



Title: *A summary of PICOS WP3 sub-phase 3.1 deliverables*

Editors: *José Luis Vivas & Isaac Agudo (Universidad de Málaga)*

Reviewers: *Elsa Prieto (Atos Origin Sae, ES (ATOS))*
Katja Böttcher (Goethe Universität Frankfurt)

Identifier: *D3.4.1*

Type: *Deliverable*

Version: *1.0*

Date: *02.09.2010*

Status: *Final*

Class: *Public*

Summary

Assurance must be an integral constituent of the PICOS solution that should be pursued in a holistic manner. For this reason, in WP3 we adopt a holistic approach emphasizing the relation between the parts and the whole. WP3 gives input to the implementation of the PICOS prototype with respect to privacy and trust by providing an assurance evaluation of the design and its documentation in both sub-phases 3.1 and 3.2 of the project

This deliverable summarizes the three WP3 deliverables produced in sub-phase 3.1: D3.1.1, concerned with the WP4 platform design; deliverable D3.2.1, concerned with WP5 platform prototype; and deliverable D3.3.1, concerned with the WP6 community prototype. We provide a presentation of the following points:

- Which methodology has been used.
- What has been done to evaluate the platform design, the platform prototype, and the community prototype.
- Which specific results have been obtained.
- How these results are taken into consideration in the project.



Grant Agreement no. 215056

- Lessons learnt after this evaluation process.



Grant Agreement no. 215056

Members of the PICOS consortium:

Johann Wolfgang Goethe-Universität (Coordinator)	Germany
Hewlett-Packard Laboratories Bristol	United Kingdom
Hewlett-Packard Centre de Competence France	France
Universidad de Málaga	Spain
Center for Usability Research & Engineering	Austria
Katholieke Universiteit Leuven	Belgium
IT-Objects GmbH.	Germany
Atos Origin	Spain
Deutsche Telekom AG	Germany
Leibniz Institute of Marine Sciences	Germany
Masaryk University	Czech Republic

The PICOS Deliverable Series

These documents are all available from the project website located at <http://picos-project.eu>.

D9.1 Web Presence	February 2008
D2.1 Taxonomy	July 2008
D2.2 Categorisation of Communities	July 2008
D2.3 Contextual Framework	November 2008
D2.4 Requirements	November 2008
D4.1 Platform Architecture and Design	March 2009
D9.2.1 Exploitation Plan v1	April 2009
D9.3.1 Dissemination Report	April 2009
D5.1 Platform description document v1	October 2009
D6.1 Community Application Prototype v1	December 2009
D7.2a First Community Prototype: Lab and Field Test Report	February 2010
D9.2.2 Exploitation Plan 2	March 2010
D9.3.2 Dissemination Report 2	March 2010
D8.1 Legal, economic and technical evaluation of the first platform and community prototype	April 2010
D6.2a First Community Application Prototype v2	April 2010
D5.2a Platform Prototype 2	May 2010

Copyright © 2010 by the PICOS consortium - All rights reserved.

The PICOS project receives research funding from the Community's Seventh Framework Programme.



The PICOS Deliverable Series

Vision and Objectives of PICOS

With the emergence of services for professional and private online collaboration via the Internet, many European citizens spend work and leisure time in online communities. Users consciously leave private information; they may also leave personalized traces they are unaware of. The objective of the project is to advance the state of the art in technologies that provide privacy-enhanced identity and trust management features within complex community-supporting services that are built on Next Generation Networks and delivered by multiple communication service providers. The approach taken by the project is to research, develop, build trial and evaluate an open, privacy-respecting, trust-enabling platform that supports the provision of community services by mobile communication service providers.

The following PICOS materials are available from the project website <http://www.picos-project.eu>.

Planned PICOS documentation

- Slide presentations, press releases, and further public documents that outline the project objectives, approach, and expected results;
- PICOS global work plan providing an excerpt of the contract with the European Commission.

PICOS results

- *PICOS Foundation* for the technical work in PICOS is built by the categorization of communities, a common taxonomy, requirements, and a contextual framework for the PICOS platform research and development;
- *PICOS Platform Architecture and Design* provides the basis of the PICOS identity management platform;
- *PICOS Platform Prototype* demonstrates the provision of state-of-the-art privacy and trust technology to leisure and business communities;
- *Community Application Prototype* is built and used to validate the concepts of the platform architecture and design and their acceptability by covering scenarios of private and professional communities;
- *PICOS Trials* validate the acceptability of the PICOS concepts and approach chosen from the end-user point of view;
- *PICOS Evaluations* assess the prototypes from a technical, legal and social-economic perspective and result in conclusions and policy recommendations;
- *PICOS-related scientific publications* produced within the scope of the project.



Table of Contents

Members of the PICOS consortium:	3
The PICOS Deliverable Series.....	3
Vision and Objectives of PICOS.....	4
1 Introduction	8
2 The methodology	9
2.1 <i>A three-dimensional perspective</i>	<i>9</i>
2.2 <i>Assurance Based Development (ABD)</i>	<i>10</i>
2.2.1 <i>Assurance Case</i>	<i>10</i>
2.2.2 <i>PICOS Assurance Methodology</i>	<i>11</i>
3 Evaluation of platform design, platform prototype and community prototype	16
3.1 <i>The platform design</i>	<i>16</i>
3.1.1 <i>Step 1: The trust and privacy principles</i>	<i>16</i>
3.1.2 <i>Step 2: The PICOS functionality and the use cases.....</i>	<i>17</i>
3.1.3 <i>Step 3: The PICOS architecture components</i>	<i>18</i>
3.1.4 <i>Step 4: The PICOS architecture assurance case</i>	<i>18</i>
3.2 <i>Evaluation of the platform prototype.....</i>	<i>18</i>
3.3 <i>Evaluation of the community prototype.....</i>	<i>19</i>
4 Specific results	20
4.1 <i>Specific results: the platform design.....</i>	<i>20</i>
4.2 <i>Specific results: the platform prototype</i>	<i>21</i>
4.3 <i>Specific results: the community prototype.....</i>	<i>24</i>
5 The impact of assurance in PICOS.....	26
6 Lessons learnt after the evaluation process.....	27
7 Conclusions	28
References	29
Appendix A Trust and Privacy Principles.....	30
Appendix B Relation Between Trust & Privacy Principles and PICOS Principles.....	32



Table of Figures

Figure 1 Structure of an Assurance Case	11
Figure 2. Example of Claim decomposition.....	12
Figure 3. Dependencies in the Assurance Case.....	13
Figure 4 Assurance Cases and System Models	15



List of Acronyms

<i>ABD</i>	<i>Assurance Based Development</i>
<i>CA</i>	<i>Client Application</i>
<i>Dx.y.z</i>	<i>[PICOS] Deliverable: Work Package x, Deliverable y, Cycle z</i>
<i>PDD</i>	<i>Platform Design Decisions</i>
<i>PICOS</i>	<i>Privacy and Identity Management for Community Services</i>
<i>PP</i>	<i>PICOS Principle</i>
<i>PPFSD</i>	<i>PICOS Platform Functional Specification Document of the First Prototype</i>
<i>PrP</i>	<i>Privacy Principle</i>
<i>TrP</i>	<i>Trust Principle</i>
<i>PUC</i>	<i>PICOS Use Case</i>
<i>RPC</i>	<i>Remote Procedure Call</i>
<i>SDLC</i>	<i>Software Development Life Cycle</i>
<i>WPn</i>	<i>Work Package number n</i>



1 Introduction

The present deliverable summarizes three PICOS WP3 [WP3] deliverables: deliverable D3.1.1 [D3.3.1], concerned with the platform design provided in the WP4 [WP4] deliverable D4.1 [D4.1]; deliverable D3.2.1 [D3.2.1], concerned with the platform prototype described in the WP5 [WP5] deliverable D5.1 [D5.1]; and deliverable D3.3.1, concerned with the community prototype described in the WP6 [WP6] deliverable D6.1 [D6.1]. These deliverables were intended as input to other work packages, and contained accordingly a great amount of low level analysis and observations useful for architecture designers and developers. The current deliverable provide an overview of these deliverables that may be useful for people not directly involved in development, and therefore not interested in low level details. It provides also a general introduction for those interested in reading the mentioned WP3 deliverables.

Assurance is intended to be an integral constituent of PICOS, and to be pursued in a holistic manner. WP3 provides an integral solution to PICOS assurance and emphasize the interdependence of its parts. WP3 consists of two sub-phases, 3.1 and 3.2, corresponding to the two sub-phases of the PICOS project. Each of the deliverables in WP3 is produced according to the partial results of the project in sub-phase 3.1, and will be reviewed, updated and extended in sub-phase 3.2.

Deliverable D3.1.1 presents the evaluation of the D4.1 PICOS architecture, its components and functionality, with regard to the trust and privacy principles established in D2.4 and D4.1, and which were refined in order to suit the purposes of assurance. A complete list of these principles may be found in Appendix A in this document, and the relation of these principles to the corresponding D4.1 principles from which they were extracted or derived is shown in Appendix B. We have done an extensive analysis of the relation between, on the one side, the trust and privacy principles of PICOS, and on the other side the nine use cases that specify the basic functionality of PICOS [D4.1 Chapter 13]. We have also analysed the functionality of each component relevant for trust and privacy with regard to what is required of them in the specification of the use cases.

Deliverable D3.2.1 evaluates the documentation and functionality of the D5.1 platform prototype concerning its conformance with the established trust and privacy principles. The main focus of the analysis was on how the privacy and trust principles relate to the WP5 functionality and components, in contrast to WP4, and the differences between the architecture and the prototype are highlighted. The approach is the same as in D3.1.1, except that we now have to do with a slightly different functionality and a different set of components.

Deliverable D3.3.1 presents the outcome of the analysis and evaluation of the trust and privacy functionality of the D6.1 Community Application Prototype 1. The main focus of the evaluation has been the detection of non-conformances in the specification and implementation of the prototype with respect to the established PICOS privacy and trust principles. In contrast to the previous WP3 deliverables, there is no component analysis in D3.3.1. The platform is regarded in this document as an external component.

We provide below a presentation of the following points:

1. Which methodology has been used,
2. What has been done to evaluate the Platform Design, the Platform prototype, and the Community Prototype,



3. Which specific results have been obtained,
4. How these results are taken into consideration in the project,
5. Lessons learnt after this evaluation process.

Below, a chapter is dedicated to each one of these points.

2 The methodology

System development might be viewed as a process of continuous refinement and decomposition of a system, from a more abstract view to increasingly more detailed and concrete ones. If the main facts for the assurance argument are produced during the design process itself, then this process of decomposition and refinement should be at the heart of the assurance case. Our methodology is intended to put into test this conjecture, and to provide a means to enable going from low level to high level properties, therefore facilitating traceability and maintainability.

2.1 A three-dimensional perspective

The three basic elements in the assurance evaluation of PICOS are *principles*, *use cases*, and *components*. *Principles* are the trust and privacy requirements that PICOS should enforce; *use cases* denote the basic functionality of PICOS; the *components* are the elements or functional units (abstract or real) responsible for realizing the functionality of PICOS through their behaviour and in mutual interaction. We analyse the principles in the light of the use cases. A use case may contradict, enforce, or not be relevant to a principle. At the highest level of abstraction, a use case involves only one component, the system itself, which is seen as a black box, i.e. a system viewed solely in terms of its input, output and transfer characteristics without any knowledge of its internal workings. However, as the system is refined and decomposed, several internal elements are defined whose behaviour should realize the initial basic functionality. At this stage, the components are also analysed, always in the light of the established use cases, whose functionality they should implement, and with relation to the principles that are relevant for the use case in question. The end product of this process should be an assurance case that gives documented evidence that the principles are enforced in the context of the use cases and the components involved in them.

It is easier to understand most assurance commentaries provided in the WP3 deliverables by viewing them in the context of points or planes in a 3-dimensional space with coordinates of type (*principle*, *use case*, *component*). The principles and use cases can be seen as relatively fixed and are commonly established at the initial stages of the software development process, but the set of components varies and becomes more refined along system development. Hence, at the highest, most abstract level, we see the system as a single component, in interaction with external agents. This corresponds to the traditional view of use cases, where the system is seen as a single component, a black box. We can regard system development as a continuous activity of decomposition and refinement of components. The functionality of a given component in a given abstraction level is determined by the role it plays in the use cases in which it takes part. For each pair (*use case*, *component*) we analyse how each



principle is affected. Many times there is no relation between the pair and a determined principle, in which case no comments are necessary. Often we implode one of the dimensions in order to establish more general remarks. For instance, if in a determined point (*principle, use case, component*) we implode the second dimension, *use case*, we would obtain a pair (*principle, component*). The observations corresponding to this pair would concern the relation between the given *principle* and the given *component* in ALL use cases; in other words, with regard to the overall system functionality.

The assurance commentaries can be ordered starting along any of the three dimensions. Each of the three ways of ordering the assurance commentaries (by principle, use case, or component) can be useful for different purposes. For instance, if the information is ordered by principle, it would benefit e.g. people doing legal analysis or interested in understanding how well the principles are enforced by the system as a whole; if it is ordered by component, the information could be most interesting for component developers, who may be able in this way to see which principles must be taken into consideration; finally, if ordered by use case, the information may benefit e.g. business process analyst. Moreover, once a given dimension has been chosen, we still could organise the rest of the information along any of the two remaining dimensions. The need of tools for managing this complexity becomes evident here.

By keeping in mind this topology it is probably easier to understand most many observations in each deliverable. It is also important to point out that these observations are usually very specific in nature and related to the description of the components and their functionality in the input deliverables, i.e. in D4.1, D5.1 and D6.1. They cannot be understood without these descriptions. The idea is to make these observations are in general useful at first hand for developers in their work, not primarily for external reviewers. This conforms to the guidelines of the Assurance Based Development methodology, explained in the next section, in which assurance should be part of the development process itself and not only an evaluation of the final system for external reviewers. As such, the deliverables must be regarded basically as **tools** for the development of the PICOS platform and community prototypes, not as final products intended only for external reviewers.

2.2 Assurance Based Development (ABD)

In the evaluation of the trust and privacy aspects of PICOS we decided to adopt a novel Assurance Based Development (ABD) approach based on *assurance cases*. The idea is to integrate assurance and system development by letting the different stages of the system development life-cycle be reflected in the structure of the assurance case, turning assurance cases themselves into a system development tool.

2.2.1 Assurance Case

A security assurance case is a structured collection of security-related claims, arguments, and evidence. It presents an argument showing how a top-level claim is supported by objective evidence. Assurance cases typically consist of at least three parts:

Claims: embody what is to be shown.

Arguments: show how a top-level claim is supported by subclaims and ultimately by evidence.

Evidence: can be regarded as claims that do not require further argument; may include testing, code review, formal mathematical proofs, arguments about the nature of the development process, the reputation of the development organisation, and the trustworthiness of the developers, among others.

The structure of an assurance case may be illustrated as in Figure 1.

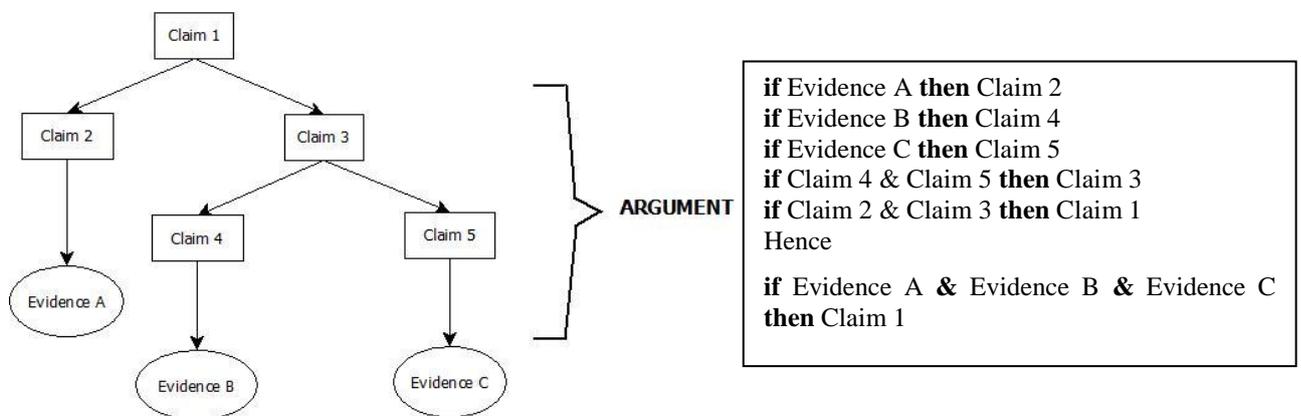


Figure 1 Structure of an Assurance Case

In PICOS we distinguish between two kinds of refinement for deriving subclaims from a claim. In the first one, subclaims follow logically from the parent claim; in the second, the subgoals represent basically countermeasures supporting the fulfilment of the parent goal. Subgoals to functional requirements are often of the first kind, whereas subgoals of security requirements, often in the form of a negated assertion, are commonly of the second type.

We allow for multiple inheritance in assurance case trees. The reason is that a claim about a low level entity, for instance a software module or component, may affect several higher level claims or system goals. For instance, a certain cryptographic mechanism may be used to ensure that different goals, such as confidentiality and integrity, are achieved.

2.2.2 PICOS Assurance Methodology

The starting point for our assurance methodology [VAL10] is the thesis that if the properties of a system are related to the properties of its components, then the structure of an assurance case should somehow reflect the structure of the system. An initial claim about a system in an assurance case might be viewed as an assertion about certain properties of this system, and as a result a property associated with a component of this system might be seen as a subclaim to the initial claim about the

system, viewed as a whole. In this way, a hierarchy of claims emerges, encompassing different levels of abstraction and different phases of system development, therefore facilitating the important property of traceability.

The methodology does not necessarily assume a specific development process methodology, but it does require the use of some more or less rigorous development process. A use-case driven development methodology would probably be the most suitable.

A typical Software Development Life Cycle (SDLC) process may include the following phases:

1. Requirements
2. Architecture and design
3. Development
4. Testing
5. Deployment

A security risk management process within the SDLC, on the other hand, may include the following:

1. Security requirements specification and risk assessment
2. Security architecture and design
3. Secure implementation
4. Security testing
5. Secure deployment and assurance

Ideally, one phase follows from the previous one according to choices made with regard to what was established in the previous phase. In a well structured development process, each phase would yield a certain representation of the system. Our starting point is therefore the conjecture that, if each phase follow from the previous phase in a well determined way, then a claim made concerning a feature in the representation of the system in a given phase must correspond in some sense to a claim about one or more features of the system in the ensuing phase, at least in a well structured development process.

Figure 2, extracted from [GLW07], can be used to illustrate one important point in this context.

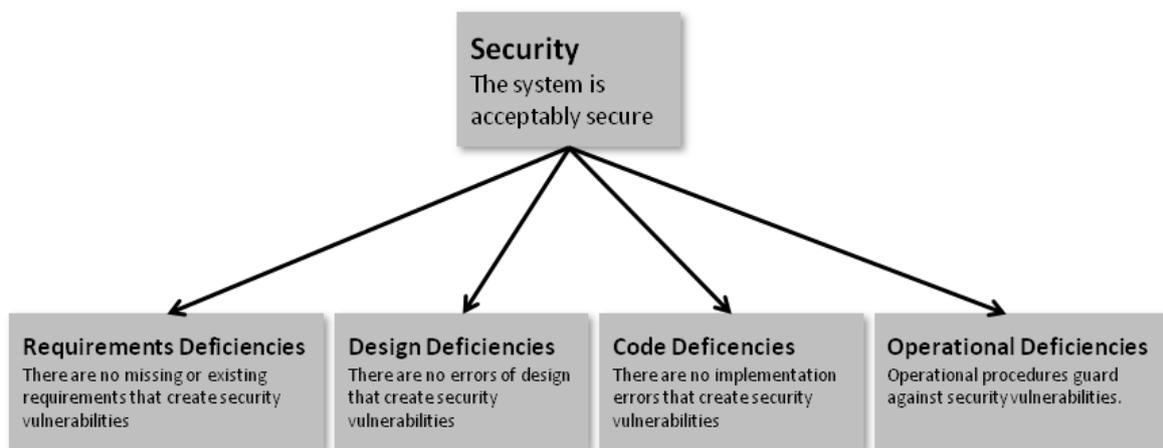


Figure 2. Example of Claim decomposition

As we may see from this figure, the claims concerning the different stages of development are not related to each other. Hence, the claim concerning coding, “Code deficiencies” in the figure, says simply that “there are no implementation errors that create security vulnerabilities.” This claim cannot be related, for instance, to the claim that there are no design errors, in “Design Deficiencies” at the same level in the tree. Although this example was created only for a tutorial purpose, it possibly gives us a good illustration of the current state of the art of assurance. In a real-world assurance case, we may believe that some parts of the code have been better-tested than others, and that the level of confidence that may be put on correctness of some part of the code or module is never absolute. However, which higher-level claims or requirements would this part of the code affect? We can assume that a given code component is related to a certain set of requirements, but not to others. Hence, code concerning cryptographic mechanisms may probable affect a confidentiality or privacy requirement, but maybe not availability. However, there is no way to read this from this assurance case structure.

Figure 3, on the other hand, shows possible dependencies among the claims of the previous figure. What we need therefore is an assurance case where errors or even changes made to any element at a certain level of system abstraction may be traced back to relevant requirements or claims at a higher level. In other words, we want to ensure traceability.

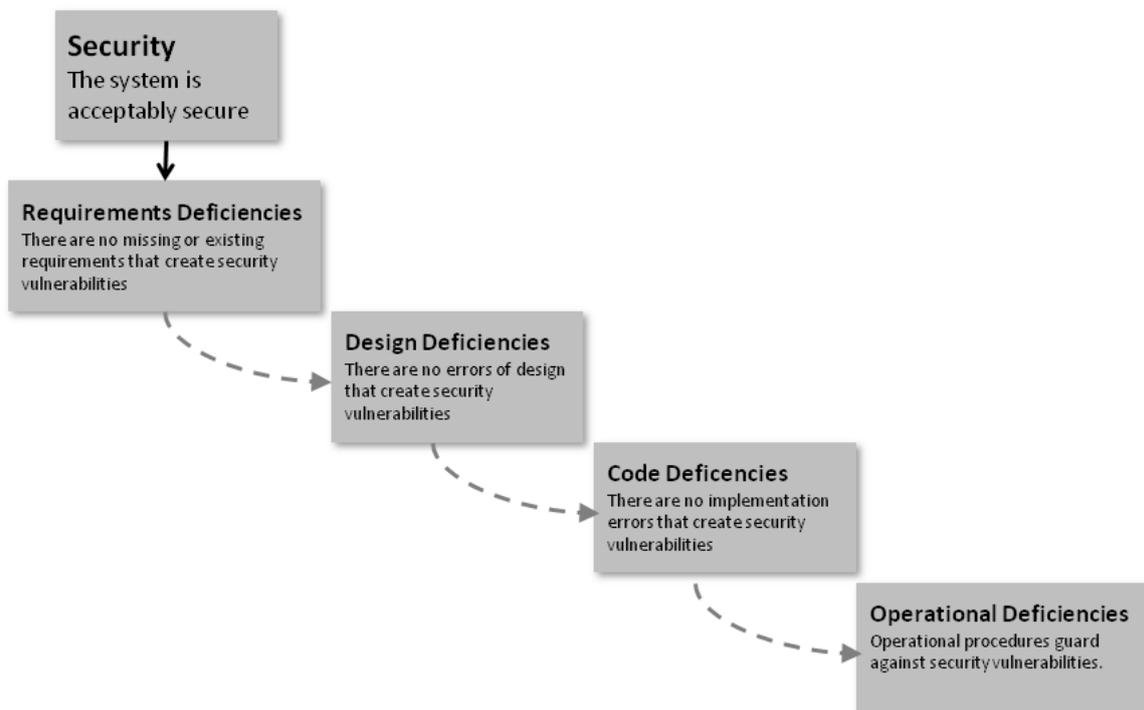


Figure 3. Dependencies in the Assurance Case

At the higher levels of an assurance case, the relationship between claims and the system representation or development phase is not hard to establish. We simply let the system goals



themselves become the high level claims. For other phases of development we adopt a vulnerability-based approach which will be explained below. Hence, in the context of assurance cases the requirements are turned into claims, the only difference being that whereas a requirement is typically stated in the subjunctive mode (e.g. “confidentiality SHOULD BE enforced by the system”), a claim is enunciated in the indicative mode (e.g. “confidentiality IS enforced by the system”).

What we are proposing amounts therefore to an integration of security engineering and ABD with the help of assurance cases. The idea is to integrate assurance and system development by letting the different stages of the system development life-cycle be reflected in the structure of the assurance case, turning assurance cases themselves into a system development tool. This is done first by turning the high level requirements and system goals into the assurance claims. These claims are followed by lower level ones making assurance claims that reference design and architectural concepts. At a still lower level, nodes may correspond to assurance claims making reference to implementation and deployment concepts.

Figure 4 illustrates the relation between the structure of an assurance case, as we envisage it, and the system model structure. A system development strategy might involve the phases shown in the right side of the figure: a requirements phase, a design phase, an implementation phase, and a deployment phase. In the figure, the distinct phases are separated by curved lines, and relations between entities at different levels are shown by arrows crossing this line. Typically, at each phase there is a set of entities used to define the system model at the corresponding level of abstraction. Hence, using a notation similar to UML and a use case driven development methodology, the requirements phase might consist of actors, documents, use cases, etc. Three use cases were included in the figure: *UC 1*, *UC 2*, and *UC 3*. The design phase might on its turn consist of system components. Three components are shown: *CP 1*, *CP 2*, and *CP 3*. Each one may take part in the realization of the functionality described in one or several use cases. In the figure, for instance, *CP 1* is shown to take part in the development of *UC 1* and *UC 2*. The implementation phase might consist of classes and packages, each one realizing the functionality of one or more components. Finally, at the most concrete level, the system model might consist of devices like mobile phones, PDAs, servers, code, etc.

On the left side of the figure we see the corresponding assurance case. The structure consists of a main claim (e.g. “the system is acceptably secure”), four subclaims (*Subclaim 1* to *Subclaim 4*, e.g. “confidentiality is enforced”, “integrity is ensured”, “authentication is guaranteed”, etc). These claims are related to high level requirements. More specific high level claims could also be included here as subclaims to any of Subclaims 1 to 4, for instance “the authentication of *Actor 1* in *UC 2* is guaranteed,” or “the integrity of *Doc 1* in *UC 3* is enforced.” Claims are thus predicates about entities in the corresponding system model, and eventually in higher level system models, but not in lower level ones.

The lowest level claims at the requirements phase are then decomposed into subclaims that belong to the design phase of development, and accordingly refer to entities that belong to the design model, e.g. *CP 1*, *CP 2*, and *CP 3*. For instance, *CP 2* could be a PKI-module, *Subclaim 1* could be “authentication is enforced,” and *SubC 1.2* could be “PKI is used for authentication.” *CP 2*, in this case, could be a PKI module used in order to provide authentication in *UC 2*. Relationships between claims and system model entities are denoted by links in the figure.

This procedure is repeated until we reach the lower level claims, corresponding to the deployment phase, with entities such as mobile phones and servers. For instance, on the system model part of the figure we see a mobile phone *MP*, which is related to *Class 1* and *Class 2* above (e.g. by being an

instance of these classes). We show in green all the entities at higher level system models that are directly or indirectly related to the *MP*. Likewise, we show in the assurance case all the claims that are directly or indirectly related to this entity. Claim *SubC 1.2.1.2.1* is immediately related to *MP*, which means that this claim predicates something about *MP*, for instance that it stores public keys or is able to encrypt any outgoing messages. Going up through the claim hierarchy, we see that claim *SubC 1.2.1.2.1* is a subclaim to *SubC 1.2.1.2* and *SubC 4.1.1.1*, which on their turn are subclaims to *SubC 1.2.1* and *SubC 4.1.1*, and so on. Changes in *MP* could in principle have an impact on all these claims in the assurance case. With the aid of a suitable metrics giving a specific weight to each subclaim of a given claim, it might become possible to establish the level of impact of a determined change in any entity of a system model on the validity of any dependent higher level claims. Moreover, the links between the claims in green on the left side of the figure, and the entities on the right, also in green, would allow us to cross-check the relationships between the assurance case and the system models in case of any changes. This picture shows that a rich and complex set of dependencies is obtained that could be of great help in the analysis of the impact of any choices related to the system entities, and whose complexity might become manageable with the aid of dedicated tools.

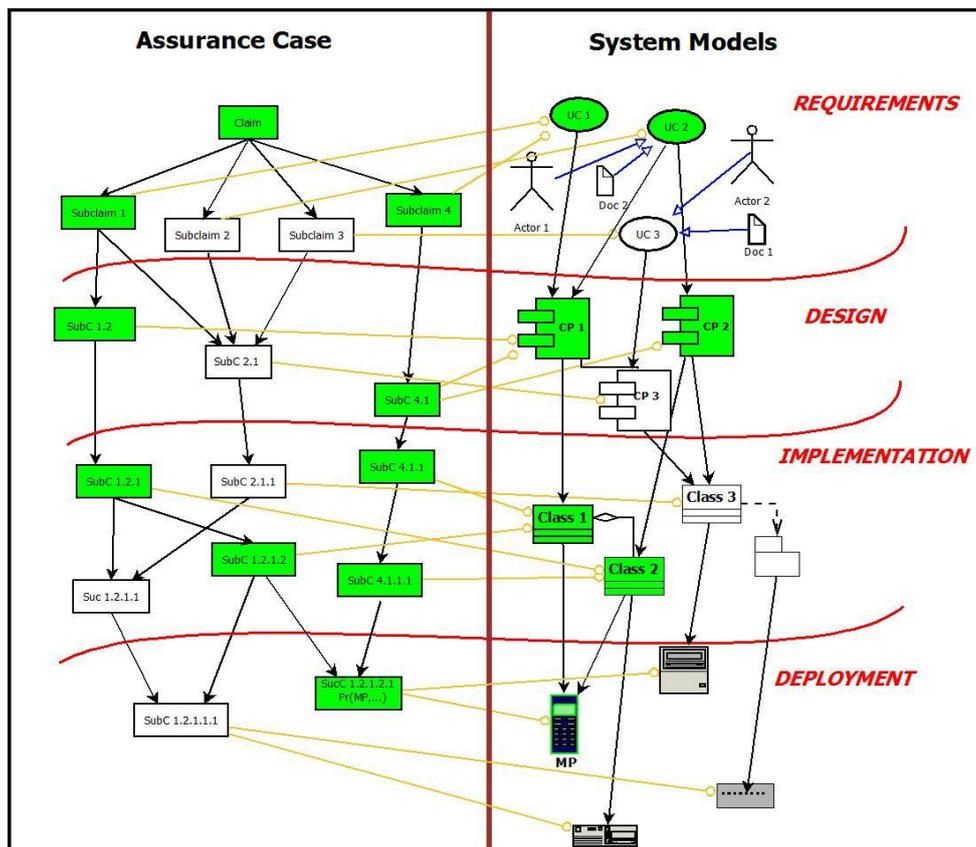


Figure 4 Assurance Cases and System Models



3 Evaluation of platform design, platform prototype and community prototype

In this chapter we present the evaluation of the platform design [D3.1.1], the platform prototype [D3.2.1], and the community prototype [D3.3.1]. A section is dedicated below to each one of these topics.

3.1 The platform design

We required basically three elements from the architecture in order to elaborate a first assurance evaluation of it:

- The trust and privacy requirements of PICOS (provided in D4.1 Chapter 7)
- The basic (architecture-free) functionality of the PICOS system
- The PICOS components, agents or functional entities that implement this functionality (provided in D4.1 Chapter 13)

The basic steps of our approach were the following:

1. investigate the given requirements with the purpose of establishing a set of well-defined high-level trust and privacy principles (called claims in the context of assurance cases) suitable for further assurance work (Chapter 3 in D3.1.1)
2. investigate the functionality of PICOS, as described in use cases, with regard to the privacy and trust PICOS principles established in step 1, searching for eventual vulnerabilities (Chapter 4 in D3.1.1)
3. analyse the functional components intended to support the functionality of the system with regard to the results of step 2 (chapter 5 in D3.1.1)
4. build the (partial) assurance case tree (chapter 6 in D3.1.1)

The trust and privacy requirements are also called *principles* in PICOS. Hence, we use both terms indistinguishably below.

3.1.1 Step 1: The trust and privacy principles

Before analysing the platform design as such, our first task was to analyse the established PICOS privacy and trust requirements and put them in a hierarchical fashion suitable for further assurance work. These requirements became the high level goals or claims in the assurance case. The claims



were iteratively decomposed into lower level claims until we reached a level of abstraction that was considered appropriate for further assurance analysis of the functionality and the components of the system with regard to privacy and trust requirements (basically a vulnerability or threat analysis).

The need to decompose the initial claims followed from the fact that some of the initial requirements were too abstract and in need of further refinement. For instance the requirement that PICOS must be compliant with all legislation, regulation and best practices that exist in the geographical regions in which the Community operates. A more concrete claim such as “*Notice is provided to the Data Subject of the purpose for collecting personal information and the type of data collected,*” which is a consequence of the previous claim since it is required by the legislation, is more useful. Altogether we obtained 24 privacy principles for PICOS. Together with 8 principles related to trust, we obtained a set of 32 privacy and trust principles that were turned into claims for the PICOS platform. These claims were intended to guide the assurance work during the whole PICOS development lifecycle. The principles are presented in Chapter 3 in D3.1.1.

3.1.2 Step 2: The PICOS functionality and the use cases

The functionality of PICOS was provided in the shape of use cases, which was a request also from the assurance team to PICOS system designers. Nine use cases were provided. However, these use cases involved components of the PICOS architecture, but for a better understanding of the PICOS functionality we needed as a first step to analyse these use cases without reference to architectural concepts. The solution was to extract the component-free functionality of the system from the description of these use cases, and then to investigate how the resulting functionality affected the PICOS trust and privacy principles. Thereafter we investigated whether the principles were supported by the specified functionality of PICOS. Since the functionality of the PICOS platform is extensive and cannot be established in a single step, we decided to focus exclusively on the use cases provided, as these were intended to describe the basic functionality of PICOS. The approach was to investigate each of the 32 privacy and trust principles in the light of each use case. Next, for each principle it was established whether and how it was relevant to each use case. A principle was considered relevant for a use case mainly for two reasons: (i) the use case displayed a functionality that supported the principle; (ii) the use case contradicted the principle in question or exhibited some form of vulnerability that could lead to a state in which the principle could be negated. The procedure involved a kind of threat and vulnerability analysis. The result was presented in the form of a 2-dimensional matrix whose rows corresponded to the principles and the columns to the use cases. A mark in any of the slots of the matrix denoted that there was some kind of relation between the corresponding use case and principle. This may be viewed also as the case were we have only one component, the system itself, and the 3-dimensional space (*principles, use cases, components*) may be seen as isomorphic to a 2-dimensional one of type (*principles, use cases*). A short argument was also given for each existing mark. This was the starting point for the vulnerability analysis, and in general for all further assurance work.



3.1.3 Step 3: The PICOS architecture components

Once the relation between the use cases and the principles was established, the subsequent step was to focus on the architectural aspects and investigate the functionality of the components included in each use case with regard to the established principles, which yielded again a 3-dimensional space. This was done by concentrating on each principle at a time (the first dimension). For each principle, we focused on each of the use cases (the second dimension) to which the specific principle was considered to be relevant, i.e. there was a mark in the matrix. Next, we analysed the provided functionality of each component (the third dimension), as described in the use case, searching for properties that would be relevant for the principle in question. Each principle is therefore related to each relevant use case, and the components involved in the use case and regarded to be relevant for the principle are listed and analysed. Sometimes, a component not mentioned in the use case but deemed relevant for the principle was also included. For each listed component a detailed analysis is provided concerning the role it played in the use case with regard to the given principle. This corresponds to a single point in the 3-dimensional space. We focused on questions such as whether the described behaviour of the component was consistent with the behaviour required by the use case, whether any functionality was missing, and similar issues.

This work resulted in a series of observations about the components, their functionality and dependencies. Several omissions were detected, as well as discrepancies between the description of the components and their behaviour in the description of the use cases. These observations constituted at this stage the main input of the assurance work to developers for the subsequent stages of development of the PICOS platform.

3.1.4 Step 4: The PICOS architecture assurance case

The claims at highest levels of the PICOS Assurance Case tree correspond to the trust and privacy principles presented above. The strategy at this stage for deriving subclaims from claims was the logical decomposition of high level privacy and trust principles into more concrete ones. The strategy for deriving *architectural claims* from the principles, on the other hand, was based on the study of the provided use cases and on an informal but careful threat and vulnerability analysis. Subclaims in this case should consist mainly of countermeasures to the observed threats.

The assurance case structure obtained in this way is intended to suffer changes, adjustments, refinements and extensions throughout the system development life cycle of PICOS. The assurance case tree is intended be completed at its lowest levels only when evidence arises at later stages of development.

3.2 Evaluation of the platform prototype

We analysed the platform prototype with relation to the architecture presented in D4.1. The approach is basically the same as in D3.1.1, except that we had to deal this time with a slightly different functionality and a distinct set of components. The results were presented in D3.2.1.



At the beginning there was a gap between the specification of the components in the platform design and the first PICOS prototype. There was little information about the design components that were selected for implementation, the rationale for this selection, and how they had been implemented in the first version of the prototype. In consequence we initially urged developers to fill this gap by providing additional information on the implementation of the design components in the first PICOS prototype. As a result, an internal document was produced to this end, the *WP5 Platform Design Decisions* [WP5 PDD], which became the main focus of our analysis. We regarded the architecture [D4.1] as the requirements for the platform prototype, and stressed the fact that the evaluation of the architecture would be fruitful only if its features were clearly associated with the features of the platform prototype. Assurance Based Development requires that the distinct phases of development are closely related.

Apart from [WP5 PDD], the more extensive *PICOS Platform Functional Specification Document of the First Prototype*, which became the WP5 D5.1 deliverable [D5.1], was also considered. This deliverable deals only with the external functional specifications, and is therefore more difficult to evaluate at the present stage, since what we have here is basically only the signature of the external interfaces, not a description of the components of the platform prototype, their behaviour and interactions.

The differences between the architecture design and the PICOS first prototype were highlighted in D3.2.1. In D3.1.1 we analysed the functionality of the system, as specified in the WP4 use cases, with regard to the trust and privacy requirements of PICOS. In D5.1 we have a new description and a partial redefinition of these use cases, in the form of Call Flows. The list of components was reduced in relation to D4.1, and the functionality was redefined in several ways. New components were introduced, and the functionality of some of the preserved components was altered. The relationship between the WP5 prototype and the WP4 architecture and design was described in [WP5 PDD].

In the same way as for the architecture, an analysis of the components included in the first platform prototype was carried out. We documented also the changes that were made with relation to the components specified in WP4, and how the functionality of WP4 components was implemented by WP5 services. Furthermore, we gave also an account of how this implementation affected the privacy and trust principles that were considered to be relevant for each specific component. The approach was basically the same as in the case of the platform design, described in the previous section.

3.3 Evaluation of the community prototype

The main purpose of assurance with regard to the community prototype was to analyse and evaluate the trust and privacy functionality of the community prototype 1, described in D6.1. The focus of the evaluation was the detection of non-conformances in the specification and implementation of the prototype with respect to the established PICOS privacy and trust principles. The specification and implementation of the PICOS community prototype 1 was assessed with regard to the initial set of trust and privacy requirements. The results are provided in D3.3.1. The assurance of the prototype in PICOS has evolved as a continuous interaction between the assurance and the developer teams, and the present deliverable includes only the final results of this interaction, not its development.



In D3.1.1 we analysed the functionality of the system, as specified in the D4.1 use cases, with regard to PICOS trust and privacy requirements. This analysis was refined, but not basically altered, with the study of the call flows given in D5.1.

In contrast to the previous WP3 deliverables, components are not included in D3.3.1, since these are part of the platform. The platform is therefore regarded in this document as a black box, and only its external functionality was considered. We are hence back to the case of a 2-dimensional space, or a 3-dimensional one in which the last dimension consists only of a single component, in this case the PICOS platform prototype. We can thus view the community prototype as an implementation of the use cases. Basically, this work concentrates on how the prototype implements the functionality provided by the platform with regard to PICOS privacy and trust requirements.

We focused therefore on the functional descriptions of the community prototype, presented in sections 4 and 5 of D6.1. We analysed this functionality in the light of the nine defined PICOS use cases (and a tenth undefined one related to data management which we suggested to include in the description of the platform design functionality) and with regard to the trust and privacy principles relevant for each case, as established in Chapter 3 in D3.1.1.

The assurance of the prototype was more straightforward and less complex than assurance of the platform, since no design choices should in principle be made here. What we did was basically to check if the functionality and protection mechanisms offered by the platform were correctly used. Hence, our task was basically to check if some privacy or trust enhancing functionality provided by the platform was omitted or used deficiently by the prototype.

4 Specific results

The purpose of Assurance Based Development is not to provide an overall evaluation, a certificate or a degree of confidence for a system. There are thus no general results so far; all results are specific, and consist basically of detailed remarks and observations intended to help developers to improve the system under construction with regard to the trust and privacy requirements. The idea is that developers use these remarks to correct eventual faults, to highlight system features that has been eventually overlooked, and in general to consider low level aspects and requirements of security, privacy and trust very early in the system development life cycle.

4.1 Specific results: the platform design

We have carried out a detailed evaluation and analysis of the provided PICOS Architecture, refined the initial trust and privacy principles [D3.1.1 Ch.3], and done an extensive analysis of the relation between, on the one hand, the trust and privacy principles of PICOS, and on the other hand the nine use cases specifying the basic functionality of PICOS [D3.1.1 Ch.4]. As a result of this analysis, we have found that some important functionality, especially concerning the management of data and profiles, has not been covered by the specified uses cases, and have proposed the specification of a new one for this purpose. We have next analysed the role played by the components of PICOS with regard to the established trust and privacy principles and in the context of each of the use cases [D3.1.1 Ch.5]. We have also studied the functionality of each component relevant for trust and privacy with regard to what is required from them in the specification of the use cases. Several gaps and omissions have been pointed out. Finally, we have built the assurance case with the aid of the findings concerning the components and their relation to the trust and privacy principles in the context of the

specified use cases [D3.1.1 Ch.6]. The result is an assurance case intended to be extended, refined and updated during the whole lifecycle of the PICOS project.

Examples of this analysis are detailed observations concerning how the components of the architecture support the principles. For instance, the analysis concerning the principle *PrP9 Limitation of Collection* illustrates well the character of the assurance process developed for PICOS. The idea was to offer low level observations, remarks and analysis that might be useful early in the process of system development.

The relevance that each component in every use case might have with regard to this principle was analysed [D3.1.1 Sect. 5.2]. For instance, the *Limitation of Collection* principle was regarded to be important for the Registration use case. However, this feature was not mentioned in the description of the use case given in [D4.1]. We included therefore the following remark:

No mention in the registration component about limitation of collection (data minimisation). Limitation of collection should be enforced by the registration process, eventually by the Registration component itself [D3.1.1 Sect. 5.2.9.1, p.85].

As may be seen here, this observation is basically intended as a warning for developers to consider this issue, especially in connection with the development of the registration component. Moreover, data collection could take place in other contexts than during registration, something which is not reflected in any of the use cases. We recommended therefore the definition of a new case concerned with data management, and suggested that this principle should be considered within the context of this use case in case personal data is collected at any other time than during registration.

The other principles were analysed in the same way. Moreover, the use cases by themselves were also analysed in the light of the distinct relevant principles, and similar low level observations and remarks were provided [D3.1.1 Ch. 4]. For instance, with regard to use case 4, Multiple Partial Identities, we raised the issue that the principle PrP 10, which requires that personal data may be collected only by fair and lawful means, could come in conflict with the building of profiles for partial identities. Hence, we wrote the following:

Collection of profile data during creation of new partial identities must not involve collection of personal information, as this would contradict the principle that personal data must be collected fair and legally. It is not appropriate to collect personal data during the creation of a partial identity. [D3.1.1 4.4.3.1]

The intention here is to tell developers that this feature must be taken into consideration during the development of the functionality for building partial identity profiles. Letting these issues be highlighted already at this stage of development might increase assurance that the principle is respected in the final system.

4.2 Specific results: the platform prototype

The most important results in the analysis of the platform prototype concern the assessment of the provided call flows or use cases [D3.2.1 Ch.3]. The results are extensive and very specific, including the analysis of each included component, mention of excluded components from WP4, an assessment of the functionality of those components, and their relation to the specific PICOS trust and privacy principles.

For each call flow provided, the normal sequence of interactions was specified in order to enhance common understanding and to build a basis for further analysis. Next, a detailed analysis of the resulting sequence was provided, including a flowchart, similarly to the analysis of the use cases in the platform design, explained in the previous section. Thereafter the role played by the included components was analysed, and detailed commentaries were provided. A comparison was also made between the specification of the use case functionality in the platform design and the corresponding call flow in the platform prototype. Next, an analysis was given of each principle relevant for the call flow in question. This is done for each provided call flow.

An example will be used to illustrate it, the *registration call flow*. First, the normal sequence of interactions in the call flow is specified in order to enhance common understanding and to build a basis for further analysis. Then a detailed analysis is carried out. The results of this analysis are reproduced below [D3.2.1 Ch. 3.1.1.1]:

Some questions remain to be clarified in this use case. Steps 3-6, concerning the terms and conditions, are missing in the sequence diagram of the register call flow [WP5 PDD p.29]. Also, the end user orchestration functions performed by client application prototype are not shown. This functionality is important with regard to trust. Although the client application is considered to be trusted here, it is not clear how this could be enforced in the future. This is especially important for the registration process, in which the sequence of actions should be enforced and both the user and community be made accountable for the actions performed during registration. Many important details about these issues should be clarified.

An important deviation from the WP4 specification is the decision that the partial and root identities will be used only internally. The importance of this alteration in the original conception with regard to trust and privacy issues should be analysed further.

Subsequently, the role played by the included components is analysed. For instance, concerning the Registration component, we made the following observations [D3.2.1 Ch. 3.1.2]:

The registration functionality is declared in the WP5 Platform Design Decisions document to be implemented “though a combination of the Registration service, the Partial Id service, and the Public Community service.” [WP5 PDD 3.1] However, we note that neither the Partial Id service nor the Public Community services are included among the components of the Registration call flow.

As noted in D3.1.1 [D3.1.1 5.2.1.1], notice of collection should be added to the registration process, eventually performed by the Registration component. This takes place in the platform “before the final registration of a user as the first screen before data collection” [WP5 PDD 6.1]. According to [WP5 PDD 6.2], policy notification is “displayed before the final registration of a user as the first screen before data collection.” Limitation of collection and collection by fair and lawful means during registration should be enforced with the help of the Registration component [D3.1. 5.2.9.1 and 5.2.10]. Safeguards to ensure secure communication, as recommended in [D3.1.1 5.2.18.1], is not included in the current description of the register call flow. Moreover, ways to provide guarantees that declared personal information is accurate [D3.1.1 5.2.19.1] are not provided, since this can only be done at the application level.

Next, a comparison is made between the specification of the Registration functionality in the platform design, and that of the platform prototype. The level of detail offered defies summarisation, so we reproduce the text here [D3.2.1 Ch. 3.1.3]:

In WP4 PUC 1, a prospective member approaches the community and presents to the Registration component some kind of identification evidence (prior registration with a real-world community, etc.)

[D4.1 13.1.3]. This feature is included in the register call flow. Notification by the Policy Management of what is considered an acceptable form of registration (step 2 in PUC 1) is also lacking.

In WP4 PUC 1, a root identity is thereafter created within the Registration component. In the register call flow the root identity is apparently created by the Authentication component, which is missing in PUC 1, and communicated to the Registration component.

Thereafter the user profile is created, both in WP4 PUC 1 and in the WP5 register call flow, a procedure involving the Registration and the Profile Manager. However, there is no hint, either in WP4 PUC 1 or in the WP5 register call flow, about how the user is involved in this procedure. In WP4 PUC 1, the Consent Management component is used to set elements of the personal profile, but this component is absent in the WP5 register call flow, its role apparently having been taken over by the Profile Manager.

Thereafter, in the WP5 register call flow there is an interaction between the Registration and the Contact component. The Contact component is not included in D4.1, and it is not mentioned in the description of the PUC1 Registration, except in the sequence diagrams. It can be assumed that its function during the registration is to create an empty list of contacts for the new member.

The Presence component first, then the Location component, are communicated to the root identity of the new member in register call flow. The Location component is not included in WP4 PUC 1.

Finally, in the register call flow the policy and the reputation are set, involving the Policy and the Reputation Manager resp. Both interactions are missing in WP5 PUC 1.

An analysis is also given of each principle relevant for the Registration call flow. For instance, the principle *TrP 8 Member Accountability*, stating that

Member must provide accurate personal information and are accountable for this. If this data is not accurate the community may decide to take actions against the user, like removing them from the community.

is analysed in the following manner [D3.2.1 Ch. 3.1.4.2]:

No mention in the register call flow about how member accountability can be enforced during registration. The fact that part of the registration functionality is performed by the client application might have important consequences for the accountability requirement, although mechanisms to support accountability (i.e. logging, cannot be performed on client side). Further details about the registration functionality must be made available in order to evaluate how this principle is enforced.

Each call flow is given the same treatment.

Next, we focused on the components of the platform prototype, and analyse them, one by one, again at the same level of detail as before [D3.2.1 Ch.4]. Each component was analysed in the light of each use case, and each relevant principle in connection with the use case and the component was studied. What we obtained here was a series of remarks on specific aspects of each component in the context of each use case and with relation to each relevant principle (in other words a point in the 3-dimensional space). The developers of the corresponding component should take this information into consideration when developing the component, together with information about other principles involved. In this way we might ensure that the privacy and trust aspects relevant to this particular component are not disregarded by the developer.

We illustrate the kind of results obtained with the example of the *Profile Manager*. According to our analysis, this component was considered to be related to the following principles: *PrP10 Fair and Lawful Means*, *PrP13 Third-Party Disclosure*, *PrP21 Data Management*, *PrP24 Multiple Persona*,



TrP1 Openness and Transparency, and TrP6 Objective/Subjective Trust. We show an example of our analysis concerning PrP10 Fair and Lawful Means, which was regarded to be related only to use case PUC4 Partial Identities, hence we have to do here with the vector (PrP10, PUC4, Partial Identity Manager) [D3.2.1 Ch. 4.3.1.1]:

The linking of all partial identities to a root identity is enforced by the Partial Id manager in the platform, as well as the restrictions to the ability to link partial identities.

As stressed in D3.1.1 5.2.10.2, the Collection of profile data during creation of new partial identities must not involve collection of personal information, as this would contradict the principle that personal data must be collected by fair and lawful means. It is therefore not appropriate to collect personal data during the creation of a partial identity. For PICOS this implies that personal data that has been disclosed by a member, as part of a partial identity profile or as imported content, shall never be used in the personal profile of the member, i.e. the profile of the root identity, since this would amount to collection of personal information without the consent of the data subject and without having previously given notice of it. Whereas this kind of information need not be accurate, and has been declared by the data subject on his or her own free will and without request, the data subject should be able to delete it at any time, which is not the case with collected personal information.

As can be seen once again, what we have here is a series of remarks on specific aspects of the component *Partial Identity manager* in the context of *PUC4 Partial Identities* and with relation to principle *PrP10 Fair and Lawful Means*: The developers of this component should take this information into consideration, together with information about other principles involved. In this way we might ensure that the privacy and trust aspects relevant to this particular component are not disregarded by the developers.

4.3 Specific results: the community prototype

In the deliverable dedicated to the community prototype, D3.3.1, we presented an analysis and evaluation of the trust and privacy functionality of the community prototype 1, described in D6.1. The main focus had been the detection of non-conformances in the specification and implementation of the prototype with respect to the established PICOS privacy and trust principles. We could establish that the final specification and implementation of the PICOS community prototype conforms in a satisfactory way to the initial trust and privacy requirements, itself a result of the close interaction between WP's 3 and 6 before and during the development of the prototype. The assurance of the prototype in PICOS evolved as a continuous interaction between the assurance team and the community prototype developers, and the deliverable includes only the final results of this interaction, not its development.

The platform on which the prototype is built is regarded in D3.3.1 as a black box, which amounts to assuming the realistic perspective of an application developer using the PICOS platform as a single external component. The confidence in the assurance case of the prototype depends therefore on the confidence in the assurance case of the platform.

What we have here is thus basically a two-dimensional space of type (*principle, use case*), or rather a three-dimensional one as before but with the third dimension collapsed to a single point, the platform. So the analysis was carried out along two dimensions: first, use case by use case [D3.3.1 Ch.2], and then principle by principle [D3.3.1 Ch.3]. After the analysis of each use case, the relevant trust and

privacy principles were studied in the context of the use cases. This kind of analysis was provided for each use case and relevant principle. We give below an example of each one.

In the use case dimension, we consider here *Registration* once again. There were two versions for this use case in D6.1, the second mainly a revision of the first that was carried out as a result of input from the assurance analysis. We show below an analysis of the first version. We derived the main sequence steps of the Registration process according to the description provided. We quote in full our analysis once again [D3.3.1 Ch. 2.1]:

In step 4, is the registration form sent without involvement of Platform, or is there previously some communication between the CA and the Platform in which the latter would provide the community policies to the CA before they are displayed to the User? (In [D6.1 2.2.6] we see clearly that NO communication is taking place between CA and Platform. We believe that the policies should be provided by the platform, which in this way would facilitate updating.)

In step 5, User should accept the policies before filling out the required data; how this happens should be made clear. There is a pair of methods, `getGeneralCommunityPoliciesRequest` and `getGeneralCommunityPoliciesResponse`, allowing CA to retrieve the general policies of the community that are to be displayed and accepted, but when and how this method is called, and when the policies are displayed (if at all) to User, is not clear in this early design. In Figure 1 in [D6.1 C.1] we see that the Picos policy is shown within the registration screen, not before the latter is displayed. (However, in [D6.1 2.2.6] we see that this happens in step 2, prior to the display of the Registration screen, and it is also stated that if User “accepts these terms and conditions then the main registration screen will be displayed where the end-user must enter some mandatory data...” This should be considered, from the privacy standpoint, the right order of events.)

In step 9, it is not clear what the interactions are between Platform and CA; maybe a `registerPartialIdRequest/ registerPartialIdResponse` pair of methods should be used, in analogy to step 7. (The creation of new partial identities is covered by the Partial Identities Management Use Case, described [D6.1 2.2.8]. In [D6.1 2.2.6] we see that no interaction takes place between CA and Platform at this stage. Here, the partial identity is created by User together with the root identity using a single method provided by the platform in its latest version, the `register()` method.)

In Exceptions: the interactions between CA and Platform are not explicitly shown in the exceptions.

It is not clear what happens at step 12 and after.

In general, it is not clear from the description above what are the interactions between CA and Platform. How these interactions are defined may have consequences for trust and privacy. (This was corrected in the latter version [D6.1 2.2.6], where these interactions are clearly specified.)

After this analysis, the relevant trust and privacy principles were studied in the context of the use case. For instance, with regard to *TrP8 Member Accountability*, the following analysis was provided [D3.3.1 Ch. 2.1.1.2]:

Member must provide accurate personal information and are accountable for this. If this data is not accurate the community may decide to take actions against the user, like removing him or her from the community. There is no way to know if user data are accurate, but the PICOS platform administrator is able to remove or disable accounts. Member accountability must be enforced by event logging, which is performed by the PICOS platform.

Finally, an analysis was provided regarding the fulfillment of each principle. For instance, concerning *PrP23 Multiple Persona*, the previous analysis of the use cases provided these findings [D3.3.1 Ch. 3.2.24]:



PrP23 Multiple Persona: *PICOS allows members to have multiple persona.*

The partial identity manager of the community application allows one to create and manage multiple independent identities (partial identities). The principle is enforced throughout in the prototype. Users always interact through one active partial identity

5 The impact of assurance in PICOS

Assurance in PICOS must be seen as a process driven in continuous interaction with the other work packages during the whole software development life cycle, in accordance with the main guidelines of the Assurance Based Development (ADB) approach. Assurance should therefore not be seen as a product until the whole assurance case has been completed at the end of the system development. This is what differentiates the ABD approach from current standard approaches. Hence, strictly speaking there are so far no “results” as such, rather the continuous influence of the process of assurance on the development of the PICOS project, which is documented at determined points in time related to the established milestones in the PICOS development process, especially those related to the two-cycle approach adopted in PICOS. This must be seen not as a result as such, but rather as documented evidence about how the assurance process is being driven, as well as input for developers to the second cycle. However, the most important input is being given on a daily basis and has had a significant influence in the way the development of PICOS is evolving, as explained below.

The interaction between the WP3 assurance team and the rest of the work packages has happened continuously since the beginning of the project, e.g. in the form of teleconferences, presentations and discussion at the plenary meetings and in parallel sessions, emails, and special meetings like one held in Madrid on 16-17th December 2009, gathering people from WP3 and WP6 for a discussion of all relevant issues for the assurance of the community prototype.

One of the main requirements at the beginning of the assurance work from the WP3 team to those responsible for WP4 was that the specification of the functionality of PICOS should be defined in terms of use cases, which were finally adopted to define and validate the functionality of the platform. Since then, use cases have been pivotal in PICOS not only for the description of the WP4 platform design, but also for the WP5 platform prototype, the WP6 community prototype, and of course also for WP3 assurance work. Use cases may be regarded as the backbone of PICOS specification and development.

Another impact has been produced by the documentation itself, which to an important extent has been provided with a level of detail suitable for assurance analysis. On request from the assurance team, an internal document [WP5 PDD], linking more closely the architecture and the platform prototype, was provided within WP5 for the benefit of the assurance effort.

WP6 considered very early the assurance work carried out in relation to WP4 and WP5, which is clearly indicated in the deliverable D6.1:

Other work package results have been important for us, namely from the Assurance perspective (WP3) and the User Trials Plan (WP7) [D6.1, p. 2].

The work will start with a review of the project previous results, directly the Platform Architecture 1(D4.1) and the ongoing Platform Prototype 1 (D5.1 draft), and indirectly their



respective assurance results (D3.1.1 and D3.2.1 draft... The results of the user assessment and PICOS internal assurance and evaluation of the prototype will be a direct input to improve results in successive prototypes in PICOS second cycle. [D6.1, p. 8].

The PICOS prototype construction is directly based on the PICOS platform prototype results which constitute its natural foundation, but also links to WP2 (Requirements), WP3 (Assurance) and WP4 (Architecture) incorporating in an evolutionary process their results [D6.1, p. 9]

Moreover, Chapter 4 in D6.1 has been dedicated entirely to issues of privacy and trust assurance, using WP3 previous work and contributing directly to D3.3.1.

The impact of assurance includes also the re-specification of the Registration procedure, and the decision to make the platform solely responsible for logging the events for auditing purposes.

The deliverables D3.1.1, D3.2.1, and D3.3.1 are also intended to be used as input for WP4 D4.2, WP5 D5.2, and WP6 D6.2 in the second cycle of PICOS.

Due to the close and continuous interaction between WP3 on one side, and WPs 4, 5, 6 and 8 on the other side, it is hard to estimate with precision the level of impact of assurance on the other work packages. However, there is no doubt that it has been significant, and will become even more in the second cycle of PICOS.

6 Lessons learnt after the evaluation process

We have introduced a novel assurance methodology inspired by Assurance Based Development approach, integrating security engineering and assurance with the aid of security assurance cases. The methodology is growing out of our experience in providing trust and privacy assurance to the evolving European project PICOS. It is therefore not a pure theoretical construction. No major practical or theoretical obstacles have been met so far. Therefore, the main lesson for us has been the validation of the proposed assurance approach.

The evaluation process is unfinished yet. In fact, there has not been so far an evaluation process in the strict sense of the word, since evaluation is commonly performed on the final product. The intention behind our holistic approach is to offer a means to ensure that the final PICOS platform architecture and platform prototype are accurate and consistent with the trust and privacy technical objectives planned. The objective is thus not strictly to offer an evaluation of the product under development, but a means to improve the final product. This has been done by detecting, in both the architecture design and the platform prototype, non-conformances that might lead to unexpected consequences or to a final product which does not conform to the established trust and privacy principles and requirements. Not unexpectedly, potential problems have been observed, and the input from the assurance work has decreased the risk that these problems might remain unsolved or undetected until the end of the project. This seems to confirm the view that the introduction of assurance concerns at an early stage of development will contribute to the delivery of a better end product.

The need for a dedicated tool, based on the 3-dimensional perspective, was felt already from the very onset. We were confronted with three kinds of entities: claims, use cases, and components. There were in total 32 claims or principles, 9 use cases, and a score of components. The number of relations among these elements is therefore very high and renders the analysis work almost unmanageable.



Each one of those three entities brings forth a different view of the assurance case. The first view corresponds to the claims (given a certain claim, what is its relation to the use cases, or to the components?); the second one to the use cases (given a certain use case, which principles are relevant to it?); and the third to the components (given a component, what is its relation to the use cases and to the principles?). These are questions that a designer or developer would find appropriate to ask. A tool would facilitate answering this sort of questions. It would also facilitate building the assurance case tree and keeping track of the way one entity is related to the others, for instance which principles are relevant to a determined component, or the other way around, which components are relevant to a determined principle. This would be especially helpful whenever changes are made to either components or use cases, or even claims, helping us in this way to easily update the assurance case tree when changes in any of the entities are introduced.

Finally, the introduction of a metric would greatly enhance the assurance case. Often, a subclaim can be more or less important for the strength of a claim, something which it is not possible to express in our assurance case. Moreover, a claim is seldom simply true or false: what we usually have is more or less evidence or confidence on the validity of a claim. A metric would allow us to express this, and also to track the impact that a certain change in the validity of some evidence might have on the claims that depends on this evidence. It would moreover allow us to decide the level of confidence required from the available evidence in case the amount of confidence on a claim that depends on this evidence is required to be on a determined level. In general, a metric would considerably enhance the quality of an assurance case since it would render it more realistic, enabling us to better capture the inherently fuzzy nature of real word evidence.

7 Conclusions

The assurance methodology adopted has allowed us to take a consistent and uniform approach to every phase of the PICOS development life cycle. The end product will be an assurance case indicating the level of confidence that PICOS complies with the established set of trust and privacy requirements. However, the main contribution is not an end product but rather the development of an assurance process intended to increase the level of confidence that the final system meets those requirements.

Among these requirements, the privacy requirement that PICOS “*must be compliant with all legislation, regulation and best practices that exist in the geographical regions in which the Community operates*” was the hardest to deal with, since this legislation and regulation is extensive. We needed to reduce this principle to a manageable set of more concrete principles that would be suitable for the kind of analysis we intended to carry out. We decided to adopt a decomposition based on the taxonomy provided in [IST07], which resulted in 20 new privacy principles, out of a total of 32 principles for PICOS (24 privacy principles and 8 trust principles). There is no doubt that this was an important decision that would have a big impact on the rest of the assurance work, since it would concentrate basically on this set of principles. However, on 4-6 November 2009, the Madrid Resolution [MAD09], was adopted by data protection and privacy regulators at the 31st International Conference of Data Protection and Privacy, integrating legislation of five continents. We are satisfied to observe that this resolution defined a set of privacy principles which closely matches the one we have adopted, and that for the second cycle we could adopt the principles established in the resolution



without any major alterations in our previous work. This seems to strengthen the reliability of the framework we have adopted in PICOS for privacy assurance.

References

[D3.1.1] Vivas, J. and Agudo, I., “D3.1.1 Trust and Privacy Assurance for the Platform Design”, Final Confidential Deliverable of EU Project PICOS, Apr 2009.

[D3.2.1] Vivas, J. and Agudo, I., “D3.1.2 Trust and Privacy Assurance Evaluation of the Platform Prototype”, Final Confidential Deliverable of EU Project PICOS, Sep 2009.

[D3.3.1] Vivas, J. and Agudo, I., “D3.1.3 Trust and Privacy Assurance of the Community Prototype”, Final Confidential Deliverable of EU Project PICOS, Jan 2010.

[D4.1] Crane, S., “D4.1 Platform Architecture and Design v1”, Public Deliverable of EU Project PICOS, Mar 2009. Available at http://picos-project.eu/fileadmin/user_upload/fmgr/Deliverables/WP4_Architecture_and_Design/D4.1_Platform_Architecture_and_Design_1/PICOS_D4_1_Architecture_v1_4_Final_Public.pdf (last access: Aug 2010).

[D5.1] Kyritiades, L., “D5.1 Platform Prototype 1”, Public Deliverable of EU Project PICOS, Oct 2009. Available at http://picos-project.eu/fileadmin/user_upload/fmgr/Deliverables/WP5_Platform/D5.1_Platform_prototype_1/PICOS_D5_1_Platform_Prototype_1_v1_1_Final_Public.pdf (last access: Aug 2010).

[D6.1] Crespo, A., “D6.1 Community Application Prototype 1”, Public Deliverable of EU Project PICOS, Dec 2009. Available at http://picos-project.eu/fileadmin/user_upload/fmgr/Deliverables/WP6_Application_Prototype/D6.1_Community_application_prototype_1/PICOS_D6_1_Community_Application_Prototype_v1_Final_Public.pdf (last access: Aug 2010).

[IST07] ISTPA International Security Trust and Privacy Association, Analysis of Privacy Principles: Making Privacy Operational, Version 2.0, May 2007.

[GLW07] John Goodenough, Howard Lipson, and Chuck Weinstock. Arguing Security - Creating Security Assurance Cases, Carnegie Mellon University, 2007. Available at <https://buildsecurityin.us-cert.gov/daisy/bsi/articles/knowledge/assurance/643-BSI.html> (last accessed September 19th, 2008).

[MAD09] International Standards on the Protection of Personal Data and Privacy. The Madrid Resolution. International Conference of Data Protection and Privacy Commissioners, 5th November 2009.

[VAL10] José Luis Vivas, Isaac Agudo and Javier Lopez: Security Assurance During the Software Development Cycle: a Model Based Approach. In Requirements Engineering, Volume 15, Number 3, Springer London, 2010 (accepted for publication).

[WP3] Assurance of Technical Trust and privacy properties.

[WP4] Platform Architecture & Design.

[WP5] Platform prototype Development.



[WP6] Communities Prototype Construction.

Appendix A Trust and Privacy Principles

Here follows the complete list of the resulting trust and privacy principles used for assurance:

<p>TrP 1 Openness and Transparency: PICOS offers services that handle personal information in an open and transparent way.</p>
<p>TrP 2 Trust between communities: PICOS recognises trust as a common currency when exchanged between PICOS communities.</p>
<p>TrP 3 Provenance: PICOS ensures that members can rely on the provenance of information.</p>
<p>TrP 4 External services: PICOS ensures that externally hosted services are delivered in a trustworthy way and that members are aware when external services are less trustworthy than internal services.</p>
<p>TrP 5 Audit: PICOS allows processes to be fully auditable by a trusted entity.</p>
<p>TrP 6 Objective/subjective trust: PICOS supports both objective and subjective methods for assessing trust.</p>
<p>TrP 7 Consensus: PICOS guarantees that no single entity can act in a way that might compromise the trust and privacy of the community.</p>
<p>TrP 8 Member accountability: PICOS ensures that Members are accountable for their actions while a member of the Community.</p>
<p>PrP 1 Notice of Collection: Notice is provided to the Data Subject of the purpose for collecting personal information and the type of data collected.</p>
<p>PrP 2 Policy Notification: Data Subject is notified of the applicable policies in terms of Consent, Access and Disclosure.</p>
<p>PrP 3 Changes in Policy or Data Use: Notice must be provided if and when any changes are made to the applicable privacy policies or in the event that the information collected is used for any reason other than the originally stated purpose.</p>
<p>PrP 4 Timing of Notification: The purposes for which personal data are collected should be specified not later than at the time of data collection.</p>
<p>PrP 5 Sensitive Information: Data Subjects must be informed of, and explicitly consent to, the collection, use and disclosure of sensitive information (i.e. medical or health conditions, racial or ethnic origins, political views, religious or philosophical beliefs, trade union membership or information regarding sex life) unless a law or regulation specifically requires otherwise.</p>
<p>PrP 6 Informed Consent: The Data Subject must provide informed consent to the collection of personal information unless a law or regulation specifically requires otherwise.</p>
<p>PrP 7 Change of Use Consent: Consent must be acquired from the Data Subject to use personal information for purposes other than those originally stated at time of collection.</p>
<p>PrP 8 Consequences of Consent Denial: Data Subjects must be made aware of the consequences of denying consent.</p>



PrP 9 Limitation of Collection: Only personal information relevant to the identified purpose may be collected.
PrP 10 Fair and Lawful Means: Information must be collected by fair and lawful means.
PrP 11 Acceptable Uses: Personal Data may only be used for the purposes stated at the time of collection.
PrP 12 Data Retention: Personal Data is retained no longer than necessary to complete the stated purpose.
PrP 13 Third-Party Disclosure: Notice and Consent of the Data Subject is required to disclose information to third parties. The PICOS architecture must uphold the member's wishes with regard to information flow.
PrP 14 Third-Party Policy Requirements: Organizations must ensure that any third parties are informed of their privacy policies and will follow them or possess equivalent policies.
PrP 15 Access to Information: Data Subjects are able to determine if an organization maintains data on them and should be able to request access to said information.
PrP 16 Provision of Data: Requested information is provided clearly, at reasonable cost and within a reasonable timeframe.
PrP 17 Correcting Information: Data Subjects are able to update or correct personal information held by the organization.
PrP 18 Safeguards: Organizations must be sure to include safeguards to prevent loss, misuse, unauthorized access, disclosure, alteration and destruction of data.
PrP 19 Data Accuracy: Organizations will ensure that all personal information is accurate, complete and kept up-to-date.
PrP 20 Public Policies: An Organization must ensure that its privacy policies are clearly published and publicly available.
PrP 21 Data Management: PICOS must allow members to express how to store and process their data and uphold their wishes in this regard.
PrP 22 End-to-end privacy: PICOS supports end-to-end privacy.
PrP 23 Authentication: PICOS supports multiple forms of Member authentication, while continuing to respect privacy.
PrP 24 Multiple Persona: PICOS allows members to have multiple persona.



Appendix B Relation Between Trust & Privacy Principles and PICOS Principles

Here follows the list of trust and privacy principles together with the corresponding WP4 PICOS Principles (PPs) [D4.1 Ch.7], from which those principles were directly extracted or derived.

T&P Principle	Name	PP	PP name
TrP 1	Openness and Transparency	PP5	Openness and Transparency
TrP 2	Trust between communities	PP6	Trust Between Communities
TrP 3	Provenance	PP12	Provenance
TrP 4	External services	PP13	External Services
TrP 5	Audit	PP14	Audit
TrP 6	Objective/subjective trust	PP16	Objective/subjective trust
TrP 7	Consensus	PP17	Diversity
TrP 8	Member accountability	PP23	Trust
PrP 1	Notice of Collection	PP1	Compliance with Legislation
PrP 2	Policy Notification	PP1	Compliance with Legislation
PrP 3	Changes in Policy or Data Use	PP1	Compliance with Legislation
PrP 4	Timing of Notification	PP1	Compliance with Legislation
PrP 5	Sensitive Information	PP1	Compliance with Legislation
PrP 6	Informed Consent	PP1	Compliance with Legislation
PrP 7	Change of Use Consent	PP1	Compliance with Legislation
PrP 8	Consent Denial	PP1	Compliance with Legislation
PrP 9	Limitation of Collection	PP1	Compliance with Legislation
PrP 10	Fair and Lawful Means	PP1	Compliance with Legislation
PrP 11	Acceptable Uses	PP1	Compliance with Legislation
PrP 12	Data Retention	PP1	Compliance with Legislation
PrP 13	Third-Party Disclosure	PP1	Compliance with Legislation
PrP 14	Third-Party Policy Requirements	PP1	Compliance with Legislation
PrP 15	Access to Information	PP1	Compliance with Legislation
PrP 16	Provision of data	PP1	Compliance with Legislation
PrP 17	Correcting Information	PP1	Compliance with Legislation
PrP 18	Safeguards	PP1	Compliance with Legislation
PrP 19	Data Accuracy	PP1	Compliance with Legislation
PrP 20	Public Policies	PP1	Compliance with Legislation
PrP 21	Data Management	PP2	Data Ownership
PrP 22	End-to-end privacy	PP9	End-to-end privacy
PrP 23	Authentication	PP17	Authentication
PrP 24	Multiple Persona	PP18	Multiple Persona