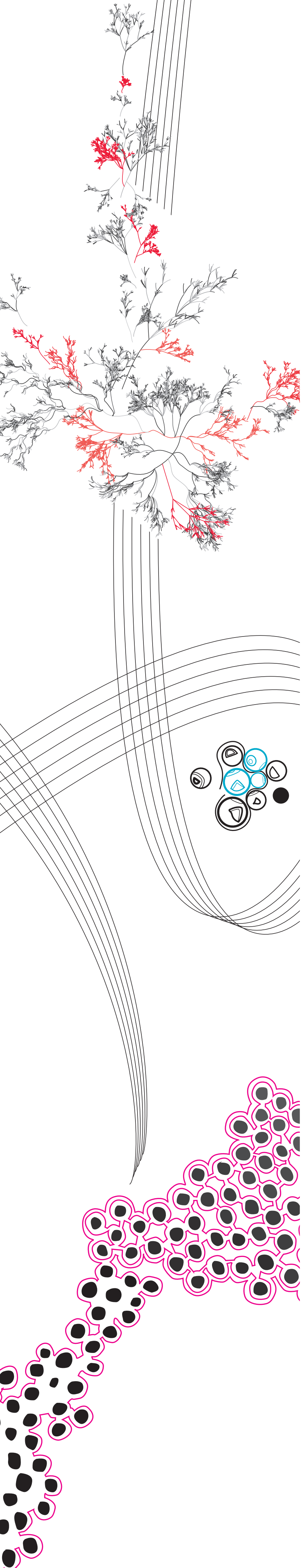

August 19, 2014

Master Thesis

Development and Validation of a Personal Information Security Assistant Architecture

Roeland H.P. Kegel

Electrical Engineering, Mathematics and Computer Science (EEMCS)
Services, Cybersecurity and Services (SCS)

Exam committee:
prof.dr. R.J. Wieringa
dr.ir. D. Hiemstra

UNIVERSITY OF TWENTE.

This thesis presents and validates the first iteration of the design process of a Personal Information Security Assistant (PISA). The PISA aims to protect the information and devices of an end-user, offering advice and education in order to improve the security and awareness of its users. The PISA is a security solution that takes a user-centric approach, aiming to educate as well as protect, to motivate as well as secure. This thesis first presents the method and its application by which stakeholders are elicited and classified. Requirements are then elicited using these stakeholders. 4 architectural alternatives for PISA are then proposed. Finally, these alternatives are validated by a traceability analysis, a prototype implementation of a specific alternative and feedback by a focus group of experts. In summary, this thesis presents stakeholders, goals, requirements and proposed architectures for the PISA and contains a validation of the latter.

Acknowledgements

As with any significant work, this thesis would not be the same without the effort and time of many other individuals. As such, I would like to explicitly thank the following people for making my thesis what it is today: my main supervisor, Roel Wieringa, for devoting time to systematically consider all parts of my thesis, leading to major revisions and a better document. My other supervisor, Djoerd Hiemstra, for showing enthusiasm and interest for my thesis and its content even though his research interests lie in different areas. André van Cleeff for his unabated interest, sharp insights and unique views on every part of my work. Wim Kegel and Wim Geurts from Logius for giving their valuable time and extremely relevant views on the strengths and weaknesses of not only my architectural proposals, but also the thesis itself. Haydar Cimen of KPN for his time, feedback and for convincing me of the relevance of my research. Geert-jan Laanstra for his invaluable help with a myriad of practical issues and the implementation of my prototype. And finally Vera Smulders, for putting up with me through this entire process. Thank you all.

Roeland Kegel

Contents

Chapter 1

Introduction

The world has seen many changes in the last few decades: by and large, each of these changes has transformed it to a faster paced, more connected place than it was before. From mobile phones to the internet and our increasing reliance on it, it becomes harder and harder to keep up with all the information that comes our way. As the amount of services that are offered digitally multiply, so does crime adapt to take advantage of new vulnerabilities created by such a change. It has been shown on many occasions that the end user of these services is one of the weaker links in the security chain[19]. This is unsurprising since the amount of information requiring a user's attention can become overwhelming in this day and age. Phishing, scamming and other types of cybercrime typically target this human link of the chain. While there are many good ways for enterprises to defend centralised data repositories (varying from social, to software, to hardware based solutions, see also chapter 2), these are often not suitable for end-users, as they lack the expertise and resources available in an enterprise context. Threats to a person that have inspired this research include:

Social engineering Users may be ill equipped to guard their own information. As such, malicious individuals target (often with great success[25]) the end-user itself with attacks such as phishing[3].

Spyware/malware Designed to infect a device in order to create/exploit vulnerabilities in such devices, spyware and malware can be used to gather confidential user information. Spyware is incredibly widespread, with not enough users aware of the dangers it represent[24].

Open Source Intelligence Not all information that is associated with a user is found on the user's devices: some of it can be found publicly, by searching and correlating many data sources. Open Source Intelligence, also known as OSINT is emerging as a new way to gather intelligence on people. Though there are many positive things to be said about the emergence of OSINT[11], the need to educate the user about the danger of giving out information to unknown parties does also increase in urgency.

1.1 Research goals and context

The last few years, there has been a general trend towards the development of agent technology: in computer science, software agents are systems that act on behalf of the user, often containing a form of learning ability in order to enhance their effectiveness over time. The field of computer science is not an exception, as other fields of research also use agent technology in their research[20]. Large companies such as IBM incorporate personal agent technology in their vision of security in the future[1].
In this context, the University of Twente has started a research project that aims to deliver a *Personal Information Security Assistant (PISA)*. This PISA aims to enhance the information security of end-users via agent technology, containing educative, motivating and machine-learning elements. This thesis presents an initial design for such an assistant, forming the basis for future work in this context. PISA aims to be user centric, available on every device that users own and advising them on security matters where appropriate. The PISA design presented in this thesis aims to take into account (and address) the following challenges:

Multiple devices When one wants to safeguard the information of a user, one has to acknowledge the fact that today's users have multiple devices, possibly containing sensitive information on any one of them. Examples include a user's smartphone, laptop, desktop computer and assorted other devices such as smart-TV systems. Users may store information or access digital services on any of these devices; this complicates matters for PISA, as it needs to be able to have ”eyes and ears” on every device the user interacts with.

Many roles When considering what information should be kept secure and what is public, one has to consider what role the user is fulfilling at that point in time. A parent might wish to keep adult content away from children; a customer might only be concerned for the safety of his billing account; while an employee might have sensitive information that must be kept secure. PISA's users may cycle through an arbitrary amount of these roles during a typical day. All of these roles

[1] http://www.research.ibm.com/cognitive-computing/machine-learning-applications/identity-theft-protection.shtml#fbid=nlvRINFOMxu. Retrieved 16-8-2014.

have different priorities when it comes to the security versus functionality tradeoff, which leads to different functionality requirements. The PISA system needs to recognise these needs (and shifts in them) and adjust functionality accordingly.

Many contexts The physical context in which a user device finds itself directly influences how secure such a device is: a user connecting via a secure corporate network will have more defenses against external probing attacks than e.g. the same user connecting via a public, unsecured Wi-Fi connection. In addition, such contexts might themselves create functionality/security requirements to which PISA needs to adhere. PISA needs to take these variable contexts into account as best as possible in order to provide the correct tradeoff of security versus functionality.

1.2 Thesis structure

This thesis covers the design process and validation of several proposed architectures for a PISA system. First, existing solutions are covered in chapter 2, both to illustrate the current solutions to the problem and to illustrate why PISA is better suited to end-user security. Specific research questions are then proposed in chapter 3 in order to formalize the scope of this thesis and define the specific questions to be answered. A stakeholder analysis is performed in chapter 4 using a use scenario to place the PISA system in a specific context. The identified stakeholders and their goals are introduced and classified according to the theory of stakeholder salience by Mitchell et al[18]. Requirements are elicited using these goals in chapter 5. Four alternative architectures are then proposed and discussed in chapter 6. The validation of these architectures is performed in chapter 7, divided in three parts: a traceability analysis based on the goals and requirements derived in previous chapters; a prototype implementation of a specific architecture; and a discussion of the prototype and the proposed architectures by industry professionals. Finally, conclusions and avenues for future work are explored in chapter 8.

Chapter 2

Existing Solutions

2.1 Earlier attempts at improving end-user security

No research or development project exists in a vacuum, and the PISA project is no different. This chapter covers existing solutions that improve end-user security and reasoning for why these solutions are as of yet insufficient. Combined with the research context and motivation section (1.1), this chapter illustrates the added value of a PISA tool in today's world.

2.1.1 Awareness and training programs

These are training programs designed explicitly to educate the users on how to secure their devices. The need for such programs has been demonstrated in the past, especially when considering the current multi-device environment that a user now generally has [12]. There are several issues with this approach, however. Apart from the fact that most of these trainings are dependent on an organizational context (i.e., the employer provides the training to its employees), an awareness program does not necessarily make users act. For example, to inform citizens of the importance of strong passwords is one thing, but to make them actually create strong passwords for all their user accounts is much more difficult [27]. Examples of these awareness training programs can be found in many places/sites[1]. Additionally, standards exist to define these training programs [17].

2.1.2 Legal solutions

Many legal solutions are being, and have been proposed, for example to mandate minimal security precautions. However technological developments simply outpace legislation [14] and global corporations can store their data at a location where the least restrictions apply. Though the myriad of laws and regulations regarding privacy are hard to get a handle on, sites and blogs exist to stay on top of these developments as they occur [2].

2.1.3 Managed security services

These services include, but are not limited to automated backup systems (available as the iCloud service on iOS devices, delivered as an integrated service in Windows operating systems, or available as commercial product) [3], virus filtering services from ISPs, end-point security solutions embedded in corporate infrastructure [4] and even operational intelligence gathering systems such as SPLUNK [5]. These solutions reduce the workload for end-users, obsoleting the need for technical know-how at the user end. Unfortunately, they also come at the cost of user awareness and customization, since most services have a one-size-fits-all approach. Additionally, and these services can break down themselves, of which the user will then be unaware [6].

2.1.4 Tools

Perhaps the most common and/or well-known methods of securing end-point devices are the tools associated with it currently: antivirus programs such as Symantec Antivirus [6] and software-based firewall programs such as the built-in firewall available on the Windows operating system. In addition to these tools, however, many other solutions exist to improve the user's security, ranging from relatively easy-to-use backup utilities to tools requiring expert technical knowledge such as encryption utilities[7], intrusion detection systems [8] etc. All these tools are hard to keep track of: because there are so many, it is hard for consumers with little time or experience to select an effective set of tools to secure their devices. As

[1] e.g. www.securingthehuman.org. Retrieved 16-8-2014.

[2] e.g. www.huntonprivacyblog.com. Retrieved 16-8-2014.

[3] e.g. http://www.techradar.com/news/software/applications/best-free-backup-software-11-programs-we-recommend-1137924. Retrieved 16-8-2014.

[4] http://www.kaspersky.com/business-security/endpoint-advanced. Retrieved 16-8-2014.

[5] see: www.splunk.com. Retrieved 16-8-2014.

[6] http://www.symantec.com/. Retrieved 16-8-2014.

[7] http://en.wikipedia.org/wiki/BitLocker. Retrieved 16-8-2014.

[8] http://www.windowsecurity.com/software/Intrusion-Detection/. Retrieved 16-8-2014.

an alternative, complete technical solutions such as a secure operating system [16] have been proposed in the past. Unfortunately, these are not economically feasible because they require new software and/or hardware.

2.1.5 Agent technology

Related, but not limited to the field of security is agent technology: localised instances and tools that monitor and secure end-point devices. Where end-point security systems are typically aimed at enterprise deployment, PISA aims for a personal version of this security mechanism to help the user secure its devices. An active area of research [2] in many disciplines[20], PISA uses agent technology to realise a broader scope of defensive measures in order to secure the user in an environment with multiple devices, many roles and changeable contexts.

2.2 The PCSO

As a precursor to the PISA project, a tool was developed[23] to assist users of the social media website Facebook to manage their privacy policies. This tool, the Personal Chief Security Officer (PCSO), combines elements of education, risk management and communication between users to set up a network of trusted friends, along with the management of one's own privacy settings.

2.2.1 Usage

The PCSO works by integrating it as a Facebook application. Users link the PCSO to their Facebook profile and answer a set of questions pertaining to their risk appetite, as if they were doing a lightweight risk assessment on their personal lives. This assessment is subsequently used to both create a policy that is shareable with other users of the tool and to suggest a series of changes to the user's profile privacy settings. The policy created is sent to people the user wants to be friends with, giving them an overview of the demands and requests associated with being a friend of the PSCO user (e.g., "don't tag me in any photos"). A similar policy is sent as a response; when both parties accept, they will be friends with a degree of insurance that their privacy needs will be respected.

2.2.2 Design

The PCSO consists of 3 elements:

The Facebook Application Runs on the Facebook servers. This contains the program logic and interface of the PCSO.

The PCSO Server A server containing the database which houses the information needed for the PCSO to operate: mail addresses and the policy settings are stored here.

Shared Risk Repository A server that does not contain any personal information, only templates for policies and the information needed to do the risk assessment on the client side. This repository is updated by security experts.

2.2.3 Lessons learned

Though the creators state that the majority of the test subjects found the tool easy to use (61.9%) and would continue to use the tool (76.19%), there are several studies that seem to support a less user-involved approach to security as being more successful. Considering the majority of users of such a tool does not have extensive motivation and/or technical knowledge, Petty & Cacioppo's Elaboration Likelihood Model[22] suggests offering the user less information and more action as being a more persuasive manner of communication. Additionally, several design principles seem to support a non-interrupting form (i.e., do not hinder the user's ability to continue working on other things) as being more effective when it comes to persuading users to adopt technology: the Technology Acceptance Model [5] and its subsequent iterations [26] define ease of use as one of the major factors deciding technology adoption, while the Persuasive Systems Design Model proposed by Oinas-Kukkonen and Harjumaa [21] offers reduction ("making a task easier for the user to complete") as major design principle when structuring persuasive systems. As a result, while keeping in mind the need for users to define a risk profile that accurately models their risk appetite, a set of requirements is derived from this prototype and included in the functional requirements section, 5.3.

Chapter 3

Research Questions

This chapter formalises and justifies the aims and scope of the research presented in this thesis, beginning with the research questions that this thesis aims to answer and ending with an explanation of the direction and extent to which the prototype's functionality was considered.

3.1 Summary of research questions

The primary aim of this thesis is to design and validate an architectural proposal for the PISA system. To do this, the following research questions have been defined:

1. *What are the stakeholders and goals of the PISA system?*

2. *What requirements can be used to describe the PISA system's goals?*

3. *What design alternatives exist for the PISA system?*

4. *How well do these design alternatives fulfill PISA's goals?*

  4.1 *Which architectural alternative best fulfills the elicited requirements?*

  4.2 *How well does an implementation based on such an architecture fulfill the elicited requirements?*

  4.3 *What is the opinion of industry professionals on these architectures?*

Question 1 has been defined to visualise the context in which the PISA system may function. It serves as a starting point from which the next questions may be answered. This question is answered in chapter 4. Based on these stakeholders, their relative importance and their goals, question 2 can be answered. This question is answered in chapter 5 and yields information necessary to validate the architectures proposed later in the thesis. Question 3 needs to be answered to implement a prototype. The answer to this question is given in chapter 6. Finally, as a validation of all that has gone before, question 4 determines the relevance of the findings in the previous chapters and the direction of future work. Answers to questions 4.1, 4.2 and 4.3 are found in the associated sections of chapter 7: 7.1, 7.2 and 7.3.

Questions 1 and 4 are *knowledge* questions (i.e., gather and analyse *existing* knowledge), while questions 2 and 3 are *design* questions (i.e., these define *new* knowledge).

3.2 Justification of scope

Considering this thesis deals with the creation of a prototype designed to elicit knowledge and requirements for subsequent refinements of the concept of a PISA system, several restrictions to the scope of this research apply, in order to keep to a realistic design and development schedule. These restrictions have consequences w.r.t. the choice of architecture in chapter 6, and as such need to be taken into account. These are as follows:

The Browsing Scenario : The prototype used for validation deals specifically with the scenario involving browsing behavior. This means that other threats requiring specialised parts of the PISA system (such as intrusion detection, advanced human interaction/education, inter-agent communication and other considerations) are explicitly left as future work.

Single Device : In the interests of time and testing considerations, the prototype involves only a single device. While multiple agents/devices are accounted for in the architecture, the implementation is restricted to this due to time and resource contraints within this master's project.

Chapter 4

PISA Stakeholders and Goals

To design a system that conforms to the needs of its users, one has to identify both the stakeholders of a system, as well as their goals when using it. To elicit these stakeholders, one needs to consider the socio-technical context in which the PISA system will operate. This is not a trivial task, since the design of the PISA system eventually needs to encompass a variety of situations and environments (ranging from securing devices in a corporate environment to keeping a home computer safe). Each of these situations and environments may contain different sets of stakeholders and goals, leading to different requirements of the system. Though the architectures proposed in chapter 6 aim to be largely situation agnostic, i.e. be able to handle a large selection of these, to contain the scope of this research a scenario is presented in order to elicit a *specific* set of stakeholders for the system.

This chapter first illustrates a use scenario which is used to create an illustration of the socio-technical context in which the PISA system will function. A list of stakeholders derived from this context is then presented and a summary of their goals is given. These goals are derived from assumptions based on the assumed design challenges defined in section 1.1. These stakeholders are then classified using the terminology proposed by Mitchell et al. [18]. At the end of this chapter, a list of goals is available prioritised by assumed relevance to the project based on the stakeholder classification.

4.1 Use scenario: the external collaborator

The following is a scenario designed to illustrate a typical situation in which the PISA system might function. It has 2 *personas* (archetypical users) and illustrates some of the day-to-day interaction of PISA with its users.

4.1.1 Personas used

James is a married man of 30 years and has two children. He is a consultant with a contract with a company. He works mostly at home, and so has access to the company's website via an application on his phone and home computer. He is a civil engineer and is not very tech-savvy.

Abby is 13 years old and the daughter of James. She occasionally uses the computer that James uses for his work to play games and browse the internet.

Sam is a security specialist working at the company that maintains the centralized servers that PISA uses. He is responsible for identifying current threats and acting upon them by pushing updates to the PISA applications on a user's devices. He is an expert on current security affairs and risk assessment, able to make snap-judgements on actions that need to be taken in order to secure a user's devices and information when needed.

4.1.2 A day in the life of James

James starts up his work computer in the morning and performs a few tasks related to the company with which he has a contract. He leaves the computer on while he gets coffee, but is called away to work unexpectedly, leaving his computer unlocked. Abby finds the computer unlocked and proceeds to browse the internet on James' account. PISA monitors the browsing activity and detects an abnormal pattern (i.e., non-work related browsing on an account that is used by James). It sends a warning to James' phone, alerting him his account is being used in an unusual manner. James then elects to have PISA log out Abby from James' account to prevent any mishaps.

That same morning, Sam has identified a malicious infection on a well-known news site. Knowing that there's a reasonable chance that some PISA users will be infected by visiting this site, he issues a warning in the form of a new *policy*, a rule linking an event occurring on PISA-protected devices to an action. In this case, when a user tries to visit the site, a message is generated by PISA to inform the user that the site is temporarily unsafe, offering several alternatives to redirect the user in the meantime. James happens to be a reader of this website and encounters the warning when he tries to access this site at work. He acknowledges the risk and visits the top alternative offered by PISA instead for his news-itch.

Later that day, James is writing an email containing work related information to a diverse group of people, most of whom do not know each other. He has failed to see, however, that he has not used the BCC field, instead writing all the addresses in the *To:* field. PISA detects this and when James clicks the send button, it intervenes by asking for a confirmation to send this mail without moving some email addresses to the BCC field. James realises his mistake and asks PISA to correct the mail before actually sending it.

In the evening, James uses his computer for private affairs by browsing an online auction site. He does this because he has received a mail that several items he is interested in are up for auction right now. He tries to log in to his account on the site. PISA, however, detects that this site does not have a valid certificate verifying its authenticity. It then blocks James from entering his password, advising him that there is a high likelihood that he has stumbled upon a phishing site that is trying to get his account details. James scrutinizes the page closely, confirms that something is wrong and leaves the site instead of logging in.

The set of examples above illustrate several ways in which the PISA might interact with its environment. When the (rather eventful) day of James is considered, one can identify several elements in PISA's context. Of these elements, a subset can be used as stakeholders for the elicitation of requirements later in the thesis.

4.1.3 Identifying the context

PISA instances Specific PISA applications that interact with the user. This is part of the system to be developed. The other elements specified below are part of its context.

Primary user An owner of one or more devices on which PISA runs. the PISA is charged with protecting this user.

Primary user's devices Devices that belong to the primary user and run a PISA application.

Confidential information Sensitive information that PISA needs to protect. Could be e.g. files on devices or personal information of the primary user.

Secondary user A user that uses, but does not own the devices on which PISA applications run.

Centralized PISA server A server application that PISA applications contact in order to get updated policies governing what it needs to look out for/act upon. Part of the system to be developed.

Policy database A database that is used by the PISA server to store all policies of all PISA users. Part of the system to be developed.

Security expert Responsible for updating the policy database based on current events and developments.

Malicious user Any person without legal access to the previously defined confidential information that tries to gain this information by accessing the primary user's devices or tricking the primary user.

Information owner Any person with a legal claim to confidential information in possession of the primary user.

A diagram depicting the relationships of these elements is given in figure 4.1.

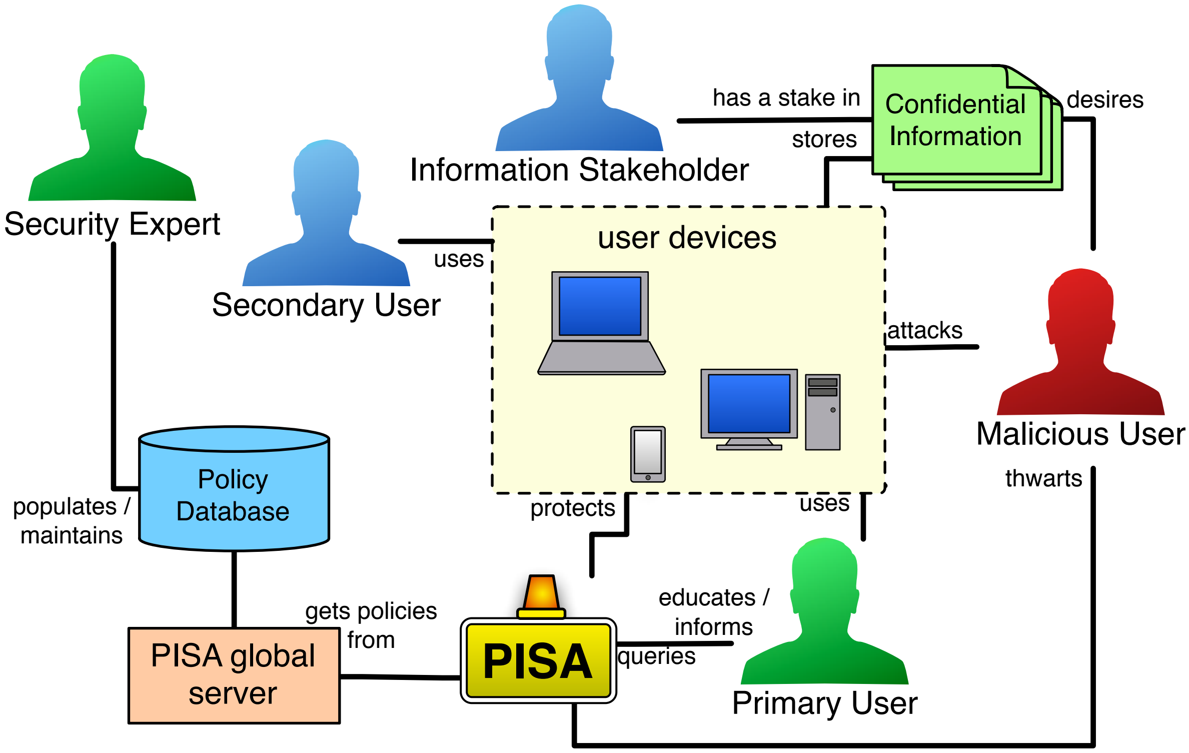

Figure 4.1: An illustration depicting the context as described in the use scenario of 4.1

4.2 Stakeholder theory

In order to prioritize stakeholders, the concepts and associated entities within the PISA system are first mapped to the concepts proposed by Mitchell et al. This yields stakeholders that are classified using an indicator known as *salience*. In order to adapt the theory proposed by Mitchell et al to the context of a system such as PISA (rather than an organisation), we define two terms presented in the theory in the following manner:

Organisation The primary object with a relation to the stakeholders. This traditionally is the firm/organization for which the stakeholder analysis is performed, since the theory of stakeholder identification comes from the management sciences. In this context, however, we define the organization itself as the PISA instances, the centralized PISA server and the policy database (see the previous section for a definition of terms). These three are chosen since they encompass the system that needs to be designed whereas the other elements defined above are existing elements of the environment.

Stakes The relevant elements to stakeholders in the identification. In the case of the PISA system, this concerns the primary user's devices and the confidential information as defined above. The latter element is divided in a category that is relevant to the information owner and one that is not (i.e., the information in question is the sole property of the primary user). This distinction is relevant for the legitimacy/urgency claims, see below.

Mitchell et al. define 3 dimensions in which stakeholders can be categorised: legitimacy, power and urgency:

Legitimacy Defined as *"a generalized perception or assumption that the actions of an entity are desirable, proper, or appropriate within some socially constructed system of norms, values, beliefs, and definitions"*, which Mitchell et al define on the societal, organisational and individual level.

Power Defined as *"the extent [a party] has or can gain access to coercive, utilitarian, or normative means, to impose its will in the relationship."* Coercive, utilitarian and normative means are physical (for example, a gun), material (money, goods) and symbolic (popularity, prestige, esteem…) respectively.

Urgency The "dynamic" aspect of stakeholder identification/classification, symbolizing both the criticality to the claimant and the degree to which the claim is time-sensitive. In this static stakeholder identification context, we disregard the time-sensitive nature of urgency, but continue to gauge the criticality to the claimant as a relevant dimension to the analysis.

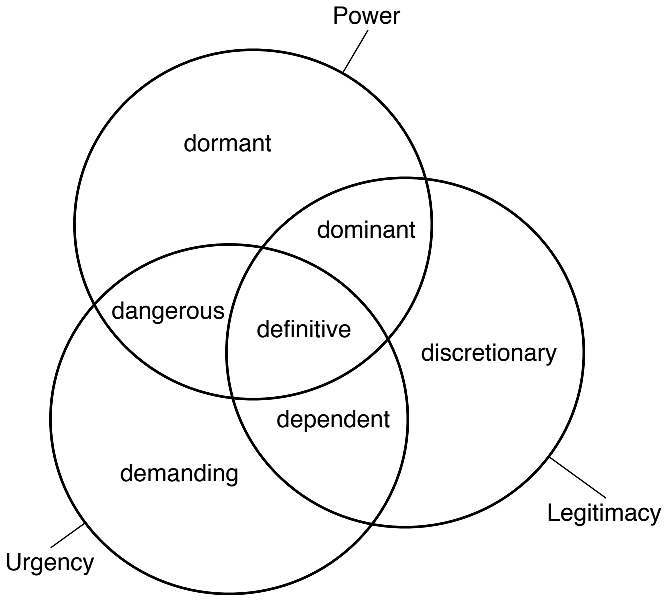

Figure 4.2: An illustration of the different categories of stakeholders available in the classification of Mitchell et al.

Finally, these are categorised according to the amount of these elements existent in these stakeholders: the more elements are present, the higher the salience of the stakeholder and its demands. Conversely, if none of the above attributes are present in actors (the actor has no claim, no power and no urgency over the stakes or system), they are not labeled as a stakeholder. The types of stakeholders that exist according to this classification are:

Latent Those stakeholders that possess only one of the three attributes.

> **Dormant** Stakeholders with power, but no legitimate or urgent claim. These have the means, but not the motivation to use their influence.
>
> **Demanding** Stakeholders with urgency (i.e., they have an issue they perceive as time-sensitive and/or critical to them) but no real power or legitimate claim. They may want things, but the system has no obligation to provide it.
>
> **Discretionary** Stakeholders with legitimacy, but no power to press this claim or urgent cause to do so. There is no need for the system to engage in an active relationship with these stakeholders, but a defense can be made for doing so. An example in the PISA system would be the PISA system protecting information of individuals that are not the users of its devices.

Expectant Stakeholders that possess two out of the three attributes.

> **Dangerous** Stakeholders with power and urgency. While these stakeholders do not have a legitimate claim, they do have power and a perceived cause to use it, and as such are dangerous to the system and the stakes it is protecting. Malicious users are a typical example of this category.
>
> **Dominant** Stakeholders with legitimacy and power. Combining into something that is described by some as authority, these stakeholders can and have the right to influence the system and its stakes.
>
> **Dependent** Stakeholders with urgency and legitimacy. Lacking the power to influence anything, these stakeholders do have a legitimate claim and a reason to make this claim known. They are dependent on others to realize their requests, however.

Definitive Stakeholders that possess all three attributes. These users are the primary stakeholders of a system; in the case of the PISA system, a prime example of a definitive stakeholder is the primary user.

4.3 Stakeholder classification

Combining the context as defined in chapter 1 and the terminology presented above, the stakeholders of the PISA system can be identified and classified. The terminology proposed by Alexander et al[1] is used in order to place these stakeholders in a relatively familiar frame of reference:

Primary user Alexander et al's terminology lists several possible classes for this stakeholder. In this context, the primary user is treated as a normal operator. The primary user is considered to be the owner of the device and the primary user of the PISA system. Risk profiles and roles are primarily derived from this person. These stakeholders possess *legitimacy* because of their ownership of the devices and information, *power* through control of these devices (and thus the PISA system) and *urgency* through the need to secure the private information and devices entrusted into its care. Thus, the primary user is a **definitive stakeholder**.

Secondary user Defined by Alexander et al as either a functional beneficiary (PISA secures the behavior of secondary users too), normal operator (the system is interacted with by the user) or negative stakeholder (the added hassle of interacting with the PISA system may be considered as a hindrance by secondary users). A person that uses the primary user's device from time to time (with the primary user's permission; otherwise see *malicious user*). Examples include family members or colleagues that borrow the device (e.g. for a meeting, to look something up on the internet). This class of user might exhibit a large range of behaviors; as such, care has to be taken that PISA is able to handle a wide variety of behaviors, adapting its risk profile accordingly. A secondary user possesses *power* through (given) access to the device, but has no legitimacy or urgency w.r.t. the stakes listed above, and as such is classified as a **dormant stakeholder**.

Malicious user Defined by Alexander et al as a negative stakeholder. Any security system has its detractors, most notably malicious users out to compromise the security (be it either confidentiality, integrity or availability) of the device on which the security system operates. PISA must not only be able to guide the user towards safer behavior, but also be able to detect and thwart malicious users from gaining access to/compromising the system. Malicious users are negative stakeholders with a varying degree of *power* and *urgency* over the stakes presented above. As such they belong mainly in the **dangerous stakeholder** category. Mitchell et al. does not explicitly recognise negative stakeholders, but the concept of salience can just as easily be applied to thwarting claims/demands as a form of dealing with them.

Security expert Defined by Alexander et al as operational support and/or maintenance operator (depending on which task of the security expert is considered). A person that assesses risk to generic stereotype users and populates the risk repository with possible threats to the system. Separate experts may exist to perform threat assessments, policy definitions and categories of policies (risk profiles). While one of the design goals of PISA is to adapt to a user's needs, creating a personalised risk profile that matches the user's risk appetite, the initial setup will require a categorization of policies into predefined profiles a user can use when no usage data is available. Security experts are needed to define these profiles. They have *power* over the PISA system, but no legitimate claim to the stake, nor an urgent need. As such, these stakeholders fall under the **dormant stakeholder** category.

Information Stakeholder Defined by Alexander et al as functional beneficiary. When a user's device contains confidential information related to someone else, an information stakeholder becomes interested in the security of the device in question. As such, an information stakeholder might need to be given access to the security status of the system as seen by PISA. There is a degree of *legitimacy* to their claim, as well as a sense of *urgency* when it concerns sensitive confidential information, but since the devices owned by the primary user, no power can be exercised by these stakeholders. As such, they are **dependent stakeholders**.

This list of stakeholders should not be viewed as a comprehensive list when considering the PISA system, but rather as a subset that could be identified by considering the use scenario presented earlier in this chapter. It should also be noted that a combination of stakeholders is possible (i.e., a secondary user could also be an information stakeholder). This has consequences for stakeholder salience, but a consideration of the possible permutations of user combinations is considered to be outside the scope of the current research.

4.4 Goals

In this section, goals are listed and associated with stakeholders. These goals have been defined by expert discussion (see chapter 7) and consulting literature[15], as well as considering the goals of the PISA project (See section 1.1). This list of goals can then subsequently be ordered by *salience* as defined by Mitchell et al and potentially used to prioritise requirements derived therefrom.

4.4.1 Stakeholder goals

Each of these stakeholders above have goals with regards to the system, the stakes and its usage. Below, we summarise the goals explicitly associated with and derived from the stakeholders listed above.

Primary user

A user's priorities may differ, but the following elements are assumed to be present in one form or another:

- **Securing** the user's devices, making them less vulnerable to digital attack;
- Being able to perform their tasks **without impediment**: the PISA system should endeavor to forbid a user as little as possible;
- Being able to perform their tasks **without interruption**: the PISA system should interact with users on *their* terms;
- **Providing intelligence** to the user, creating awareness of current flaws in the user's security status;
- The need to be **educated** in the area of digital security, improving the safety and security of the user's devices and information;
- The need to be **motivated** to act in a more secure manner, analogous to being reminded to act in a healthy manner.

The goals above are listed from (assumed) high to low priority in an average user of the PISA tool.

Secondary user

When comparing goals of the secondary user with that of the primary user, the goals **no impediments** and **no interruptions** still apply, but other concerns are mostly irrelevant. It may be the primary user or developer's aim to educate the user, but this is not a priority for the secondary user itself.

Malicious user

Considering this user is a negative stakeholder, the note applies that PISA's goal is to address the stakeholder's goals by *thwarting* them. As such, **security**, **education** and **motivation** are the three primary concerns of the PISA system w.r.t. a malicious user, as all of these system goals provide a measure of increased security to the possible targets of its attacks.

Security expert

This stakeholder is mainly interested in:

- A **scalable and usable policy format** for the risk repository and policy database, as policies are the main form of interaction of this stakeholder with the system;
- A **light-weight risk assessment** method associated with PISA to aid risk profile establishment, as creating initial risk profiles is part of this stakeholder's responsibilites;
- **Machine-learning** in its policies and risk profiles, as this aids the stakeholder in maintaining the system (which is part of its responsibilities).

Information stakeholder

An employer has much the same security concerns as a contact of the primary user (i.e., the primary user's devices contain private information belonging to the stakeholder). Much like with enterprise end-point security solutions, however, information stakeholders may want a degree of *assurance* that their information and the devices on which it is stored are secure. As such, the following goals are discerned w.r.t. the primary user's information stakeholders:

- **Security** of those of the primary user's devices that contain relevant confidential information;
- **Education** of the primary user in question;
- **Motivation** of the primary user to act in a secure manner;
- An **assurance mechanism** to give insight in the security status of the primary user's devices.

4.4.2 Project goals

An implicit stakeholder in most systems' development is the developer of the system, especially when it is constructed in the context of a research project. This is acknowledged in Alexander et al's work by defining the developer stakeholder. This stakeholder is added in this stakeholder analysis due to the context of the development of this prototype (i.e., research). Following Mitchell et al's methodology, the developer/researcher behind this project is considered to be a stakeholder with the ability to influence the PISA system (*power*), with *urgency* derived from the need for scientific data collection. Legitimate claims cannot be made to the stakes involved and as such, the developer/researcher is classified as a **dangerous** stakeholder. The following goals can be derived by considering the PISA project's goals:

- **Educating** & **motivating** the user w.r.t. Security. A primary goal of the project, this is considered to be only way to affect structural behavioral changes in a primary user;
- Providing an **API**/mechanism for security software to communicate and cooperate. In order to differentiate the PISA from existing technology, a coordinating function is considered desirable to attain synergy between different measures protecting a user's devices;
- Adding a **machine learning** component to risk profiles/policies. These elements are considered desirable to add relevance to the project on a scientific level;
- Providing **assurance** for third parties that a user is secure. Also part of the project's aims, this goal is added for much the same reason as a cooperation mechanism;
- Providing **lightweight risk-assessment** methods for the user. While risk assessment methodologies exist, it is considered relevant to create and/or test existing light-weight versions in order to be able to apply this to end-users, improving their security.

4.5 Conclusions

In this chapter, a list of stakeholders has been generated and classified according to salience. In order to use this in the next chapter when considering requirements and their relative importance, a prioritisation of goals according to stakeholder types is in order. As such, we use salience in the following way: all goals with a stakeholder of higher salience are considered to be relatively more important than those with a lower level of salience. As such, any goal of a definitive stakeholder is more important than any goal that has no definitive stakeholder associated with it. When both goals have stakeholders of similar salience levels, the amount of such stakeholders is considered to be a tiebreaker (i.e., when considering goal A with 1 definitive and 2 expectant stakeholders, and goal B with 2 definitive and no expectant stakeholders, goal B would be considered as having a higher priority).

Using the combination of stakeholders ordered by salience and internal assumed order of importance for stakeholder goals, a prioritisation of goals is given in table 4.1. One can see here that the education, motivation and securing of devices are the key goals of the PISA tool, as expected. The researcher/developer is included as an expectant stakeholder, leading to the prioritization of education and motivation over the security of devices. The next chapter uses this ordered listing of goals to provide traceability to the requirements of the PISA prototype.

Goal	description	Definitive	Expectant	Latent
G01	Provide education	1	3	
G02	Provide motivation	1	3	
G03	Secure devices	1	2	
G04	Provide an assurance mechanism	1	1	
G05	No impediments to digital activities	1		1
G06	No interruptions during digital activities	1		1
G07	Provide intelligence	1		
G08	Contain a machine learning component		1	1
G09	Provide an API for cooperation		1	
G10	Provide a risk assessment method			1
G11	Provide a usable & scalable policy format			1

Table 4.1: An overview of stakeholder goals sorted by number of stakeholders and their level of salience

Chapter 5

Requirements

In this chapter, the requirements derived for the PISA tool are discussed. This chapter first provides a section detailing qualitative requirements and functional requirements. An overview and discussion of the discovered requirements is given in section 5.3.1. The chapter then concludes with a validation of the requirements listed by providing a traceability matrix between goals and listed requirements. This set of requirements is used in the following chapters for validation of the different proposed architectures.

5.1 Definition of PISA's requirements

The requirements listed in this section are divided into the qualitative and functional categories. All of these requirements were elicited by considering the PISA project's aims, its predecessor (the PCSO, see 2.2) and the goals derived in the previous chapter.

5.2 Qualitative requirements

These requirements are derived from general qualitative principles in software engineering (such as usability, scalability etc). A reasoning for each of these requirements follows after the specification.

R01 *Portability: the PISA's architecture shall be designed so that it can be converted to run on any device*
reasoning: In order to protect a user, the PISA should have a presence on every device the user interacts with. Though a case can be made that only devices that store personal information need to be protected, users may enter personal information from memory at any location. As such, ideally PISA needs to be able to run on every device that user can conceivably interact with.

R02 *Maintainability: the PISA shall endeavour to use a minimum of technologies*
reasoning: Due to the diverse nature of interactions the PISA has in a user's device ecosystem, it is easy for the amount of technologies used to achieve PISA's functionality to get out of hand. Care has to be taken that the use of every additional technology/language is a conscious decision in order to add functionality to the PISA, since maintainability decreases with each additional technology/language used.

R03 *Scalability: the PISA and its back-end (the policy and risk databases) shall be able to support an arbitrary amount of users and devices*
reasoning: With the goal of being a definitive security solution for end-users, scalability has to be taken into account when designing the PISA's architecture. Each additional user can mean an arbitrary amount of devices added to the system; PISA has to be able to cope with this possibly rapidly increasing number of elements.

R04 *Security: the PISA shall possess security mechanisms to secure its communications between components*
reasoning: The reasoning for this is twofold: as the PISA aims to provide an API for cooperating applications, care has to be taken that this public protocol is robust and cannot be exploited by malicious users (by creating a "cooperating" application that gathers information about the PISA system). Secondly, since the PISA handles and secures a large amount of a user's personal information, special care should be taken if/when any information related to the user is transmitted.

R05 *Extensibility: the PISA shall use an extensible API to ensure cooperation with a broad range of components is possible*
reasoning: The PISA cannot create the sheer amount of tools needed to comprehensibly protect a user: antivirus programs themselves are a very large industry and would require a exceedingly large effort to replicate. Instead, the PISA's added value lies in the cooperation between security solutions and the synergy that can be obtained therefrom. As such, the PISA framework should aim to incorporate the ability to communicate both with current and future technologies in its API.

R06 *Availability: the PISA shall feature self-contained elements so the PISA instance on a device can function when isolated from other user devices*
reasoning: When dealing with mobile devices, a network connection between all components is not always feasible. As such, components of the PISA framework should be able to operate in isolation whenever no connection can be established. This means, e.g., that an end-point of the PISA designed to take action should be able to monitor relevant events, lookup associated rules and take action accordingly *without* intervention from a centralised location whenever possible.

5.3 Functional requirements

This section lists functional requirements to which the PISA aims to adhere, with a reasoning and source attached to each of them. The listing is divided into top level requirements, with associated lower-level requirements classified under these categories. This naturally results in a function refinement tree as defined in literature by Wieringa[28]. This tree does not aim to be comprehensive, but rather aims to be a starting point for a more thorough evaluation of the system's requirements, see chapter 8. The function refinement tree of the listed requirements in this chapter is given as a point of reference below.

5.3.1 The function refinement tree

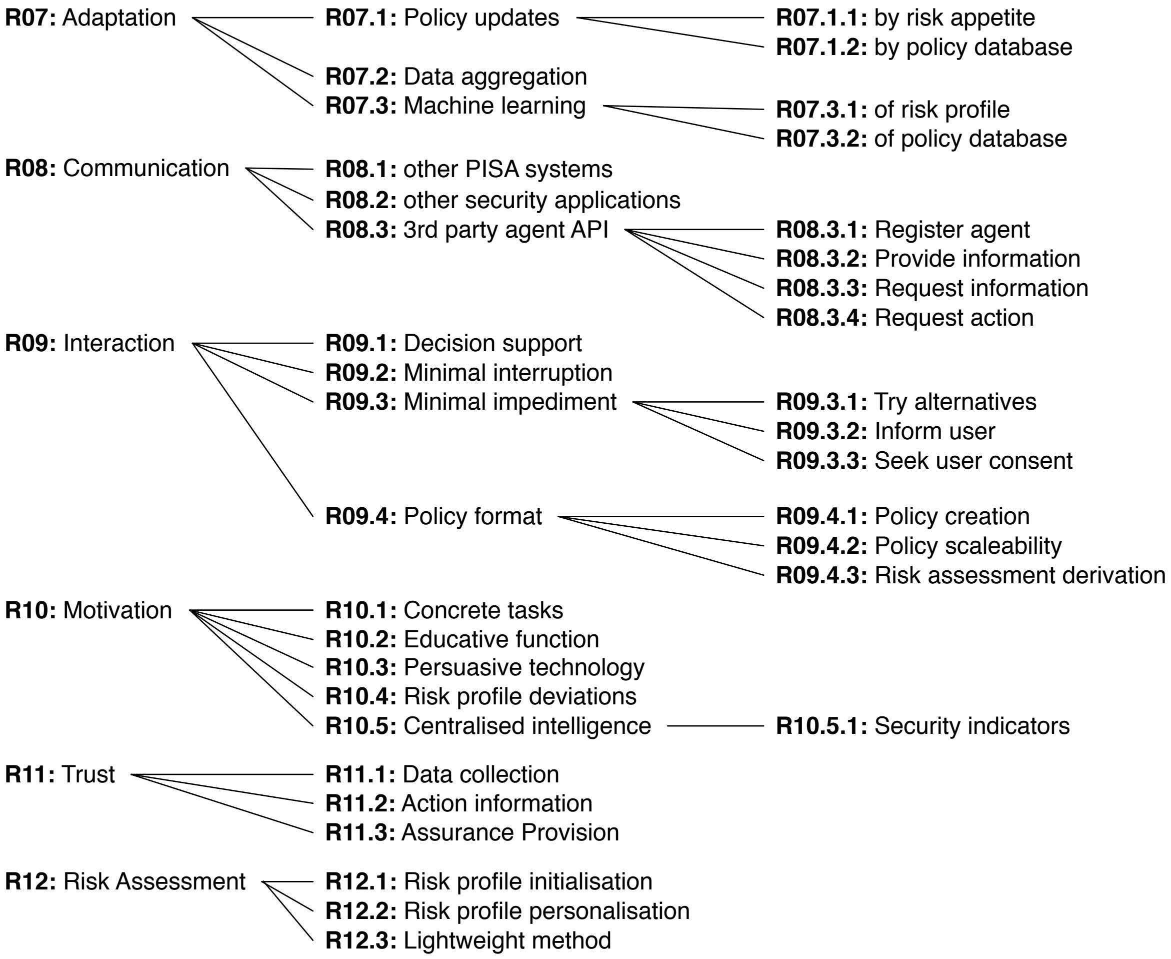
R07: Adaptation
R07.1: Policy updates
R07.1.1: by risk appetite
R07.1.2: by policy database
R07.2: Data aggregation
R07.3: Machine learning
R07.3.1: of risk profile
R07.3.2: of policy database
R08: Communication
R08.1: other PISA systems
R08.2: other security applications
R08.3: 3rd party agent API
R08.3.1: Register agent
R08.3.2: Provide information
R08.3.3: Request information
R08.3.4: Request action
R09: Interaction
R09.1: Decision support
R09.2: Minimal interruption
R09.3: Minimal impediment
R09.3.1: Try alternatives
R09.3.2: Inform user
R09.3.3: Seek user consent
R09.4: Policy format
R09.4.1: Policy creation
R09.4.2: Policy scaleability
R09.4.3: Risk assessment derivation
R10: Motivation
R10.1: Concrete tasks
R10.2: Educative function
R10.3: Persuasive technology
R10.4: Risk profile deviations
R10.5: Centralised intelligence
R10.5.1: Security indicators
R11: Trust
R11.1: Data collection
R11.2: Action information
R11.3: Assurance Provision
R12: Risk Assessment
R12.1: Risk profile initialisation
R12.2: Risk profile personalisation
R12.3: Lightweight method

Figure 5.1: A graph based representation of the functional requirements presented in this chapter

The functional requirements derived in the next section naturally fall into a pattern of refinement: sub-requirements are more detailed than their parents. As such, it is possible to categorize these requirements into a function refinement tree. A graph based representation can be seen in figure 5.1. This graph reveals that several areas require further definition before any implementation of the concept can be considered, whereas other areas are sufficiently defined to be clearly deliniated in modules and functions in an implementation. Areas such as machine learning and risk assessment are major categories in which specification needs to occur before any implementation fulfilling these requirements can be considered. In contrast, the communication and trust requirements can be implemented (with the notable exception of assurance provision, as this needs the security indicators as specified in R10.5.1). Research into persuasive technology has already been performed[15] and can be applied in the design. As such, R10: Motivation can be developed with comparatively little amounts of research.

5.3.2 Listing of functional requirements

R07 *The PISA shall contain a set of mechanisms by which it adapts to its environment and the times*

R07.1 *The PISA shall periodically update its policies to reflect changes in:*

R07.1.1 *The user's risk appetite*:
source: General useability: any profile can be improved by personalizing it to a specific user.
reasoning: Since the user's needs and patterns are both subject to change and not necessarily accurate with just the initial establishment of the risk profile, mechanisms need to be incorporated within the PISA to allow for changes in the user's risk profile based on exhibited behavior.

R07.1.2 *The policy database*:
source: The security goal.
reasoning: Security is not a static field, and as such, to maintain an adequate level of security, the PISA needs to account for new attack vectors and vulnerabilities. Updating the PISA's policy database will help the PISA tool stay competitive in the security field.

R07.2 *The PISA shall anonymize and aggregate security related information and gather it at a centralised point*
source: The machine learning goal.
reasoning: Collecting security related information and processing it can help the PISA detect patterns and changes in the behavior of a user. There are two benefits to this: first, this usage information can be used by the security analysts to detect new threats and vulnerabilities and formulate new risks/policies based on current data. Second, this usage information can be used to personalise the user's risk profile to better match the risk appetite that a user displays and/or notifies the PISA it has.

R07.3 *The PISA shall contain a component of machine learning in:*

R07.3.1 *Users' risk profiles to adapt to their risk appetite*
source: The machine learning and no interruptions goals. **reasoning:** This machine learning component will allow the PISA to learn the user's actual risk profile based on usage rather than the lightweight risk assessment method which is used in the initial setup of the user's risk profile. This minimizes the interaction (and thus interruptions) necessary with the user. It differs from R07.1.1 in that this explicitly is an automated process, whereas R07.1.1 could use user interaction.

R07.3.2 *Its policy database to reflect current threats to a user's devices*
source: The machine learning goal.
reasoning: Creating a repository that is capable of (largely) automated updates to its database in order to adequately protect the user's devices after the initial release of the PISA. This increases the user security of the system.

R08 *The PISA shall be able to cooperate with various parties to attain synergy in the defense of its devices and users*

R08.1 *The PISA shall be able to communicate with other user's PISA instances in order to establish a trust level*
source: The PCSO prototype (see section 2.2).
reasoning: In recognition of the added value of communicating security policies to friends in social networks, PISA should attempt to communicate with other instances of PISA to establish a trust level that can be used for further communications.

R08.2 *The PISA shall communicate between security solutions to secure the user's devices*
source: Qualitative requirement R06.
reasoning: One of the observations when considering earlier solutions is that creating a comprehensive end-user security solution is infeasible due to the diverse nature of the threats visited upon the system, as well as the amount of resources required to adequately develop certain aspects of the security solution (i.e., antivirus). As such, PISA aims to create synergy by taking a coordinating role between the different security applications on a user's devices.

R08.3 *The PISA shall provide a robust API for 3rd party applications to cooperate/communicate with the PISA to:*

R08.3.1 *(de-)Register as an agent for PISA*
source: The API goal.

reasoning: Since the PISA has a coordinating function between multiple applications, a (de-)registration process needs to exist to handle this dynamic set of agents communicating with the PISA.

R08.3.2 *Provide intelligence*
source: The API goal.
reasoning: Any agent communicating with the PISA should be able to offer salient information to the PISA without prompt in order to increase the amount of current intelligence available to the PISA and its cooperating applications.

R08.3.3 *Request intelligence*
source: The API goal.
reasoning: Since any security system is dependent on information available to it, a way needs to exist for the PISA to query cooperating applications (since, e.g., an extension in a browser will know more about the user's current browsing activity that any process that runs outside the browser).

R08.3.4 *Enable the PISA to take an action*
source: The API goal.
reasoning: If applications that give intelligence are considered to be the eyes and ears of the PISA, it still needs agents to be its hands. The PISA needs to be able to direct cooperating applications to take specific actions in order to secure the system (e.g., instructing a firewall to close a certain set of ports to block suspicious traffic).

R09 *The PISA shall contain features that minimize its disruptive effect and support easy interaction*

R09.1 *The PISA shall offer decision support when making security related decisions*
source: The motivation, education and security goals.
reasoning: According to the persuasion techniques of tunneling and reduction used by Fogg[9] and Oinas-Kukkonen et al.[21], offering decision support to users can lead to increased persuasive ability of the system (and, subsequently, a higher level of security).

R09.2 *The PISA shall endeavor to minimise the interruptions to a user's activities*
source: Human interface design literature.
reasoning: Johnston defines[13] one of the success factors of a security related human computer interface as *"the user is only made aware of [the security application] when necessary"*. On a similar note, it has been shown[10] that interruptions have a negative impact on user performance of current tasks. Therefor, to increase the usability of the system interruptions in the user's workflow have to be minimised.

R09.3 *The PISA shall not block a course of action without:*

R09.3.1 *Having tried reasonable alternatives*
source: The no impediments goal.
reasoning: For obvious reasons, blocking a course of action as being dangerous usually does not sit well with the user, as it is liable to impact his current activity. As such, the persuasive power of the PISA wanes rapidly if actions are forbidden without first considering other alternatives.

R09.3.2 *Informing the user*
source: Persuasive technology literature.
reasoning: According to Cialdini, a message or course of action is more persuasive if the reason behind the course of action is disclosed[4]. Several other sources support this conclusion, such as the principle of *verifiability* in Oinas-Kukkonen et al.'s persuasive systems design model[21] and the *visibility of system status* cited by Johnston[13].

R09.3.3 *Gaining consent from the user*
source: Persuasive technology literature.
reasoning: The reasoning behind this is twofold. First, gaining explicit consent from the user helps engender a form of trust by putting the user in a role of authority (i.e., the PISA respects the user's autonomy). Additionally, user agency is an important factor in the user's decision process, listed in the Reasoned Action Approach[8] as one of the three major constructs involved in the decision making process. Any action taken voluntarily has more power to become a structural behavior change than one taken involuntarily. As such, letting users come to the decision on their own has added educational value, influencing future attitudes to the action.

R09.4 *The PISA shall use a policy format that:*

R09.4.1 *Is easy to use w.r.t. creating new policies*
source: The scalable, usable policy format goal. **reasoning:** Any sizeable implementation of the PISA will involve a large amount of policies to govern actions taken and intelligence gathered by the PISA and its agents. As such, the mechanisms by which new policies are defined and stored need to be easy to use.

R09.4.2 *Is scalable*
source: The scalable, usable policy format goal.
reasoning: Much like the requirement above, not only does it need to be easy to manipulate the entries in the policy database, it also needs to be able to handle exceptionally large amounts of policies to govern every situation required to protect the diverse set of devices and situations the user might have/be in.

R09.4.3 *Can be derived from risk assessments*
source: The machine learning and scalable, usable policy format goals. **reasoning:** In order to automate any conversion from risk to policy, a transformation needs to be defined that automates whatever steps are possible in the conversion from risk to a policy that may include actions.

R10 *The PISA shall motivate its users w.r.t. information security*

R10.1 *The PISA shall give users concrete tasks they can undertake to increase their security*
source: the PCSO prototype (see section 2.2).
reasoning: One of the major factors in the usability of the PCSO was the concrete list of tasks and advantages that the initial risk profiling generated. The PISA system should also aim to give a list of concrete tasks for the user as a tool to mitigate digital threats.

R10.2 *The PISA shall have an educative function w.r.t. digital security*
source: The PCSO prototype (see section 2.2).
reasoning: In recognition of both the conclusions in the persuasive technologies research[15] and the perceived added value of illustrating the threats which the user faces when deciding a course of action, the PISA system should attempt to inform and educate the user on digital safety.

R10.3 *The PISA shall use design principles derived from research to structure information towards the user in order to persuade him/her*
source: The education and motivation goals.
reasoning: In order to have a user adopt a certain technology, certain approaches are more effective than others when communicating with the user. As such, structuring information in a specific way can get a user to behave in a more secure manner, whilst other types of communication can lead to the user eschewing the use of a technology altogether. As such, care has to be taken to use those principles and techniques that have proven to be effective by research in the past.

R10.4 *The PISA shall confront the user with their risk profile and deviations w.r.t. given risk appetite*
source: The education goal.
reasoning: Confronting users with their behavior enforces a process of reflection, forcing the user to either change their risk profile to match their appetite, or the other way around. According to Festinger's theory of cognitive dissonance [7], this realignment naturally happens. This principle can be harnessed by confronting the user with their current behavior as opposed to their earlier defined risk profile.

R10.5 *The PISA shall offer a centralised view of the user's device security status*
source: The centralised intelligence goal.
reasoning: Johnston defines[13] one of the important elements of a security application to be a centralised location that the user can use to be informed about the status of the system (see the *"Visibility of System Status"* principle. Keeping the user up to date will help the PISA to gain a level of trustworthiness. Additionally, this can be used to educate users to the current risks and vulnerabilities to which they and/or their devices are exposed.

R10.5.1 *The PISA shall define security characteristics to measure the security of a system*
source: A requirement necessitated by R10.5 and R11.3.
reasoning: Derived from both the securing devices and the centralised intelligence goal: to make any judgement over the security of a system, indicators have to be developed to measure this security in a quantifiable way. These indicators can then be used to convey centralised intelligence to users about their devices.

R11 *The PISA shall incorporate features that promote a high level of trust between the system and its users*

R11.1 *The PISA shall inform the user of the data it collects and/or stores*
source: The PCSO prototype (see section 2.2).
reasoning: Considering the PCSO server and its data collection aspects, the PISA system should inform the user both of what information is known to it, and what information it uses in the course of its operation.

R11.2 *The PISA shall inform the user of the actions it takes*
source: The education goal.
reasoning: According to the principles of persuasion of Cialdini[4], people are more inclined to acquiesce to a request if they know the reasons behind it. Informing the user of the PISA's decisions and the reasoning behind its requests will help establish trustworthiness[21] and increase the persuasive power of the system.

R11.3 *The PISA shall provide a mechanism for an assurance provider to query the security status of the user's devices*
source: The Assurance Provider goal.
reasoning: This requirement will allow the informational stakeholder defined previously to query the safety level of the confidential information to which he has a legal claim.

R12 *The PISA shall contain elements of established risk assessment methodologies to improve user security*

R12.1 *The PISA shall offer a minimalistic risk profile initiation*
source: The PCSO prototype (see section 2.2).
reasoning: PISA will require a degree of initial setup to establish a risk profile for the primary user. In recognition of research done on the persuasiveness of systems that require extensive interaction, however, this initial setup should require a minimal amount of effort to achieve.

R12.2 *The PISA shall personalise the user's risk profile*
source: The PCSO prototype (see section 2.2).
reasoning: As the initial risk profiling should be minimised, a degree of personalisation should take place during the period where PISA is active and protecting the user. This process should require a minimum amount of user interaction.

R12.3 *The PISA shall use/define a light-weight risk assessment method that a user can use to establish a risk profile*
source: The risk assessment method goal.
reasoning: This requirement will enable the PISA to perform the initial setup involved in establishing the perceived risk appetite according to the user.

5.4 Conclusions

As part of the ongoing validation process, to attain traceability between stakeholders and the architecture, traceability has to be defined between the goals and requirements listed in the section above. The second step in the overall traceability process is shown in table 5.1. If an entire category is mentioned, all related sub-requirements are considered to be relevant to the stakeholder goal (e.g., when R10.5 is considered relevant to a goal, R10.5.1 is included). Using this classification, one can observe that all listed requirements are relevant to at least one goal, justifying their inclusion in the design.

Goal	**Requirements**
G01: Education	R09.1, R10.1, R10.2, R10.4, R10.5
G02: Motivation	R09.1, R10.1, R10.3, R10.4, R12.3
G03: Security	R07, R08, R12.2
G04: Assurance provision	R11.3, R10.5.1
G05: Impediment minimisation	R09.3
G06: Interruption minimisation	R09.2, R12.3, R07.3.1
G07: Centralised intelligence	R10.4, R10.5, R11.1, R11.2
G08: Machine learning	R07.3
G09: API	R08.3
G10: Risk assessment	R07.3.1, R09.4.3, R12
G11: Policy format	R07.1.2, R07.3.2, R09.4

Table 5.1: Traceability between the stakeholder goals of the PISA and its defined requirements

Chapter 6

Architecture

In this chapter, architectural alternatives for the PISA system are discussed. To aid understanding of the terms used in this chapter, a short glossary of terms is presented at the start of this chapter. A short example of envisioned behavior follows as a technical counterpoint to the use scenario in section 4.1. Design and implementation tradeoffs are discussed for each of the alternatives and a rationale is given for the architectural option that was chosen for the prototype implementation.

6.1 Chapter glossary

This section contains the terms used throughout this chapter to describe the various parts of the architecture. They are listed in alphabetical order below:

Action : A measure taken by the PISA system to remedy a situation that is undesirable (e.g., the lack of a virus scanner could lead to an action advising the user to install one as soon as possible).

Agent An entity (extension of an existing process or separate process) containing a monitor and/or controller element. Designed to detect events that happen in the user's device eco-system and act upon them if necessary (autonomously, if possible). The term is used since the dictionary definition aptly describes the functionality of the agent: *A person or thing that takes an active role or produces a specified effect*, or in the context of computing: *an independently operating program, typically one that performs background tasks such as information retrieval or processing on behalf of a user or other program.*

Architecture Many terms exist for architecture, and as such it is relevant to define the term itself when considering this chapter. *Architecture* in the context as used in this thesis covers the description of a system according to a connector-component viewpoint as described in ISO 42010. It should also be noted that the stakeholder chapter offers a choice-agnostic part of the architecture as defined by this standard.

Controller Module within an Agent designed to affect changes within the system. Acts when instructed to by the agent's Coordinator module. From the dictionary definition: *a person or thing that directs or regulates something.*

Coordinator Module designed to facilitate communication between different elements of the PISA system, such as Controllers, Monitors and other Coordinators. The dictionary defines it as *[something that brings] the different elements of (a complex activity or organization) into a harmonious or efficient relationship.*

Database (DB) A standard database implementation to store policies, active agents and usage information. Databases exist in on 3 levels: agent level, user level and centralised level, each with their own set of policies, usage information and, when applicable, lists of active agents.

Event An occurrence of something detectable and relevant to the PISA system.

Level A subset of the architecture that has differing locations and responsibilities. Level categories are agent (one or multiple on a device, designed to perform specific actions and monitor for specific events), user (encompassing all of a single user's devices, including the user level database containing the information needed to provide the user with centralised intelligence) and centralised (containing elements such as the policy database and risk repository. This encompasses all PISA systems on all devices).

Monitor A module within an agent designed to detect events occurring within a system. From the dictionary: *a device used for observing, checking, or keeping a continuous record of something.*

PISA The personal information security assistant as a whole, encompassing all levels of the system, ranging from individual agents to the central server.

Central database communicator A module designed to interface with the policy database and to keep the database on its respective level up-to-date with policies relevant to the user's current risk profile. This module performs periodic checkups on the policy set and applies changes as needed.

Policies Guidelines based on which PISA takes action. These policies are categorised into risk profiles as needed to suit different users'/roles' needs. Policies can pertain to both the agent and user level, and a complete set is stored at the centralised level.

Status interface The human interface by which users can check their security status, as well as amend their risk profile as needed. This location conveys centralised intelligence to the user.

6.1.1 Behavior

Examples

Below, a short list of possible policies/scenerios is given to aid the reader in understanding the technical behavior and responsibilities of the system:

- A PISA agent is hooked up to a firewall and detects abnormal traffic on a specific port. It sends a signal to another agent capable of alerting the user that there has been a possible intrusion of the user's network.
- A PISA agent is installed on a device and submits a request to a service capable of registering the agent to a list of all agents linked to the user. This service then provides the agent with an updated list of all policies related to both the agent's action set and the user's established risk profile (e.g. "I wish to be informed of any unusual browsing activity").
- A PISA agent notices that several unidentified services are running on its device. It queries a service capable of identifying these services and, should they prove unidentifiable or malicious, warns the user of their presence.
- A PISA agent notices that the site a user is browsing on currently has characteristics in common with certain phishing websites. It considers the user's risk profile and elects to warn the user of a possible phishing attempt.

This list is not exhaustive, but serves to illustrate the list of more technical behavioral examples in the section below. Three distinct elements of the PISA system's behavior have been modeled as they are considered to be relevant to describe the functionality of the system: registering an agent with a relevant service, updating the policies in agent and user level databases and handling the occurrence of an event. These scenarios use the agent hub architecture, see section 6.2.1.

Registering an agent

Using an IP address, port and socket, agents register themselves with the PISA agent hub by communicating their presence to the PISA coordinator. The coordinator notes the arrival of a new agent by adding it to the database. The coordinator then retrieves an updated set of policies for the agent and sends it as part of the acknowledgement sequence. Agents should notify the agent hub when they go offline, while the agent hub's coordinator module periodically polls the agents to check their availability. In the case of a timeout or shutdown message, the agent is deregistered from the table of active agents in the PISA system. Figure 6.1 illustrates the interaction that takes place when an agent registers. It should be noted that if agents use a persistent database, it is possible for them to operate autonomously without registering with the database, partially alleviating the downside of having to keep a centralised user device available to all other devices in the user's device ecosystem.

Updating the policy set

The central database communicator is responsible for periodically updating the list of policies available in the user-level database. When the time comes, it requests a new set of policies via an update process from the central database server. It transcribes the changed entries to the user-level database and informs the PISA coordinator to send policy updates to all active PISA agents. This interaction is illustrated in figure 6.2.

Handling event occurrences

The PISA's modus operandi is responding to events generated by the agent's monitor modules by acting upon them via the controller modules. When an event occurs, it is checked against the agent-level database. If an action is found that can be executed locally, it forwards the execute command to the controller module. If it requires the action of a different agent, the entire event is forwarded to the agent hub to be handled/forwarded over there. If the information is deemed relevant for the operation of PISA or improvement of the risk repository/policy database, usage information associated with this event can then be logged in the agent to be retrieved at a later date. The user should be aware of information that is being collected at all times. Figure 6.3 illustrates the execution of a policy based on an occurring event.

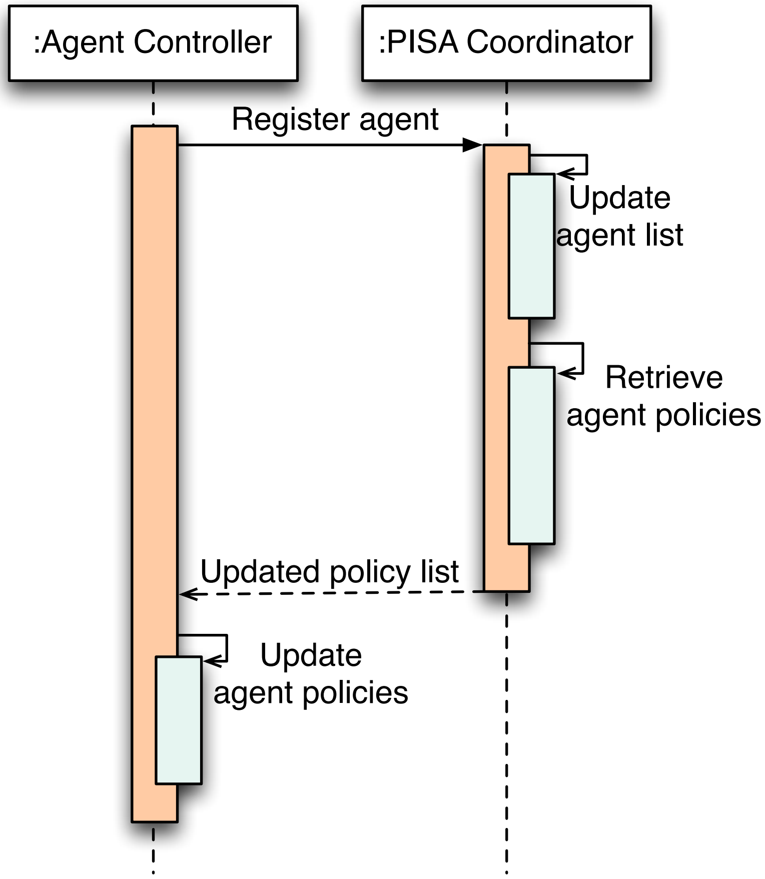

Figure 6.1: Sequence diagram of an agent registration sequence in the PISA architecture

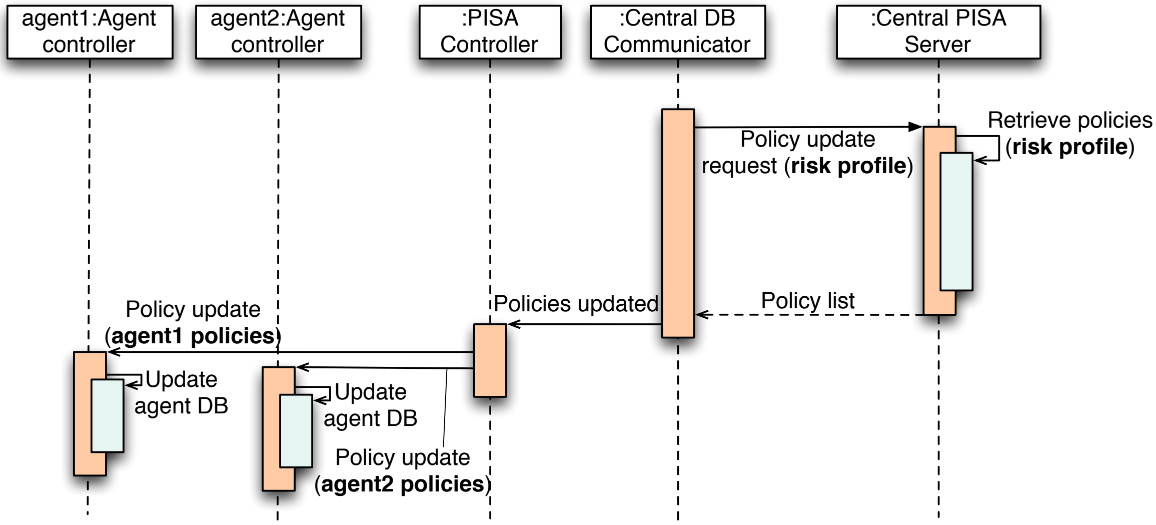

Figure 6.2: A sequence diagram of a policy update procedure

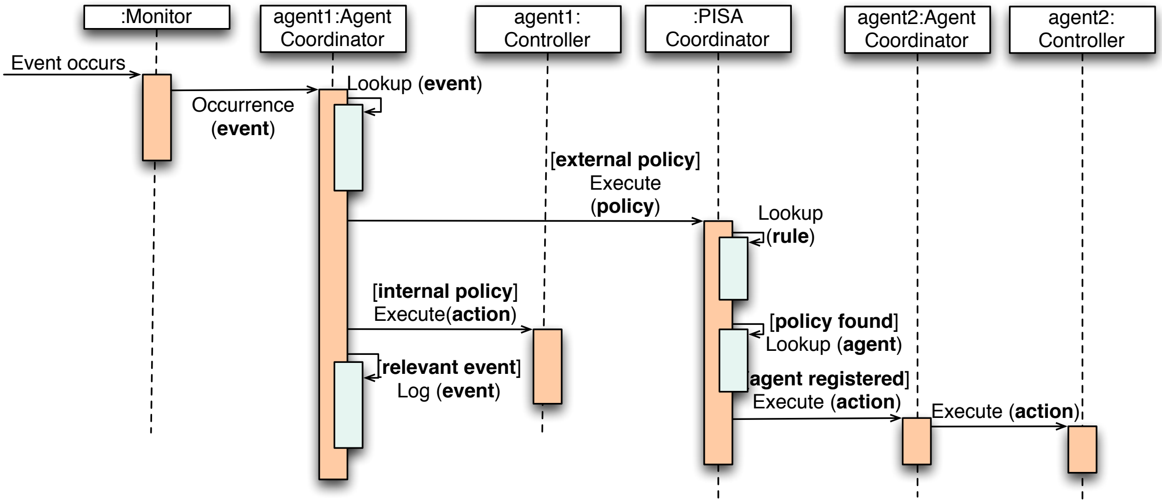

Figure 6.3: A sequence diagram of an event occurrence

6.2 Proposed architectures

In this section, 4 alternative designs are proposed and discussed: the agent hub, server centralised, peer-to-peer and master/slave architecture.

6.2.1 The agent hub architecture

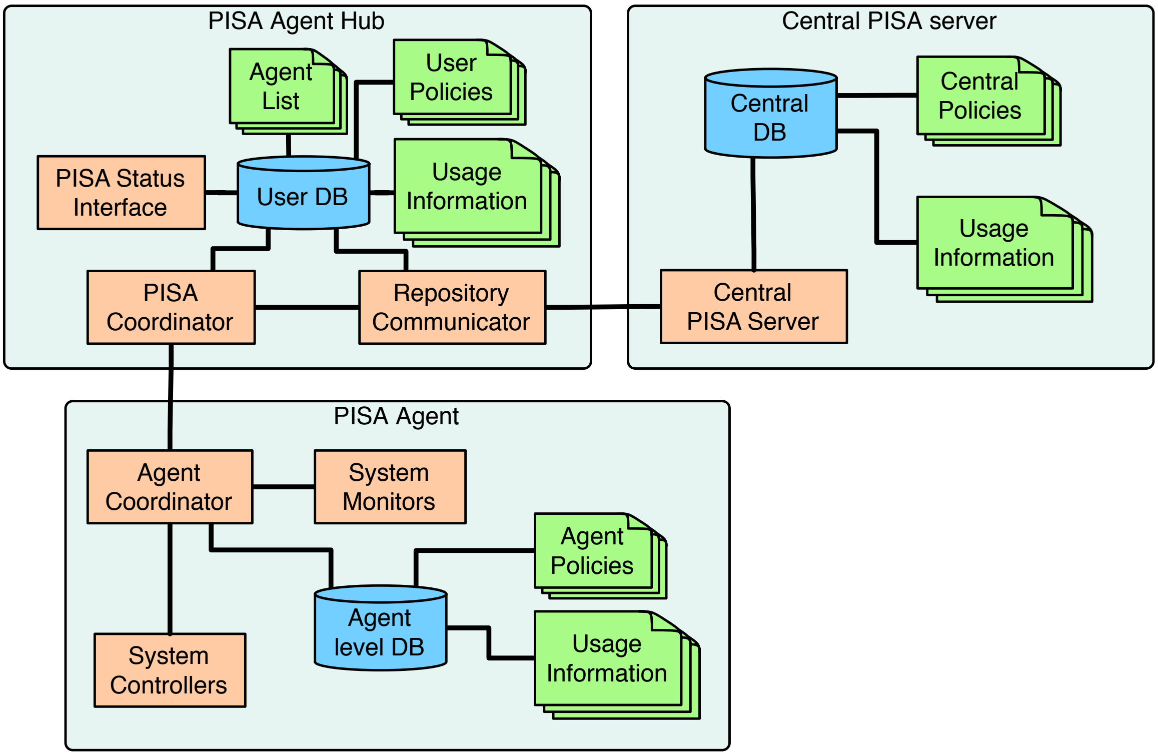

Figure 6.4: An agent hub based architecture

The PISA agent hub architecture follows a broker software pattern where one device contains a PISA *agent hub* application; this application acts as a go-between for the agent applications and the central PISA server, as well as inter-agent communication. It stores information relevant to the user and its agents, keeping the policy sets of all agents up-to-date.

The good:

Server load Due to the inclusion of a clear agent hub, only one device has to communicate with the centralised database. This minimises the load on the server, which leads to better scalability of the system.

Centralised intelligence Due to the inclusion of a clear centralised point, data collection for purposes of centralised intelligence is relatively easy to implement.

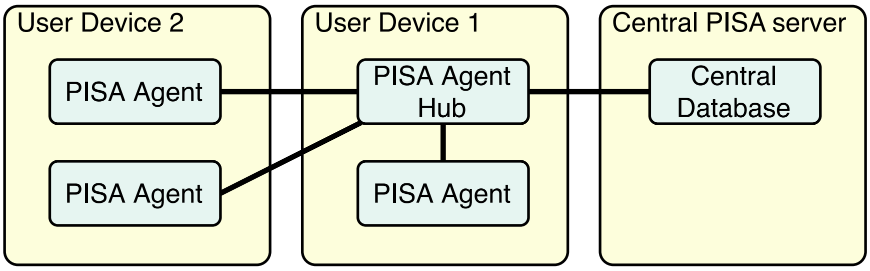

Figure 6.5: The PISA agent hub architecture in a multi-device environment

Ease of implementation With a simple client-server software pattern, the agent hub and agent application are considered relatively easy to implement, as the specialised agent applications contain less program logic. Due to the specialised nature of some possible agents (e.g., browser plugins to monitor browsing activity), including agent hub functionality in agents would possibly lead to significant code duplication.

The bad:

Availability A large point of concern is the availability of such an agent hub when it is maintained by the end-user (i.e., on a user's home computer or similar device). Inexpert maintenance of a system will quite certainly lead to lower uptimes, limiting the effectiveness of the PISA system as a whole. While the agents have their own agent level database containing policies on which they can act internally, any policy involving multiple agents is impossible to execute, nor is it possible to update policy sets while the agent hub is offline.

Security An additional layer of communications brings with it new threats to a system as lines of communication have to be opened to facilitate communications.

The design is illustrated in figure 6.4. Figure 6.5 illustrates how the system functions with regard to multiple devices.

6.2.2 The server centralised approach

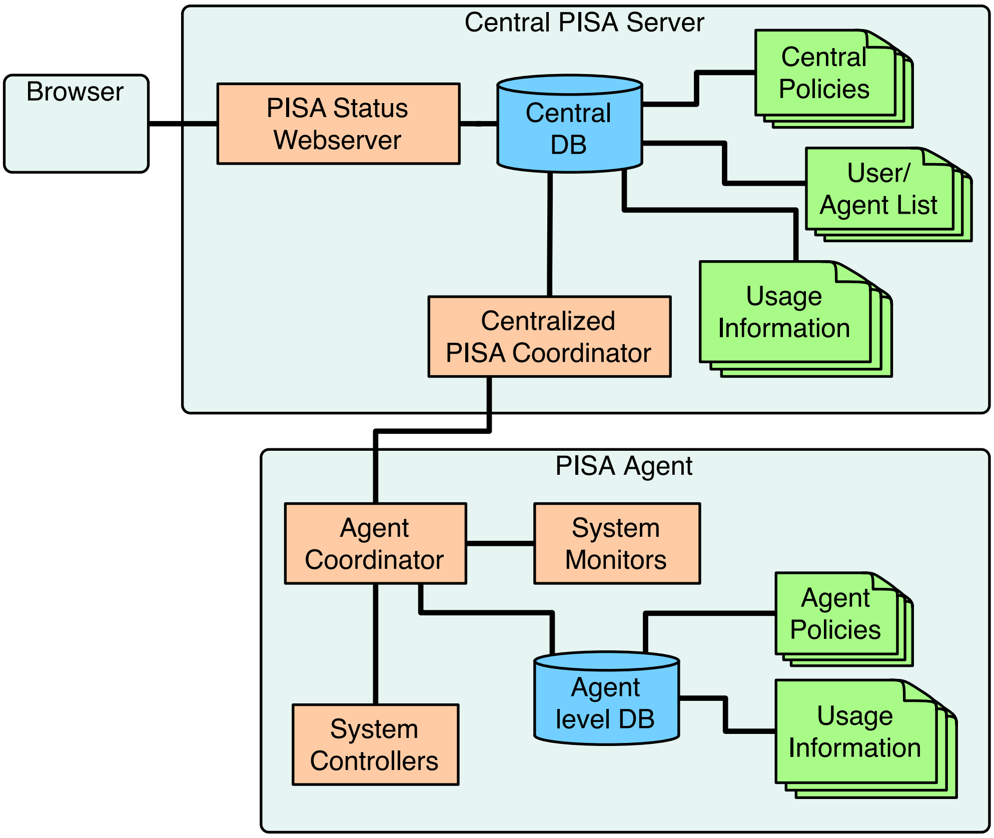

Figure 6.6: A server centralised version of the PISA architecture

One major alternative to the agent hub architecture is scrapping the agent hub concept and having decentralised agents communicate with a centralised coordination center directly (figure 6.6). This would remove much of the burden of having to maintain the hub from the user. This centralised location would be maintained by professionals and, as such, would be able to maintain a far higher uptime. Additionally, the removal of a layer of communications improves the security of the design.

The good:

Expertise In this architecture, a larger part of the system is in the hands of professionals, with the agent hub functionality effectively being integrated into the central PISA server. This guarantees higher uptime and reduces both cost (electricity) and maintenance time for the user.

Architectural robustness A removed layer of complexity makes the system more robust: less communicating links reduces the risks of something going wrong somewhere in the line of communication leading from agents to the central PISA server.

Status availability The current approach only allows for the PISA status to be polled from the agent hub application. In this architecture, a user could go to a webpage on any location and retrieve status information as desired.

Security The removal of a layer of communications obsoletes the need for PISA agents to communicate with each other, isolating devices and removing the need for a PISA agent to listen for incoming messages.

The bad:

Security Being able to log in from any location to access PISA's status brings its own set of security risks: for obvious reasons, one would not want just anyone to check the security status of every device of the user; doing so would invite targeted attack. As such, much thought would need to be put into securing this line of information.

Scalability By changing the structure to incorporate the agent hub in the central PISA server, scalability has the potential to become an issue: instead of one connection per user to the central PISA server, the server has to contend with a multitude of devices per user requesting inter-agent actions and policy updates.

Agent polling Additionally, in the chosen implementation, the agent hub periodically polls agents to see whether they are still operational. In the centralised architecture considered here, such polling has the potential to become a major performance concern, requiring both bandwidth and CPU power, as a far larger list of agents is kept up to date.

Information control Usage information that is only relevant for inter-agent policy implementation is now sent to a third party, removing control of the user over this information.

Database complexity Database complexity increases as not only a subset of the usage information and a list of policies has to be maintained, but also a list of users and agents online with the system.

Standalone operation In the agent hub version of the architecture, it is possible for the user to use a local set of policies to update the user database, effectively running the system without external intervention. To do so in this architecture, the user would have to run its own web server along with the database, requiring more expertise on the user's part.

6.2.3 Decentralised approaches

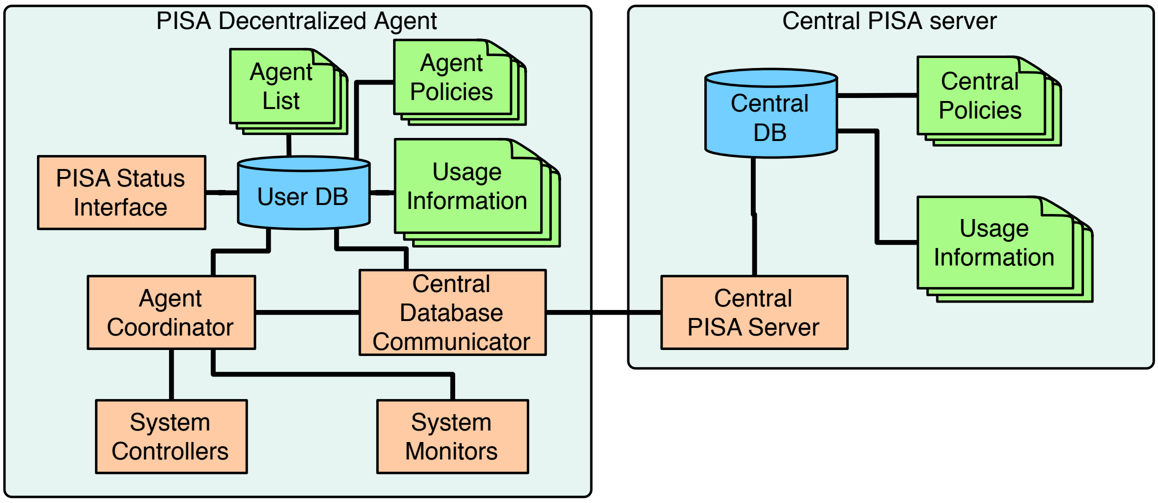

Figure 6.7: a decentralised version of the PISA architecture

Two decentralised approaches were considered: a master/slave configuration of agents and a pure peer-to-peer approach. Both use a decentralised version of the agent architecture, illustrated in figure 6.7. This version integrates a user interface and central database communicator module, as well as data storage functionality previously incorporated in the agent hub. These changes effectively subsume the functionality of the agent hub into agents. It should be noted that it is not intended that the PISA status interface implementation be mandatory; as long as at least one active agent is able to communicate the system's status to the user, no problems should arise when attempting to inform the user. Advantages and disadvantages for the two specific alternatives (peer-to-peer and master/slave) are discussed below.

Peer-to-peer

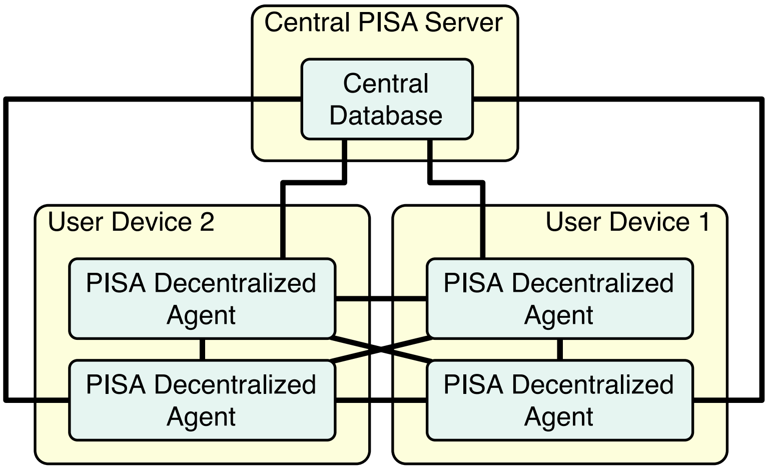

Figure 6.8: A peer-to-peer layout of the decentralised PISA architecture

A pure Peer-to-peer approach implies communication between decentralised agents and a central database without a clear point where data is collected for communication with the central database. Many approaches can be taken for the actual topology and discovery of these agents: the fully connected mesh shown in figure 6.8 does not necessarily need to be used. Clues and inspiration for specific implementations of this type of architecture can be found in wireless SCADA systems, large sensory networks and other wireless distributed networks. Some considered advantages and disadvantages are listed below.

The good:

Scalability A peer-to-peer approach is eminently scalable and as such can handle a large amount of devices for a single user.

No single point of failure If the agent hub is no longer a part of the architecture, no single point of failure exists in the user's device ecosystem. This gives the architecture the same level of resilience as, if not more than, the purely centralised server structure discussed above.

Communication with the central database Because of the peer-to-peer structure, an agent can potentially communicate with the central database without having a direct connection to it, provided that it can trace a path via agents to the central PISA server. This also allows for more efficient usage of network communications (i.e., agents can communicate amongst themselves before communicating with the central PISA server, sending a policy update request in one message rather in two concurrent ones).

Robust Agents can disappear without compromising the system itself. Additionally, it is possible for the central PISA server to act as a "middle man", linking several parts of the user's agent network if communications between them are interrupted. This adds a dimension of self-repair to the PISA network's communications.

The bad:

Complexity Peer-to-peer topologies are usually complex to set up and maintain, requiring a larger time investment to both develop and test.

Dynamic contexts Where some applications of distributed networks use static, or sparsely moving nodes, by its very nature some of the user's devices move from context to context rapidly, e.g. when considering the user's mobile devices. This adds a large dimension of uncertainty and a rapidly changing context to the peer-to-peer network, possibly making the resource cost for maintaining a stable connection between agents unreasonably high.

Interface views Without a centralised view (either in the central PISA server or the agent hub), each agent is free to implement its own interface. This adds a certain amount of uncertainty and bother for the user, since finding the right (or, indeed, any) interface can be a complex task if many agents are deployed. Even if every interface is supposed to show a unified view of the agent network, creating an architectural module that can guarantee one view on a predictable location for the user can be a challenging task. A viable alternative might be a dedicated agent for interfacing with the user (i.e., without monitor or controller elements), but this agent would have to be developed as a separate part of the PISA system, adding development time.

Data collection for status updates In the same vein, collecting data from the entire network of PISA agents the user employs can be both resource intensive and challenging to get right: how to maintain both a measure of performance and ensure information is up to date if a peer-to-peer architecture is used?

Encryption of communications via agents If information is sent (be it a requested relay of policy information from the central PISA server, or an update of usage information and status for an interface), it needs to be encrypted. While this is not too different from the requirement in a centralised architecture, the uncertain route which a message may take to reach its destination makes the requirement to properly encrypt data and manage key distribution a challenging task.

Malicious agents Much like any distributed sensor network, one has to consider the possibility of an attacker adding malicious agents to the user's network: either to request status updates (giving the attacker possible routes by which to attack the user) or to get an overview of the agents available to the user. This security issue has to be addressed properly before a peer-to-peer approach can be considered.

Master/slave

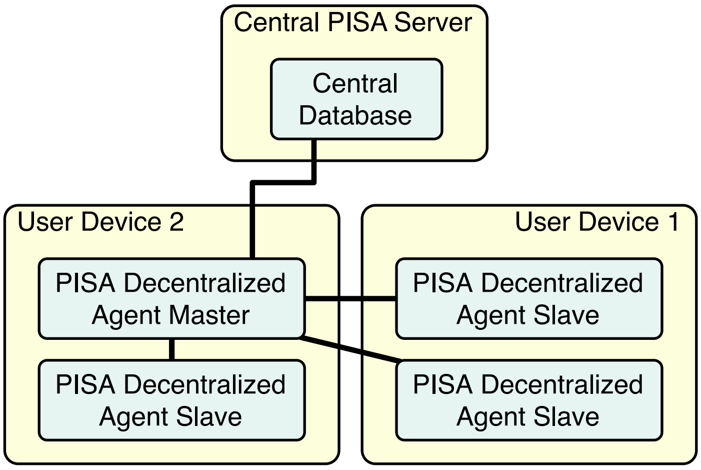
Central PISA Server
Central Database
User Device 2
User Device 1
PISA Decentralized Agent Master
PISA Decentralized Agent Slave
PISA Decentralized Agent Slave
PISA Decentralized Agent Slave

Figure 6.9: A master/slave layout of the decentralised PISA architecture

A hybrid variant of the peer-to-peer and agent hub architectures, in this mode one of the agents is selected as a primary agent (master), with the rest of the agents communicating to the central PISA server and other agents via this agent instance. This approach avoids many of the routing issues present in a peer-to-peer approach while retaining the flexibility of not requiring a central agent hub being online and available. Figure 6.9 illustrates this approach.

The good:

Scalable This approach offers better scalability than an agent based hub: with the decentralised agent architecture, routing through multiple agents is still possible, minimizing the amount of communication channels open simultaneously.

Self-repairing Because every agent has the possibility to run as a master instance, even isolated subnets of agents are able to function self-sufficiently as long as communications with the central PISA server can be established.

Predictable lines of communication Unlike the peer-to-peer approach, a master instance can be identified in this layout, simplifying the flow of information in an agent network. This can help with status displays (centralised location for information storage) and security (new agents would connect in a slave state and be less likely to cause damage to the PISA agent network).

The bad:

General peer-to-peer issues Many of the issues pertaining to security, setup and maintenance that apply to the peer-to-peer architecture still exist in this approach: dynamic contexts, data encryption and the question of user interaction continue to be issues that have to be addressed.

Malicious agents Care has to be taken when designing a protocol for electing a new master agent, since hijacking by a malicious outside agent is possible, compromising the entire PISA network of the user.

Data collection In the agent hub architecture, we have a separation of user data between the agent, user and central level. Since a master instance can become a slave instance at a later point in time, care has to be taken to persistently respect this separation of data.

Setup and maintenance A robust, secure protocol needs to exist to re-elect a master if the previous one fails to respond. Additionally, a protocol needs to exist to discover whether a master already exists and circumstances have to be defined in which the one master instance relinquishes its state for another master instance, should two sub-nets discover each other.

6.3 Conclusions

With these 4 proposed alternatives for the design of the PISA system, it is possible to choose one of these options to create a prototype. This thesis presents a prototype implementation according to the agent hub architecture. This choice has been made for the following reasons:

Centralised The scalability issues and added implementation work required on the side of the central PISA server led to the conclusion that an agent hub was a relatively more desirable option vs centralised.

Peer-to-peer Both the complexity of maintaining such a network when the devices themselves move freely to and from different networks and contexts, as well as the difficulty in setting up such a network precludes implementation in this format for the PISA prototype. It remains an attractive option for future expansion, however.

Master/slave Due to the inherent complexity of designing a robust decentralised agent network, this architectural variant is also not chosen for a prototype implementation. The concepts definitely still have merit, however, and future prototypes will most likely benefit from revisiting these architectural alternatives.

As can be noted, the main consideration when exploring the agent hub alternative was the implementation effort involved in developing a robust prototype. Future work may yet benefit from prototype implementations of the other three alternatives as a way to compare the practical merits of each design.

Chapter 7

Validation

This chapter offers validation of the previously presented work in three parts: a traceability analysis of the four presented architectures (with proposed refinements to remedy any discovered shortcomings), an implementation of one of the architectural alternatives as discussed in section 6.3 and scrutiny by a focus group of industry professionals to assess the realism and the perceived strong/weak points as seen by these individuals.

7.1 Architecture traceability

This section provides a measure of validation of the discussed architectures by checking their conformance to the requirements listed in chapter 5. Explicit failures to meet the requirements listed in chapter 5 are discussed, along with potential amendments to the architectures. The symbols used to denote conformance to a requirement (or lack thereof) are listed in table 7.1. Traceability matrices for the four different architectures sorted by goal are given in tables 7.2 and 7.3.

7.1.1 Traceability discussion

The architectural traceability diagrams in tables 7.2 and 7.3 illustrate a few things about the currently defined architecture. First and foremost is the underspecification of all of the architectures, particularly in the field of human interaction. To specify the architecture w.r.t. these requirements, other descriptive elements such as use cases need to be used. This expansion of the architectural description is an important step in the development process of the PISA system.
In addition to these underspecified elements, however, some explicit shortcomings of the presented architectures need to be addressed to ensure the final product *can* adhere to all the requirements presented in this thesis (and beyond):

R07.3 A component of machine learning is not defined anywhere in the stated architectures. Explicit modules would be necessary to contain this functionality. The specifics of what data is needed, where it is aggregated and how it would enhance the current policy sets of agents begs clarification. Future work (see chapter 8) includes the need for a closer look at machine learning in the context of the PISA.

R08.1 Inter-PISA system communications have not been defined at all in the current architectures. Both a communications mechanism/protocol and modules containing relevant functionality would need to be added to realise this feature. It should be noted that the server centralised architecture is better suited for this inter-PISA communication since the lines of communication are more centralised (i.e., the PISA agents already talk to each other via a single entity which is also in use by other user's PISA systems).

R11.3 Similar to the Inter-PISA system communications, ways for third parties to query the security status of the system have not been defined in any of these architectures. For such a refinement to take place, mechanisms need to be defined to secure communications and confidential information. Additionally, security indicators would need to be developed (see requirement 10.5.1).

Due to the underspecification of the current architecture, further specification would need to be performed in order to fully assess the relative strengths and weaknesses of each alternative. In order to make the strengths and shortcomings of the architecture more concrete, however, the development of a prototype based on the agent hub architecture is discussed below. Section 6.3 contains reasoning for the architectural choice when developing the prototype.

Symbol	Category	Description
✓	Good	Validated by the current architectural description
?	Underspecified	Needs further specification of the architecture
✗	Insufficient	The current architecture does not conform to this requirement

Table 7.1: A description of the operators used in the requirements-architecture traceability table

Goal	Requirement	Agent hub	Server centralised
Education	R09.1	?	?
	R10.1	?	?
	R10.2	?	?
	R10.4	?	?
	R10.5	✓	✓
Motivation	R10.3	?	?
	R12.3	?	?
Security	R07.1.1	✓	✓
	R07.1.2	?	?
	R07.2	✓	✓
	R07.3	✗	✗
	R08.1	✗	✗
	R08.2	?	?
	R08.3	?	?
	R12.2	?	?
Assurance	R10.5.1	?	?
	R11.3	✗	?
Impediments	R09.3	?	?
Interruptions	R09.2	?	?
Intelligence	R11.1	?	?
	R11.2	?	?
Risk assessment	R09.4.3	?	?
	R12	?	?
Policy format	R09.4	?	?

Table 7.2: A traceability matrix between PISA's requirements and proposed architectures

Goal	Requirement	Peer-to-peer	Master/slave
Education	R09.1	?	?
	R10.1	?	?
	R10.2	?	?
	R10.4	?	?
	R10.5	✓	✓
Motivation	R10.3	?	?
	R12.3	?	?
Security	R07.1.1	✓	✓
	R07.1.2	?	?
	R07.2	✓	✓
	R07.3	✗	✗
	R08.1	✗	✗
	R08.2	?	?
	R08.3	?	?
	R12.2	?	?
Assurance	R10.5.1	?	?
	R11.3	✗	✗
Impediments	R09.3	?	?
Interruptions	R09.2	?	?
Intelligence	R11.1	?	?
	R11.2	?	?
Risk assessment	R09.4.3	?	?
	R12	?	?
Policy format	R09.4	?	?

Table 7.3: A traceability matrix between PISA's requirements and proposed architectures, part 2

7.2 Implementation of the agent hub architecture

The PISA prototype has been implemented using the agent hub design as a starting point (see section 6.2.1 for details). Though several elements are not implemented, the general design adheres to the proposed architecture. Most notably, the central database and central database communicator module are absent, and the current PISA Agent is a chrome extension that does not utilise a database to store either usage data or policies (i.e., it queries the agent hub every time an event happens to check whether a policy needs to be applied). Figure 7.1 illustrates the current architecture, while figure 7.2 provides the implementation details that go along with the architectural design.

7.2.1 Prototype use scenario

The PISA prototype currently handles one scenario for demo purposes and has an agent that reflects this purpose. The scenario handled by the current prototype involves entering a password in a designated field while browsing on an insecure site. Many phishing attempts involve the use of user/password fields on a bogus site made to fool unwary users into thinking they are logging into a secure website belonging to a trusted party. While it is (relatively) easy to create a website that looks like a trusted website, however, it is hard to mimic the certificate that validates the source of the website. As such, a normal website should (and usually does) provide login forms that send data over a secured HTTPS connection, while phishing sites do not. The prototype detects the presence of password fields on a website while browsing and, if the connection is not secured via HTTPS, blocks users from entering their password. Below is an explanation of the different parts implemented to realise this scenario.

7.2.2 Description of the PISA implementation

The architecture currently consists of 4 different elements on the user's system:

Database A database containing the necessary policies for the listed scenario. In this case, this is modeled using a simple event called "password field found on unsecured connection", linked to a specific agent (the PISA agent, a Chrome browser extension). This tuple of values forms a key that leads to the action "disable passwords".

PISA java application A simple java application following the client/server pattern. It uses the `DispatcherServer` class to listen for incoming agent connections, store a list of active agents and communicate with the database. The `AgentHandler` class handles input from and output to the Chrome browser extension. It uses the Connector/J driver to communicate with the MySQL database and socket-based communication for the Chrome extension.

PISA Chrome extension A Chrome browser extension written in JavaScript to detect browsing behavior and act upon it. `Control` and `Monitor` are content scripts embedded in each webpage the user visits (this is needed to manipulate webpages) while `Background` is a continuously running JavaScript file that opens when the extension is loaded (normally, when the browser starts). `Background` receives events from `Monitor`, sends actions to `Control` and communicates to the outside world via a Native Messaging Host (see below).

NativeHostHandler Google Chrome restricts the set of functions available as API for building extensions. As a result, only content scripts (`Control` and `Monitor`) are allowed to modify webpages the user loads and certain functions are not available at all. One of these restricted functions is socket communication. This means that an extension has to either communicate via HTTP requests or via Native Messaging Hosts: cooperating applications started by Google Chrome to facilitate communication with the rest of the system. `NativeHostHandler` is such a native messaging host, written in Python. It is executed by Chrome, which communicates with Background via standard I/O. It then relays information via socket communication to the PISA Java application. This script terminates whenever Chrome closes or the PISA Java application becomes unreachable. It is started whenever a message needs to be sent and it is not yet running.

7.2.3 Prototype traceability

Any actual implementation has more clarity in what it can and cannot do over a theoretical specification. As such, requirements can be more clearly assessed as fulfilled or unfulfilled. Table 7.4 offers an overview of which requirements are explicitly fulfilled in the prototype, and which are not. Additionally, an overview of the measure in which the implementation achieves the *goals* listed in chapter 4 is given below, with proposed changes to remedy perceived shortcomings:

Education None of the educative requirements were fulfilled. For this, a user interface needs to be constructed and policies defined that interact with the user.

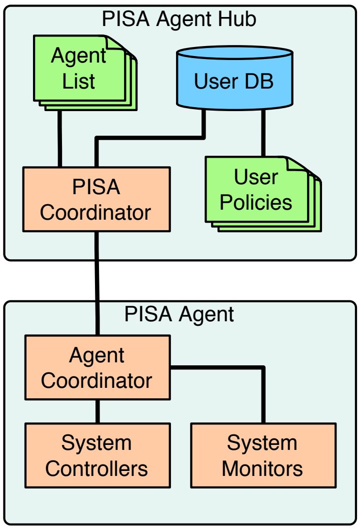

Figure 7.1: The current prototype architecture

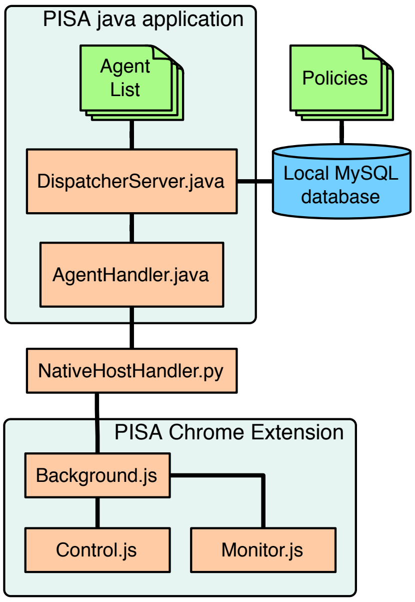

Figure 7.2: An implementation level diagram of the prototype architecture

Goal	Requirement	Prototype
Education	R09.1	✗
	R10.1	✗
	R10.2	✗
	R10.4	✗
	R10.5	✗
Motivation	R10.3	✗
	R12.3	✗
Security	R07.1.1	✗
	R07.1.2	✓
	R07.2	✗
	R07.3	✗
	R08.1	✗
	R08.2	✓
	R08.3	✓
	R12.2	✗
Assurance	R10.5.1	✗
	R11.3	✗
Impediments	R09.3	✗
Interruptions	R09.2	✓
Intelligence	R11.1	✓
	R11.2	✓
Risk assessment	R09.4.3	✗
	R12	✗
Policy format	R09.4	✗

Table 7.4: A traceability matrix between PISA's requirements and the implemented prototype

Motivation Similar to education, a basic user interface is needed in order to actively motivate a user with regards to security.

security Requirements 7.1.2, 8.2 and 8.3 were fulfilled, the rest is still lacking. For this to be improved, a larger set of scenarios needs to be constructed and the architectural shortcomings (e.g. the lack of an agent-level policy database) need to be remedied.

Assurance Assurance provision was not a part of the current prototype, and as such it does not fulfill the related requirements. Communication mechanisms and security indicators will need to be developed in order to achieve this goal.

Impediments Since the user is physically blocked from entering his password on an unsecured site while PISA is active, the user experiences (avoidable) impediments. A user interface that alerts the user and offers an alternative is needed to fulfill this requirement/goal.

Interruptions Currently, the user is not interrupted during his workflow since the PISA does not initiate any unsollicited communication. This will change if more scenarios and policies are implemented, and as such the achievement of this goal is subject to the policy database it uses.

Risk Assessment No research has been performed on specific risk assessment/risk profiling methods as it is deemed beyond the scope of this thesis. As such, these requirements are not yet fulfilled. Research needs to be performed in this area before these requirements can be fulfilled.

Policy Format Similar to the risk assessment goal, research into different policy formats is beyond the scope of this thesis. Simple event-action rules as used in the prototype are not considered to be examples of usable, scalable, risk assessment derived policies. Research needs to be performed in this area before these requirements can be properly fulfilled.

7.2.4 Deviations from the agent hub architecture

In addition to the listed shortcomings when regarding the requirements listed above, the current prototype deviates from the proposed agent hub architecture in section6.2.1. These deviations are:

A status interface Currently, the agent hub runs as a command line application. A graphical interface is needed for easier user interaction and startup, giving the user all the information relevant to the user's security.

An agent database Currently, the PISA agent (the Chrome extension) does not store its own policies, instead using predefined events and actions, letting the User level database link them as required. Essentially, every policy is now an external policy (see Figure 6.3), which does not allow for standalone operation of the agent.

A structure for inter-agent actions A scenario could/should be constructed which allows for inter-agent policies to be demonstrated, i.e., an event in one agent causes an action in another. Though effectively, this is what happens now (i.e., an event is sent to the User level for inspection, which then forwards it back to an agent), a case involving two separate agent instances would be desirable.

Usage information Usage information is currently not logged to any database. Relevant events need to be identified and a database schema needs to be designed to store this information in a secure, anonymised manner.

A central PISA server : currently, the agent hub does not communicate with a central PISA server. This PISA server and all interactions with it need to be designed in further detail and implemented.

7.2.5 Conclusions

Though the prototype is a good start for future iterations of the PISA, the current implementation is too underspecified to assess the validity of the requirements presented in chapter 5. To perform serious tests, at the very least the prototype has to be made to conform to the agent hub architecture in order to test its effectiveness. The current implementation, however, does serve to illustrate several things:

- The non-triviality of diverse technologies, validating R02;
- The need for a policy format in order to structure communications within the PISA system;
- The extensive development time needed to integrate agents in such a way that a reasonable set of actions can be performed by PISA.

the current implementation of this prototype has added value mainly in how it serves to act as starting point for future iterations of the PISA system, not by validating the architectural proposals of chapter 6.

7.3 Validation by experts

This section acts as a validation in two parts: the first part contains the findings of a usergroup meeting of several companies, illustrating possible additional goals and requirements of the PISA tool as both a product and a research tool. The second part contains the findings of several industry professionals when considering the proposed architectures in chapter 6.

7.3.1 Usergroup meeting

As part of the validation process, a panel of experts was assembled, each belonging to a company with possible interest in the PISA application for its customers and/or employees. This usergroup was shown the implementation as presented in 7.2.2. Some of the salient points for future directions are:

Two-way assurance One of the major suggestions offered was the concept of two-way assurance for the PISA system. Whereas the current PISA design envisions assurance provision as a service for 3rd parties to query the security status of confidention information they have a legitimate claim to, the concept could be reversed: PISA could act as a technology that verifies the security of a service that PISA's users use. This way users could e.g. verify a bank's security status before deciding on whether to open an account there.

Database types The databases presented in the architectures of this thesis are not explicitly defined either way, but the implementation uses a relational database. One of the possible research directions is how new database technologies such as graph-based databases could help search over potentially disparate datasets in order to identify threats and trends in a typical user device.

Data collection The current designs and prototype originate from the stakeholder analysis and goals presented in chapter 4 of this thesis. Another potentially valuable direction of research, however, is considering the potential for data collection and analysis when considering a tool such as PISA: its nature as coordinator of many monitors has excellent potential for the aggregation and interpretation of data. This needs to be considered when implementing future implementations of the prototype.

7.3.2 Industry professionals

As a second part of the expert validation of this section, industry professionals were asked to give feedback on the design process and presented architectures of this thesis. Apart from the expected differences in professional language (e.g., the precise definition of the concept *architecture*), two important security considerations were identified:

Anonymisation and data collection One of the major concerns offered when considering any of the architectures is the nature of *usage data* as defined in the different parts of the architectures (agent level, user level, centralised level). Not only does transparency need to exist (as defined in R11.1), a justification has to be exist for the data collection itself. Should it prove necessary to store any information on a level higher than the agent level (for correlation of data), an anonymising module should be included in the architecture in order to preserve confidential usage information both on other levels in the architecture and during transmission. Ideally, no processing of user specific data would need to take place, in which case fully anonymised and/or encrypted information could be transmitted to the central level, fully preserving anonymity of the user.

Communications and security Any layer of communications that requires a listening element adds security concerns: should information exchange directly between agents be needed, a listening element would need to be included in the architecture. This inherently compromises the security of the system, as any attacker needs a way *in* when attacking a system. A case can be made for the server centralised architecture when considering this aspect, as it is the only architectural alternative that does not prominently feature inter-agent communication. The storage of information could happen on the centralised server in an encrypted manner, preserving privacy of information while improving the security of individual devices by removing the need to listen for unsollicited communication.

While the first concern is important when considering any of the proposed architectures, the second concern seems to heavily favor the server centralised architecture. Future work, then, should feature a prototype implementation in this style to judge its merits.

Chapter 8

Conclusions

In this chapter the research questions of chapter 3 are considered and a summary of results is given. Future work on the PISA project is then discussed to put these findings in context.

8.1 Research findings

Below, a summary of findings is given for each question proposed in chapter 3.

1. *What are the stakeholders and goals of the PISA system?*
Stakeholders depend on the use scenario given, and as such a definitive answer to this question cannot be given without a comprehensive list of use scenarios. Given the use scenario presented in 4.1, however, an initial answer can be given (and is listed in section 4.3). The relevance of this research, then, lies in the presented method of elicitation and classification of these stakeholders.

2. *What requirements can be used to describe the PISA system's goals?*
As with stakeholders, a definitive list of requirements cannot be given without a comprehensive list of use scenarios in which PISA has to function. The initial list of requirements derived from the stakeholders presented in this thesis is given in chapter 5. In addition however, requirements of the PISA system are also derived from new insights gained from the development process. As such, this initial set of requirements defined in chapter 5 serves as a stepping stone for a larger, more comprehensive list of requirements in a new iteration of the development process.

3. *What design alternatives exist for the PISA system?*
4 alternatives are presented and evaluated in chapter 6.

4. *How well do these design alternatives fulfill PISA's goals?*

4.1 *Which architectural alternative best fulfills the elicited requirements?*
The validation by traceability has mainly served to show the level of underspecification still present in the designs presented in 6. As such, no definitive answer can be given as to the relative conformance to requirements. The traceability shows a marginal advantage for the Server Centralised architecture (see section 6.2.2).

4.2 *How well does an implementation based on such an architecture fulfill the elicited requirements?*
Currently, the implemented prototype following the agent hub approach (see section 7.2) does not adequately cover the requirements listed in chapter 5. Further implementation of features is needed fully assess the effectiveness of this architectural approach.

4.3 *What is the opinion of industry professionals on these architectures?*
Apart from the obviously valuable insights of the user group (see section 7.3), the most notable points of improvement in the currently proposed architecture are the further specification and consideration of the security aspects of communication and data storage.

8.2 Future work

Several points have been noted throughout the development process and its validation. These points form the basis for future work suggestions, as listed below:

Use scenario expansion To ensure adequate coverage of requirements and ensure an ability to adapt the the diverse set of roles and contexts as specified in section 1.1, further consideration of the use scenarios that form the basis of the stakeholder elicitation is needed.

Architecture specification The current architectural alternatives do not adequately describe and validate all listed requirements. As such, further specification of the listed architectures is required in order to implement an adequate prototype.

The server centralised approach Both the validation by traceability and by experts seem to suggest the server centralised approach as a more promising approach than the currently implemented prototype. As such, future work can benefit from an implementation of this approach so their relative merits may be compared.

Risk assessment and policies In order to realise a set of the requirements listed in chapter 5, further research is required in the areas of risk assessment and policy definition. This combination of subjects will allow for the establishment and subsequent evolution of risk profiles for users and is considered an important part of the scientific relevance of the PISA project.

Machine learning Another crucial part of the scientific relevance of this project is derived from the automatic personalisation of risk profiles and policy databases. As such, research in this area is considered an important part of future work.

Bibliography

[1] I.F. Alexander and L. Beus-Dukic. *Discovering requirements: how to specify products and services.* John Wiley & Sons, 2009.

[2] Z.A. Baig. Multi-agent systems for protecting critical infrastructures: A survey. *Journal of Network and Computer Applications*, 35(3):1151–1161, 2012.

[3] M. Blythe, H. Petrie, and J.A. Clark. F for fake: Four studies on how we fall for phish. pages 3469–3478, 2011.

[4] Robert B. Cialdini. *Influence : the psychology of persuasion.* Collins, 2007.

[5] Fred D. Davis. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Q.*, 13(3):319–340, September 1989.

[6] W.K. Edwards, E.S. Poole, and J. Stoll. Security automation considered harmful? pages 33–42, 2007.

[7] Leon Festinger. *A theory of cognitive dissonance*, volume 2. Stanford university press, 1962.

[8] Martin Fishbein and Icek Ajzen. *Predicting and Changing Behavior: The Reasoned Action Approach.* Psychology Press, 1 edition, July 2009.

[9] B. J. Fogg. *Persuasive Technology: Using Computers to Change What We Think and Do (Interactive Technologies).* Morgan Kaufmann, 1 edition, December 2002.

[10] T. Gillie and D. Broadbent. What makes interruptions disruptive? a study of length, similarity, and complexity. *Psychological Research*, 50(4):243–250, 1989.

[11] M. Glassman and M.J. Kang. Intelligence in the internet age: The emergence and evolution of open source intelligence (osint). *Computers in Human Behavior*, 28(2):673–682, 2012.

[12] M.A. Harris, K. Patten, and E. Regan. The need for byod mobile device security awareness and training. volume 5, pages 3441–3451, 2013.

[13] J. Johnston, J.H.P. Eloff, and L. Labuschagne. Security and human computer interfaces. *Computers & Security*, 22(8):675 – 684, 2003.

[14] L.M. Kaufman. Data security in the world of cloud computing. *IEEE Security and Privacy*, 7(4):61–64, 2009.

[15] R.H.P. Kegel and R.J. Wieringa. Persuasive technologies: A systematic literature review and application to pisa. Technical report, University of Twente, May 2014.

[16] G. Klein. From a verified kernel towards verified systems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6461 LNCS:21–33, 2010.

[17] Wilson M. and Hash J. Building an information technology security awareness and training program. Technical Report 800-50, National Institute for Standards and Technology, October 2003.

[18] R. K. Mitchell, B. R. Agle, and D. J. Wood. Toward a theory of stakeholder identification and salience: Defining the principle of who and what really counts. *The Academy of Management Review*, 22(4):853–886, October 1997.

[19] K.D. Mitnick and W.L. Simon. *The art of deception: Controlling the human element of security.* John Wiley & Sons, 2001.

[20] M. Niazi and A. Hussain. Agent-based computing from multi-agent systems to agent-based models: A visual survey. *Scientometrics*, 89(2):479–499, 2011.

[21] H. Oinas-Kukkonen and M. Harjumaa. Persuasive systems design: Key issues, process model, and system features. *Communications of the Association for Information Systems*, 24(1):485–500, 2009.

[22] R. Petty and J. Cacioppo. *The Elaboration Likelihood Model of Persuasion*, volume 19 of *Advances in Experimental Social Psychology*, pages 123–205. Elsevier, 1986.

[23] Banescu S., Posea S., and Calin A. Do you care about my privacy. Coursework report, November 2010.

[24] M.B. Schmidt and K.P. Arnett. Spyware: A little knowledge is a wonderful thing. *Communications of the ACM*, 48(8):67–70, 2005.

[25] P. Tetri and J. Vuorinen. Dissecting social engineering. *Behaviour and Information Technology*, 32(10):1014–1023, 2013.

[26] Viswanath Venkatesh and Hillol Bala. Technology Acceptance Model 3 and a Research Agenda on Interventions. *Decision Sciences*, 39(2):273–315, May 2008.

[27] R. Wash. Folk models of home computer security. In *ACM International Conference Proceeding Series*, 2010.

[28] R.J. Wieringa. *Design Methods for Reactive Systems: Yourdon, Statemate, and the UML*. The Morgan Kaufmann Series in Software Engineering and Programming. Elsevier Science, 2003.