASS model patterns by means of different organisational properties or find detailed attack patterns traversing the taxonomy of attack patterns. Figure 3.1 depicts the relationships between the main components of the pattern library.

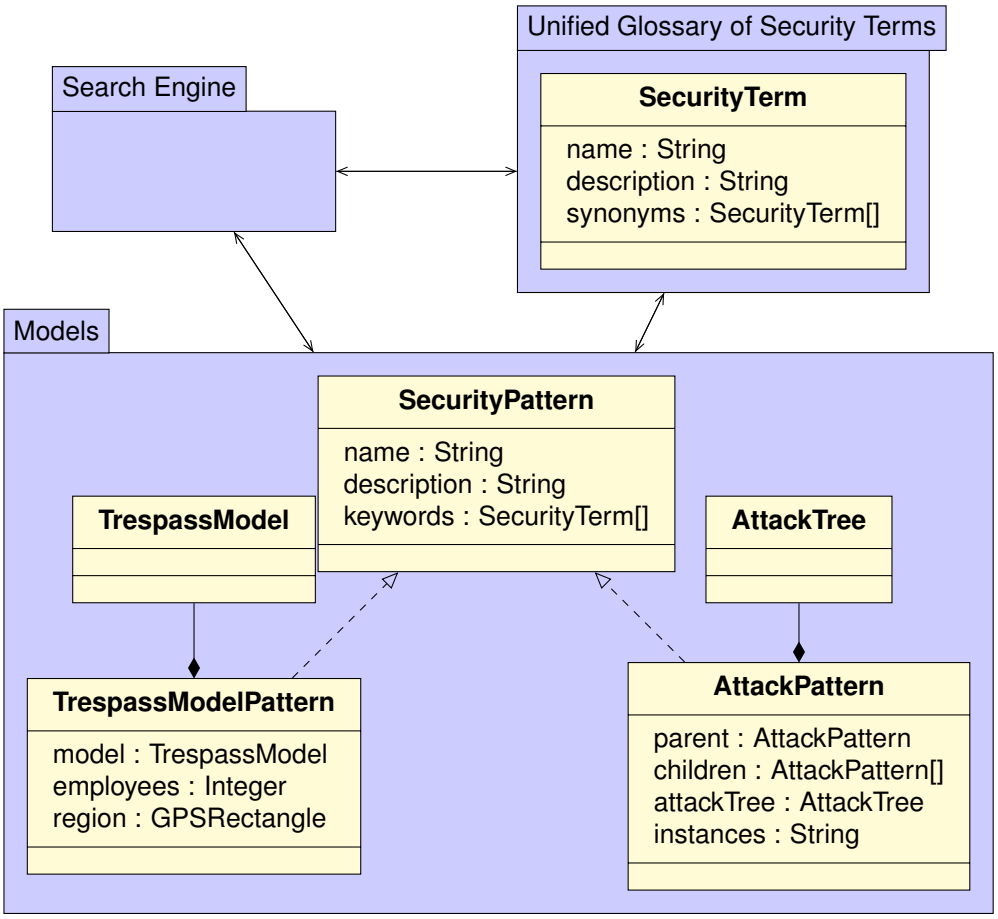

Figure 3.1.: Pattern library architecture

3.4. Community engagement

As discussed above, every effort was made by the SHIELDS project to engage with a community when developing their SVRS service. However, this service is currently un-

[2] http://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf

maintained, showing that this aspect of the SHIELDS project was not sustainable at that point.

We argue that the main reason that the SHIELDS SVRS was not accepted by the wider community was its lack of flexibility in fitting into the existing workflows and lack of support for local installations. The single standard use case foreseen by the SVRS developers was publishing, which may not be very suitable for a sensitive corporate environment.

The TRE$_S$PASS project will implement a considerably more decentralised system, utilising a distributed revision control system for the TRE$_S$PASS tools. This allows for independent installations of the TRE$_S$PASS toolset and does not introduce a single point of responsibility into the ecosystem. Using a decentralised approach we aim to achieve a more natural fit of the TRE$_S$PASS tools into existing workflows and consequently also more lasting engagement of the community.

Of course, decentralisation also introduces the potential risk of inconsistency between different branches of the library. However, we see this rather as an opportunity to enable discussion between the experts who maintain these branches.

3.5. Sharing attacker profiles

The attack trees generated from the socio-technical system models in a way represent an omnipotent attacker who can do everything in an instant. In reality, attackers have different kinds of limitations (e.g. monetary, time or skill limits). We have developed methods to take these limitations into account in the computational process (see Section 3.2.3 “Multi-parameter attack tree computation with attacker profiling” in Deliverable 3.3.1 (The TRE$_S$PASS Project, D3.3.1, 2013)). Since the limitation sets (attacker profiles) need to be formally described in any case, the APL can in principle also accommodate them in a ready-made manner. At the moment, this is a planned development for future iterations of the APL.

After completing the attacker profiling based on the resource limitations, the next step is to model attacker strategies. These strategies correspond to different attacker incentives that have implications on attack actions even if all the attack resources are equivalent. For example, terrorists and employees who are about to get fired from the organisation may have completely different motivation and hence would attack differently. Including such considerations into the profiles remains the subject for future research as well.

4. Conclusions

The issues and opportunities arising from sharing security information in different contexts are explored in considerable detail in this document. In particular, we have considered the complex practical, commercial, legal and privacy issues which may influence the choices an organisation makes regarding the sharing of data. The effects of the scope for information sharing are also investigated: within a single organisation, between organisations, more widely in a cross-border context, or equally through publication.

Existing initiatives to promote the sharing of security information, such as SHIELDS and CAPEC, have been evaluated and conclusions drawn regarding possible approaches to creating viable structures for enabling this type of sharing. The nature of community engagement has also been explored, in order to identify ways in which participants can derive value from their involvement in such an ecosystem and develop a commitment to ongoing sharing. Proposals for varying levels of abstraction have been developed in this context.

It has been identified that to some extent, the appropriate levels of abstraction may become evident from frequent repetition of particular items within a model (for instance commonly repeating sub-trees within an attrack tree may relate to theft or impersonation). This particular tendency was noted during the practical development of an attack tree to model the IPTV case study. It was found that particular elements, such as 'stealing a payments card', were common to a number of different scenarios.

A design has been developed for a standard library service, composed of the Model Pattern Library (MPL) and Attack Pattern Library (APL). This design takes a deliberately modular approach to creating the library, in order to encourage the reuse of modular elements and contribute to the process of model development. This approach has been developed with reference to ongoing activities in other work packages, especially WP1 and WP3. Over the coming months possible approaches to developing a Unified Glossary of Security Terms in support of pattern sharing will be considered.

The best practices described in relation to the case studies in the appendix will form the basis for ongoing development of the essentially decentralised approach to model sharing currently proposed by TRE$_S$PASS.

References

The 2013 information security breaches survey. (2013). Retrieved from https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/200455/bis-13-p184-2013-information-security-breaches-survey-technical-report.pdf

Ardi, S., Byers, D., Meland, P. H., Tondel, I. A., & Shahmehri, N. (2007). How can the developer benefit from security modeling? In *Proceedings of the the second international conference on availability, reliability and security* (pp. 1017–1025). Washington, DC, USA: IEEE Computer Society. Retrieved from http://dx.doi.org/10.1109/ARES.2007.96 doi: 10.1109/ARES.2007.96

Cuppens, F., & Miège, A. (2002). Alert correlation in a cooperative intrusion detection framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy* (pp. 202–). Washington, DC, USA: IEEE Computer Society. Retrieved from http://dl.acm.org/citation.cfm?id=829514.830542

Digirecord. (2013). Retrieved from http://www.digilugu.ee

Dulaunoy, A. (2012). *The attackers' principles.* (Research presentation at the University of Luxembourg, http://satoss.uni.lu/seminars/srm/pdfs/2012-Alexandre-Dulaunoy.pdf)

E-health. (2013). Retrieved from http://www.e-tervis.ee/index.php/en

E-health publications. (2013). Retrieved from http://pub.e-tervis.ee

Fang, F., Parameswaran, M., Zhao, X., & Whinston, A. (2012). An economic mechanism to manage operational security risks for inter-organizational information systems. *Information Systems Frontiers*, 1-18. Retrieved from http://dx.doi.org/10.1007/s10796-012-9348-y doi: 10.1007/s10796-012-9348-y

Frincke, D., Tobin, D., Mcconnell, J., Marconi, J., & Polla, D. (1998). A framework for cooperative intrusion detection. In *Proc. 21st NIST-NCSC National Information Systems Security Conference* (pp. 361–373).

Gal-Or, E., & Ghose, A. (2005). The economic incentives for sharing security information. *Info. Sys. Research*, *16*(2), 186–208. Retrieved from http://dx.doi.org/10.1287/isre.1050.0053 doi: 10.1287/isre.1050.0053

Gao, X., Zhong, W., & Mei, S. (2013). Security investment and information sharing under an alternative security breach probability function. *Information Systems Frontiers*, 1-16. Retrieved from http://dx.doi.org/10.1007/s10796-013-9411-3 doi: 10.1007/s10796-013-9411-3

Gkantsidis, C., & Rodriguez, P. (2006). Cooperative security for network coding file distribution. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings* (p. 1-13). doi: 10.1109/INFOCOM.2006.233

Gordon, L. A., & Loeb, M. P. (2006). Economic aspects of information security: An emerging field of research. *Information Systems Frontiers*, *8*(5), 335–337. Retrieved from http://dx.doi.org/10.1007/s10796-006-9010-7 doi: 10.1007/s10796-006-9010-7

HI7. (2013). Retrieved from http://www.hl7.org

Huang, Y.-a., & Lee, W. (2003). A cooperative intrusion detection system for ad hoc networks. In *Proceedings of the 1st ACM workshop on Security of ad hoc and*

sensor networks (pp. 135–147). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/986858.986877 doi: 10.1145/986858.986877

Jensen, J., Tøndel, I. A., & Meland, P. H. (2010). Experimental threat model reuse with misuse case diagrams. In M. Soriano, S. Qing, & J. López (Eds.), *Icics* (Vol. 6476, p. 355-366). Springer.

Koutepas, G., Stamatelopoulos, F., & Maglaris, B. (2004). Distributed management architecture for cooperative detection and reaction to DDoS attacks. *J. Netw. Syst. Manage.*, *12*(1), 73–94. Retrieved from http://dx.doi.org/10.1023/B:JONS.0000015699.50210.e3 doi: 10.1023/B:JONS.0000015699.50210.e3

Linger, R. C., & Moore, A. P. (2001). *Foundations for Survivable System Development: Service Traces, Intrusion Traces, and Evaluation Models.*

Locasto, M., Parekh, J., Keromytis, A., & Stolfo, S. (2005). Towards collaborative security and P2P intrusion detection. In *Information Assurance Workshop, 2005. IAW '05. Proceedings from the Sixth Annual IEEE SMC* (p. 333-339). doi: 10.1109/IAW.2005.1495971

Meland, P. H., Tøndel, I. A., & Jensen, J. (2010). Idea: Reusability of threat models - two approaches with an experimental evaluation. In F. Massacci, D. S. Wallach, & N. Zannone (Eds.), *Essos* (Vol. 5965, p. 114-122). Springer.

Moore, A. P., Ellison, R. J., & Linger, R. C. (2001). *Attack Modeling for Information Security and Survivability* (Technical Note No. CMU/SEI-2001-TN-001). Carnegie Mellon University.

Narasimhan, H., Varadarajan, V., & Chandrasekaran, P. R. (2010). Towards a cooperative defense model against network security attacks. In *WEIS.*

Nojiri, D., Rowe, J., & Levitt, K. (2003). Cooperative response strategies for large scale attack mitigation. In *DARPA information survivability conference and exposition, 2003. proceedings* (Vol. 1, p. 293-302 vol.1). doi: 10.1109/DISCEX.2003.1194893

Schumacher, M., Fernandez-Buglioni, E., Hybertson, D., Buschmann, F., & Sommerlad, P. (2006). *Security patterns: integrating security and systems engineering*. Chichester, England, Hoboken, NJ: John Wiley & Sons. Retrieved from http://opac.inria.fr/record=b1124387

Sindre, G., Firesmith, D. G., & Opdahl, A. L. (2003). A reuse-based approach to determining security requirements. In *Proc. 9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03* (p. 16-17).

Slagell, A., & Yurcik, W. (2005). Sharing computer network logs for security and privacy: a motivation for new methodologies of anonymization. In *Security and privacy for emerging areas in communication networks, 2005. workshop of the 1st international conference on* (p. 80-89). doi: 10.1109/SECCMW.2005.1588299

Soper, D. S., Demirkan, H., & Goul, M. (2007). An interorganizational knowledge-sharing security model with breach propagation detection. *Information Systems Frontiers*, *9*(5), 469–479. Retrieved from http://dx.doi.org/10.1007/s10796-007-9055-2 doi: 10.1007/s10796-007-9055-2

The TRE$_S$PASS Project, D2.2.1. (2013). *Technical data extraction prototype.* (Deliverable D2.2.1)

The TRE$_S$PASS Project, D3.1.1. (2013). *Initial requirements for quantitative analysis tools.* (Deliverable D3.1.1)

The TRESPASS Project, D3.3.1. (2013). *First report on stochastic analysis methods.* (Deliverable D3.3.1)

The TRESPASS Project, D6.2.1. (2013). *Initial refinement of functional requirements.* (Deliverable D6.2.1)

The TRESPASS Project, D7.1.1. (2013). *Initial requirements for implementation of case studies.* (Deliverable D7.1.1)

Trkman, P., & Desouza, K. C. (2012). Knowledge risks in organizational networks: An exploratory framework. *J. Strateg. Inf. Syst.*, *21*(1), 1–17. Retrieved from http://dx.doi.org/10.1016/j.jsis.2011.11.001 doi: 10.1016/j.jsis.2011.11.001

Waltermire, D., Quinn, S., Scarfone, K., & Halbardier, A. (2011). *The technical specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.2.*

X-road. (2013a). Retrieved from http://www.eesti.ee/eng

X-road. (2013b). Retrieved from http://www.ria.ee/x-road

X-road. (2013c). Retrieved from http://e-estonia.com/components/x-road

X-road technical report. (2013). Retrieved from http://www.cyber.ee/home/information-systems/document-files/X-Road_technical.pdf

A. Case studies of private information sharing

This section includes an overview of two case studies which demonstrate some of the concepts considered in this deliverable: the complexity inherent in sharing data at scale, the value of a decentralised approach in managing this where possible, and the vital importance of separation of duties in controlling access to potentially sensitive data which is shared between multiple parties.

The e-health case study presents a practical solution to sharing data which is both time-critical and extremely sensitive and delineates the roles of the different parties involved in this process and the relationships between them.

The X-road case study identifies the solutions to compatibility issues which might arise and the different legal and practical implications of the associated safeguards and processes. It also considers the requirements for extending this type of system to support sharing in an international context.

A.1. E-health

E-health (*E-Health*, 2013) is a datastore which is part of (and owned by) the state information systems in Estonia, in which health-related information is stored and processed. This includes information required for the establishment and fulfillment of the healthcare service contract, information required to ensure the quality of healthcare and patients' rights, to ensure the protection of public health, including maintaining data registers of information reflecting both health status and healthcare management.

The establishment of e-health can be traced back to 2003 and it evolved from a plan to start handling various medical documents digitally. The system has 2 types of end-users: the patients and the medical staff offering healthcare services. The patients' portal is the system web-based front-end through which people can access data related to their visits to doctors and healthcare institutions, the medical investigations, diagnoses, etc, that have been registered in the system since a particular medical facility has joined the e-health system. Users can log into the system using the Estonian ID Card. Starting from 1 January 2009 every medical service provider has an obligation to register their patients' healthcare data in the digirecord system.

The e-health information system consists of 4 subsystems:

1. digirecord - digital health history.

2. digiregistration - service allowing patients to get a thorough overview of available appointment times across Estonia and providing the possibility to book or cancel for a suitable time and visit a preferred doctor.
3. digiregister - digital archive where diagnostic imaging study pictures are gathered (e.g. x-ray).
4. digirecipe - a database of electronic prescriptions issued by doctors.

In this overview we will describe the digirecord subsystem in more detail. Digirecord (*Digirecord*, 2013) is the central and most important part of the state-wide healthcare information system, that aggregates the most important data on patient health. It gives a convenient and thorough overview to the doctor about the patient's diagnosis, prescriptions, vaccination and medical testing results that can easily be compared to similar results from testing undertaken previously. Digirecord allows the healthcare professionals to share documents created in the course of treatment with one another and offers a time-critical and personal data querying capability. The information system saves the patients' medical history and provides individual patient care physicians with appropriate access. All patients have full access to any data stored about themselves in digirecord at any time. The system allows the use of anonymised data for statistical purposes, for assessing the quality of healthcare services, and for medical research purposes.

A.1.1. Technical architecture

The architecture of digirecord is SOA based. Standardised communication to external information systems is performed through the central messaging center. The services are autonomous and independent from each other, while addition and maintenance of new services is convenient and operator-friendly. The system is highly scalable and fault-tolerant, being duplicated in 2 separate locations. The message transfer and processing module, called the agent center is the core module of digirecord. Additional modules can be integrated into the system and the preferred method of integration is through the agent center. The agent center offers the following services to other components of the system:

- central security solution.
- system of users' rights, access control and maintenance.
- authentication and multilevel authorisation of users.
- integration with external information systems (via X-Road platform (*X-Road*, 2013a)).
- managing and processing of messages, supporting messaging-based integration with other modules of the system.
- logging.

A.1.2. Messages and corresponding standards

Digirecord exchanges all nationally established medical documents through messaging. All messages moving in the system are standardised. The digirecord agent center is capable of handling all standardised messages. Message processing and all other functional aspects are performed by the agents (functional modules), created within the digirecord or e-health projects. The format of messages that are supported is a nationally established standard, based on the international XML-based standard HL7 v3 (*HL7*, 2013). The encipherment of the message contents using the international standards, such as SNOMED and LOINC, is also standardised. The standards applied in digirecord data sharing are established by legislation for all healthcare providers in Estonia. The standards enforced for medical documents, classifications, lists and the publication process are governed by the Estonian e-health foundation and are publicly available (*E-Health publications*, 2013).

A.1.3. System security and threats

Over the last few years the digirecord system became one of the largest critical information systems. High security requirements for the system are derived from the following risks:

1. **Centralised information system** – Thus all the confidential and private information gets aggregated to one physical location, which is a single point of failure and besides this produces a high risk of their misuse or leakage. Because of the high levels of risk and the criticality of the information, this information must be secured from everyone, including technical and maintenance staff, which means that the insider threat must be eliminated.
2. **Medical personnel have no restrictions on access to patients' health-related information** – this approach assumes a treatment relationship between the patient and the doctor, which is easily verifiable, but leaves substantial potential for an authentication attack, as this often gives an attacker the opportunity to hide his traces.
3. **Very high risk of system opacity** – unlike paper documents, a complex digital system is not observable by end users at a single glance.

To mitigate the above mentioned risks, the digirecord system design follows these three secure design principles:

1. **The risk in the central system is balanced by numerous technical and administrative security measures** – it is based on the so-called complex security principle, according to which any single atomic attack on the system will not cause significant damage. For this an attacker needs to perform a sequence of interconnected coordinated elementary attacks, that constitutes a complex attack. The system doesn't have a superadministrator, that could have access to all the data – the roles in the system are distributed according to minimal privileges, required to fulfill work tasks. Information stored in the central system in encrypted, personal data and health-related data are stored separately from one another in separate databases. All the

data saved on the hard drives is encrypted, the database accesses this information with the help of a dedicated security module - this helps to mitigate risks in case of data leakage, in case of a hard drive theft, or unauthorised copying. All actions performed on the system are monitored and analysed afterwards. Monitoring solutions help to detect early attack phases and automatically apply countermeasures that prevent an attacker from completing and succeeding with all the steps of the complex attack. The above mentioned measures make the likelihood of a successful attack on the central system negligible.

2. **For all digirecord users secure authentication mechanisms are used** – in order to log into the system authentication using an ID-card, Mobile-ID or similar technology is used. Password-based authentication is strictly prohibited in the digirecord system.
3. **For all data saved to digirecord, the principle of maximum transparency is applied**. All actions – adding new data or registration, editing, accessing, querying, viewing, etc. – leave certain traces, that cannot be changed or hidden afterwards. This enables the patient to query information about who accessed his health-related entries and when, when those entries were created, modified, etc.

The principle of separation of duty in relation to specific security personnel ensures that even digirecord technical and maintenance staff (e.g. system administrators) have no opportunity to bypass the logging system or change the log entries. Every document saved in the system is digitally signed by its respective author in order to exclude malicious modification.

A.1.4. Outline of the key features and design principles

- Sensitive and time-critical information.
- Centralised SOA based information system with central management service.
- Autonomous services.
- Easy addition and maintenance of services.
- Scalable and fault-tolerant system design (duplicated in 2 separate locations).
- Centralised security solution.
- Centralised rights and access control system and maintenance.
- Potential for integration with external information systems for data sharing and exchange.
- Modular architecture.
- Integration of modules via centralised message processing system.
- Thorough logging facilities.

- XML-based messages.
- Insider threat must be eliminated.
- Complex security principle is implemented.
- No superadministrators implemented - separation of duties enforced.
- Encryption of the stored information and the hard drives data is stored on.
- Different types of data are stored separately in isolated databases.
- Automatic monitoring of all actions and events, alerts are raised.
- Automatic response to possible incident alerts, automated application of countermeasures.
- Use of secure authentication mechanisms (password-based authentication prohibited).
- Principle of maximum transparency of systems - every action in the system leaves a unique trace that cannot be changed or hidden afterwards (even by administrators).
- Digital signatures on stored data to exclude malicious modifications and plausible deniability of authorship.

A.2. X-Road

The data exchange layer X-Road (*X-Road*, 2013b) is the backbone of the state information system e-Estonia that was launched in 2001. It is a technical and organisational environment which enables secure internet-based data exchange between the state's information systems and is used by the government of Estonia, state agencies and private companies. This invisible but crucial environment allows various state information systems, both public and private sector, to interact, making e-services possible, and allows information to be shared securely. Institutions are not locked into any one type of database or software provider. All of the Estonian e-services solutions that use multiple databases use X-Road (*X-Road*, 2013c)

X-Road allows institutions and people to exchange information securely, ensures access to the data maintained and processed in state databases. One of the key elements of e-Estonia is the decentralisation of its databases, which means that there is no single owner or controller, every governmental agency or private sector business can choose the product that suits their needs, services can be added one at a time, as they become available.

X-Road was built with scalability and extendability in mind – public and private sector institutions can connect their information systems to X-Road. This enables them to use the e-services offered by X-Road in their own electronic environment, as well as offering their e-services via X-Road to other institutions. The ready-made data exchange layer that can be used by all institutions makes data sharing more effective in the state agencies, as well

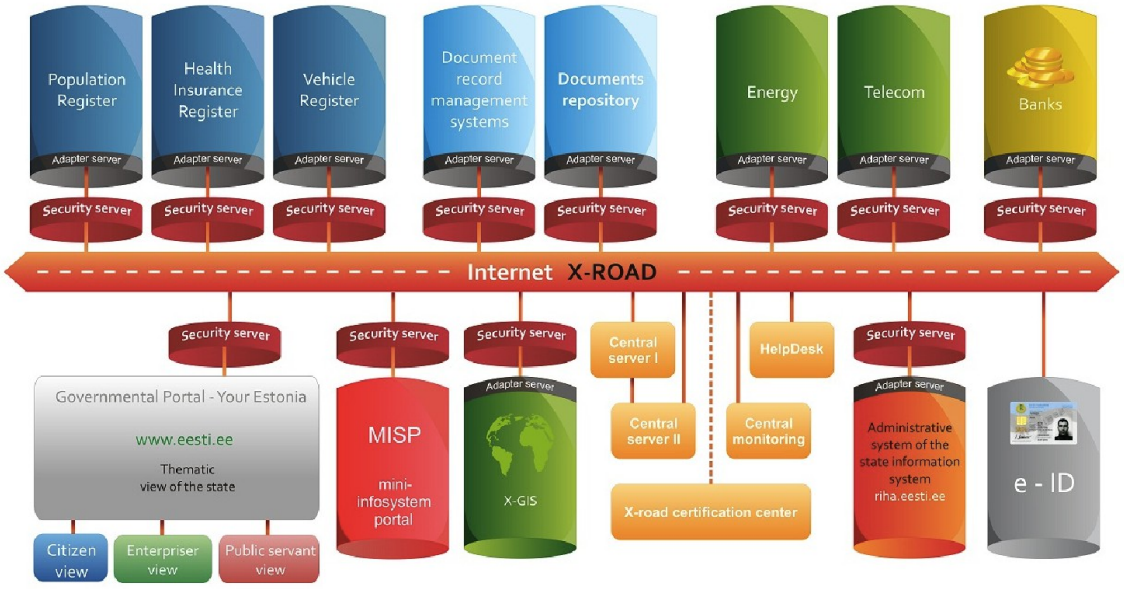

Figure A.1.: Example of integration of e-services via X-Road

as data sharing between the citizen and the state. Besides that, X-Road supports public requests for information. For citizens public web portals are available that allow state databases to be queried by an individual for information about himself/herself. Officials can use services, such as the document exchange center, in the information systems of their own institutions. This automates the officials' work and communication, as well as improving security.

The key features of X-Road are the following:

- **Maximum scalability** – X-Road can easily scale up to any number of connected databases or services.
- **Platform independence** – connecting a new service to X-Road does not depend on an enterprise's operational system, database platform or web server.
- **Evidential value** – all outgoing messages are digitally signed and all incoming messages are logged. The message log is cryptographically protected, incorporating both encryption and timestamping. These messages are legally binding for both sender and receiver and can be used as evidence in court.

A.2.1. Technical solution

If an enterprise or agency wants to connect to X-Road, either as a service provider or service consumer, it needs to install a dedicated X-Road security server on its premises, which acts as a firewall between the agency's information system and the Internet, routing messages to their recipients and securing messages both for transport (by encryption) and for long-term validation (by signing and logging). On the service provider side, an

additional adapter server is required, which transforms the requests obtained through the security server into the format understood by the register and translates the response in the reverse of this process (*X-Road Technical Report*, 2013).

Web-services were chosen as a vendor and platform neutral message exchange protocol, while transparent use of web services minimises the impact on existing systems and simplifies the integration of the X-Road services. Information is exchanged in XML-based query-response protocols, such as SOAP with support for the two-way transliteration to XML-RPC. Newer versions support SOAP attachments and asynchronous operations, where X-Road servers queue messages targeted to their organisations. The SOAP messages are protected by digital signatures and time-stamping mechanisms.

In addition to real services provided by participating organisations, X-Road also provides some meta-services that can be used to discover the structure and properties of the system. E.g. it is possible to obtain the list of organisations providing and consuming different services. For each service provider, it is also possible to discover the list of services provided to consumers (only services available to individual consumers are shown), as well as being able to obtain a WSDL description of the service and use it for automatic generation of user interfaces.

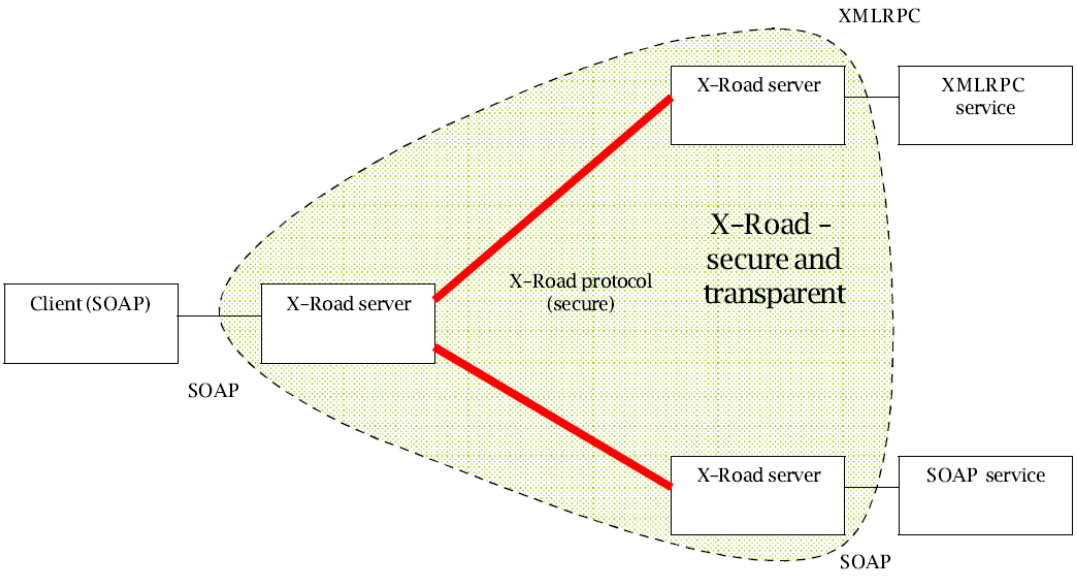

Figure A.2.: General X-Road infrastructure

The so-called Central Agency is the controlling and monitoring entity in the X-Road infrastructure, whose main goal is to ensure the legal status of the information exchanged via X-Road by enforcing the stated policies, maintaining X-Road consistency and integrity. The Central Agency provides some technical services, like a monitoring service (monitoring of all servers in the system is used for resolving operational problems, detecting security vulnerabiities, as well as collecting and analysing statistics about messages and system usage), and a web-based portal - a simple and convenient X-Road front-end for the citizens and SMEs who may lack sufficient IT capability.

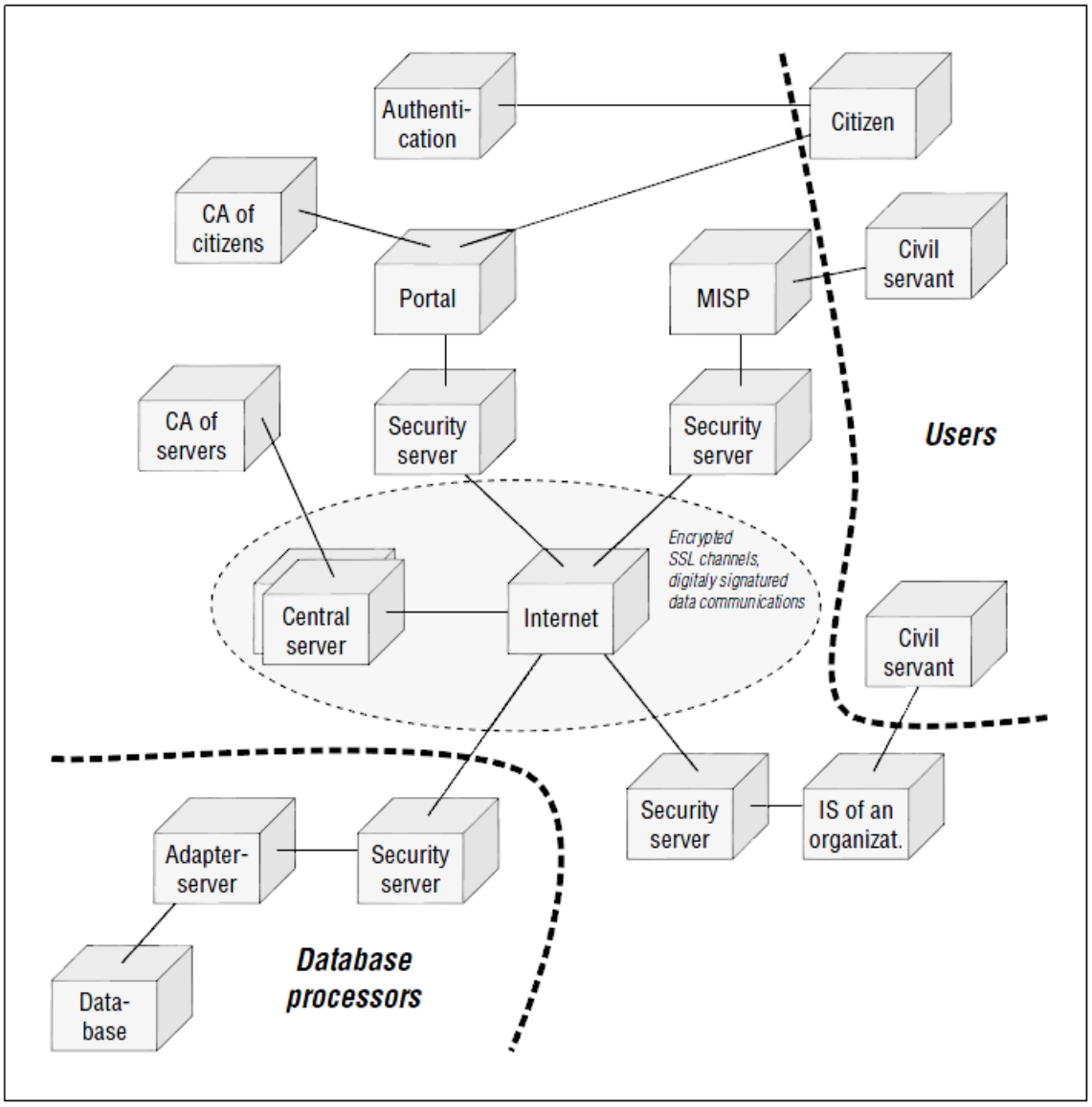

Figure A.3.: The functional structure of the X-Road system

A.2.2. Security aspects of X-Road

Due to the critical nature of information exchanged via X-Road, the security requirements are rather high – registries contain mostly personal data that in some cases is needed in real-time.

The properties of the security requirements for the X-Road infrastructure were the following (*X-Road Technical Report*, 2013):

- All applications required *authenticity*, *integrity* and *evidential value* of the data. Evidential value is the ability of the receiver to prove the origin of the data, as well as the non-repudiation of the sender of the data.
- X-Road had to provide high availability, as it is used by time-critical applications, like verifying identities on the border or performing police operations.
- Confidentiality was required in most, but not all cases.

In order to achieve high availability of the infrastructure, X-Road was designed as a distributed system with a minimal number of central services. Three kinds of security-related central services needed by the X-Road infrastructure and provided by the X-Road Central Agency are:

- **Certification** – an off-line process which is not really vulnerable to availability threats.
- **Time-stamping** – is used for log management and is thus not time-critical.
- **Directory service** – is used to distribute address and certificate validity information, which are both security and/or time critical in many scenarios. The directory service is built on top of Secure DNS (DNSSEC). Signed zones ensure that data cannot be tampered with. All X-Road servers located in participating organisations have their own local caching DNS servers to ensure the redundancy and availability of the directory information in case of (partial) network outage.

X-Road protocol supports redundant servers and load balancing. X-Road servers are also equipped with anti-D.o.S. solutions. Most of the data exchanged via X-Road is not public and has some special access rules. To combat external attackers the data exchanged is encrypted using the SSL protocol. In order to prevent internal attacks, a two-level access control mechanism is used: inter-organisational and intra-organisational. The X-Road system core deals only with inter-organisational access control, where one organisation grants access rights for some of their services to other organisations. It is the responsibility of the other organisation to ensure that only authorised people can use this service by using whatever technical or organisational means it prefers. The obligation to enforce appropriate usage of the service is stated in the service provisioning contract between the two organisations. Thus isolating the details of the authentication and access control mechanisms used internally by the organisations is one of the key success factors of X-Road.

A.2.3. Outline of the key features and design principles

- Security critical real-time data.
- Middleware.
- Decentralised architecture – no single owner or controller.
- Maximum scalability, extendability and integration.
- Platform and technology independence.

- Transparent use of services.
- XML-based secure query-response communication protocols.
- SOAP with attachments and support for asynchronous operations and two-way transliteration to XML-RPC.
- Message queueing and prioritisation.
- Digital signatures and timestamp protection on messages.
- Single controlling and monitoring entity (Central Agency).
- Automatic internal state monitoring and detection of security vulnerabilities.
- Collection and analysis of statistics on messages and system usage.
- High availability, authenticity, integrity, and evidential value of the data.
- Thorough logging.
- Logs are cryptographically secured.
- Redundancy and load balancing.
- Anti-D.o.S. solutions.
- Data stored and exchanged is encrypted (inside message). Messages encrypted in turn.
- Two-level distributed access control mechanism.
- Separation of legal obligations and responsibilities.
- Isolation of the details of the authentication and access control mechanisms.

A.2.4. X-Road

The Cybernetica AS technical report (*X-Road Technical Report*, 2013) on the X-Road system covers the problems related to international X-Road implementation (as a ready-made secure data sharing solution) and proposes several solutions to this. Here we will outline briefly the main ideas of this document.

When there is a need to deploy the X-Road based data sharing infrastructure on an international basis, future developments of infrastructure and policy standards will be needed, with both legal and technical systems needing amendments. The report to a great extent outlines the issues of creating a cross-border infrastructure, leaving the local infrastructures intact and supporting their interactions.

The approach to establish a cross-border data sharing infrastructure based on X-Road suffers from several shortcomings. Establishing a new cross-border infrastructure with a new Central Agency would need major agreements between different organisations, e.g. the question in whose premises the new Central Agency would reside needs to

be answered. It is not clear how entities with contradicting interests would reach such agreements. The complexity of the joining procedure might decrease the organisations' motivation to join the international X-Road considerably.

The major technical problem to be solved is integrating different DNSSEC directory services. Directory Service is the most security critical component of X-Road; on the other hand, information contained in the directory of one participating infrastructure (e.g. keys, addresses) must be available for other infrastructures too. Other services such as logging, time-stamping, monitoring and frontends do not require such a level of integration.

Deploying an X-Road based infrastructure on an international level is possible, but this requires that corresponding decisions have been made and agreements between the participants have been achieved.