Grant Agreement no. 215056

Title:	***D4.1 Architecture***
Editor:	*Stephen Crane, Hewlett-Packard Laboratories Bristol, UK (HPL)*
Reviewers:	*Isaac Agudo, Universidad de Málaga, Spain (UMA)*
	Eleni Kosta, Katholieke Universiteit Leuven - Interdisciplinary Centre for Law and ICT, Belgium (K.U. Leuven)
Identifier:	*D.4.1*
Type:	*Deliverable*
Version:	*1.4*
Date:	*09.03.2009*
Status:	*Final*
Class:	*Public*

Summary of deliverable

This Deliverable D4.1 presents the first version of the PICOS architecture. It describes almost fifty components that will go to make up a PICOS community, and set the context for two Work Packages that will directly benefit from this work, namely WP5 and WP6.

While the architecture is important, the process that has been used to define the architecture is equally significant. Starting with real-world requirements, derived form earlier PICOS deliverables, the Architecture Team produced detailed lists of principles, features, system requirements, trust models and interdependencies, resulting in a comprehensive and justifiable design.

Consideration is also given to the social and legal aspects of using and operating a PICOS community.

In recognising that it is rarely possible to create a community from scratch, the architecture has been designed to be compatible with existing community implementation, and to allow a gentle adoption path for those new to privacy respecting communities.

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

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
T-Mobile International AG	Germany
Leibniz Institute of Marine Sciences	Germany
Masaryk University	Czech Republic

The PICOS Deliverable Series

- D2.1 Taxonomy July 2008
- D2.2 Categorisation of Communities July 2008
- D2.3 Contextual Framework November 2008
- D2.4 Requirements November 2008

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

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 on-line collaboration via the Internet, many European citizens spend work and leisure time in on-line communities. Users consciously eave 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 identity management 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.

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 community applications;
- *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.

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

Charter

Objectives

The objectives of this WP are to create a technical architecture and design for the PICOS identity management platform. This includes the data model that contains the identity information, the toolbox of components that provide the identity management functions, the data flows between them and the protocols for them. The essential goals and attributes of the architecture and design are as described in the PICOS project objectives, namely to cater for the identity information flow needs of new, context-rich mobile communication services for communities, whilst meeting their participants' requirements for trust and privacy in an acceptable, trustworthy, open and scalable manner.

In that much of the fundamental technical innovation of PICOS will be contained within, and expressed by, the architecture, the two deliverables of this WP, which will provide a statement, on an approximately annual basis, of the research progress of the project; this can be also be used as input to the EC IST research agenda.

Description of work - Task 4.1 Platform Architecture and Design 1

The objective of this task is to define the first version of the architecture of the PICOS platform. It will form the basis of the first version of the platform prototype, which will be built and used to validate the concepts and their acceptability from a user experience viewpoint.

The work will start with a review of the Requirements document (D2.4) that is created by WP2, in order to understand the full set of requirements of the identity management system that is required to support the target sets of federated, context-rich community services that the project will focus on.

The necessary major data components, functional components and main data flows will be derived from this, together with constraints imposed by the requirements for trust, privacy and scalability. From this, an initial outline architecture will be created.

The elements of this will then be elaborated by a process of iterative decomposition until the functions of all non-trivial process modules, the contents and locations of the data structures, the data flows and the means of protecting/securing them are defined. These definitions will be reviewed by WP3 as they evolve. As necessary, the elements of this initial architecture will be re-defined and the relationships between them modified in order to meet the objectives of this WP.

When the overall architecture and the definitions of all the elements are stable, each element will be fully specified as to function, input, output, format, sequence etc., as appropriate. These specifications will also be reviewed by WP3 as they evolve; as necessary, deficient elements and/or the architecture as a whole will be changed to resolve issues that threaten the achievement of the objectives.

When the element specifications are stable, they will be finalised as deliverable D4.1 (Platform Architecture and Design 1), which will be contributed to WP5 for implementation and to WP3 for a final assurance evaluation.

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

Foreword

Deliverable D4.1 is a collective work by the WP4 Architecture team, whose members are listed below. A substantial part of the work involved identifying and describing a wide range of components that make up the architecture, and for this credit is recorded against each component description.

We are very grateful to the member of PICOS who prepared earlier deliverable, which lay the foundation for the architecture.

Special mention goes to Eleni Kosta (K.U. Leuven) for advising on legal aspects of the architecture, and to Bernd Ueberschaer (IfM-Geomar) for contributing the initial angling user scenario.

With thanks to the PICOS WP4 Architecture Team.

The Architecture Team

ATOS, BRNO, HPF, ITO, JWG, TMO, UMA and HPL

and

K.U.Leuven, IfM-Geomar

Reviewers:

Eleni Kosta, Katholieke Universiteit Leuven - Interdisciplinary Centre for Law and ICT, Belgium (K.U. Leuven)

Isaac Agudo, Universidad de Málaga, Spain (UMA)

Editor:

Stephen Crane, Hewlett-Packard Laboratories Bristol, UK (HPL)

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

Table of Contents

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

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

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

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

Table of Figures

List of acronyms

Abbr	*Abbreviation*
AES	*Advanced Encryption Standard*
API	*Application Programming Interface*
CA	*Certification Authority*
CS	*Client Server*
CSCF	*Call Session Control Function*
DRM	*Digital Rights Management*
DSA	*Digital Signature Algorithm*
Dx.y	*[PICOS] Deliverable: Work Package x, Deliverable y*
EPAL	*Enterprise Privacy Authorisation Language*
FTMGS	*Fair Traceable Multi-Group Signature*
GPS	*Global Positioning System*
GSM	*Global System for Mobile communications (originally Groupe Spécial Mobile)*
HTTP	*Hypertext Transfer Protocol*
ICT	*Information and Communication Technology*
ID	*Identity (also Identifier)*
IdM	*Identity Management (also Identity Manager)*
IM	*Instant Messaging*
IPSec	*Internet Protocol Security*
ISO	*International Organization for Standardization*
JSR	*Java Specification Request*
MAC	*Media Access Control*
OSI	*Open Systems Interconnection*
P2P	*Peer-to-peer*
P3P	*Privacy for Platform Preferences*

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

PA	*Presence Agent*
P-CSCF	*Proxy - Call Session Control Function*
pdf	*[Trademark] Portable Document Format*
PF	*PICOS Feature*
PICOS	*Privacy and Identity for COmmunity Services*
PP	*PICOS Principle*
PUA	*Presence User Agent*
PUC	*PICOS Use Case*
RMI	*Remote Method Invocation*
RPC	*Remote Procedure Call*
RSA	*Rivest, Shamir and Adleman*
S/MIME	*Secure / Multipurpose Internet Mail Extensions*
SDK	*Software Development Kit*
S-HTTP	*Secure Hypertext Transfer Protocol*
SIP	*Session Initiation Protocol*
SLA	*Service Level Agreement*
SSO	*Single Sign-On*
TLS	*Transport Layer Security*
TOR	*The Onion Router*
TSA	*Time Stand Authority*
TTP	*Trusted Third Party*
URI	*Uniform Resource Identifier*
Wi-Fi	*[Trademark] Wireless local area network*
WP	*Work Package*
ZKP	*Zero Knowledge Proof*

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

1 Introduction

WP4 is responsible for defining the PICOS architecture that will be developed further through to implementation in successive work packages, principally WP5 and WP6.

Deliverable D4.1 is the first deliverable produced by WP4. Its role is to draw together the work of previous deliverables, for example requirements gathering, and derive a technical description of the components that will make up a PICOS community. In so doing, D4.1 answers several important question that define the problem that the PICOS project aims to solve, and scopes the solution.

Earlier deliverables reports on the requirements of our reference communities, and strongly influences the architecture. By taking the original requirements, identifying features and deriving components, the PICOS architecture reflects the needs of the target community(ies) that we seek to address. The D4.1 development process is outlined below.

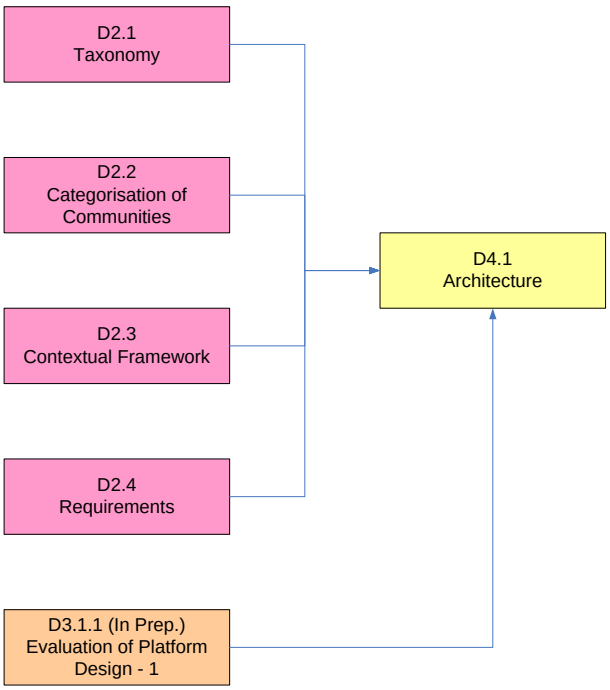


Figure 1 – D4.1 development process

D4.1 starts in **Section 3** by looking at a typical scenario that a PICOS community may serve. This is an angling scenario, and based on the experience that the project has gained from working with the angling reference community and FishBase. It tells a 'day-in-the-life' story of an angler, and touches on many of the privacy, trust and identity management issues that we believe PICOS should address.

PICOS is interested in all communities, but especially mobile communities. In **section 4** we sketch out typical topologies in order to understand the physical relationship between the various entities that make up a community. Our aim is to be as topology agnostic as is practicable.

In **Section 5** we focus on another important aspect of a community, namely Trust. Every community will have a different trust model. Some will be very trusting, while other will be distrusting. It is important to align the architecture with a trust model that best matches the type of community that PICOS is planning to address. Ideally, PICOS will support multiple trust models, and it is our aim in designing the architecture to include a wide range of models, but we recognise that in the short-term we need to be pragmatic if the following work packages are to be successful in the limited time that we have available in the project.

For the outset we understood that legislation would play an important role in defining the architecture. Compliance with privacy and law enforcement laws is mandatory, but this requirement also creates tensions in terms of trust. The PICOS architecture needs to balance these often opposing needs. **Section 6** explains the legislative requirements placed on PICOS.

In **Section 7** we start to define the architecture. We begin with a set of PICOS Principles, derived from past work in PICOS and existing published research, which establish the main features of the architecture.

In parallel with the PICOS Principles, in **Section 8** we examine the main features that PICOS will deliver, starting with user expectations which we subsequently use to derive system features.

With the Principles and key features defined, we start to create the architecture. First we define and describe low-level components (**Section 9**), ready to form the architecture which we present in a later section (**Section 12**).

Two special features of the architecture are described in **Section 10** and **Section 11**, the PICOS Toolbox and the PICOS Client respectively.

In **Section 12** we present for the first time the overall architecture, showing how components are combined.

We test our understanding of the architecture in **Section 13** by creating a set of carefully selected use cases, which we believe describe how several of the key features of the community will be handled by the architecture. The set of use cases that we chose to examine is only a sub-set of all the possible uses that the architecture may encounter, but we believe they represent the core essential use cases.

D4.1 describes the PICOS architecture at a high level, and does not include any implementation details. However, in defining the architecture it was inevitable that some implementation considerations would arise and consequently influence the design. Rather than ignore this fact, in **Section 14** we describe a practical implementation of the architecture. The description is high-level, but it gives clues as to how communications, community, trust, privacy and identity management services can by achieved.

In **Section 15** we explain the link to the follow-on work packages that will develop the architecture further, namely WP5 and WP6. These two work packages ultimately produce a prototype that can be evaluated in user trials. One role of D4.1 is to provide direction and justification for the decisions that WP5 and WP6 will need to take.

Finally, in **Section 16** we describe research opportunities that have arisen during the production of this first pass of the architecture. We hope that some of these will be pursued in the remainder of the PICOS project.

The D4.1 road map is shown in the following diagram:

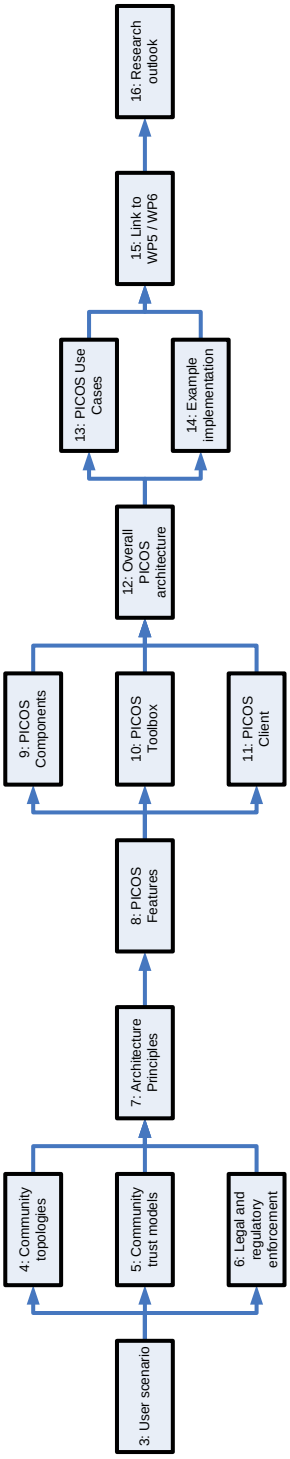


Figure 2 – D4.1 Roadmap

2 Relationship between D4.1 and 4.2

D4.1 is the first of the two architecture deliverables that WP4 is tasked to deliver. D4.1 bridges the gap between the requirements gathering exercises and the work of WP5 and WP6 (which turns the PICOS vision into reality).

D4.1 is necessarily focused on the immediate requirements of the target community. Nevertheless, it has some scope to consider the broader issues of privacy in online communities, and to reflect these requirements in the architecture, in several instances indicating the opportunity for future research. But the restricted scope does constrain what is possible in the time available.

D4.2 builds on D4.1, following on approximately one year after D4.1 is delivered. It will take D4.1 and the resulting work of WP5 and WP6 as a starting point, and adapt and/or extend the architecture to address any concerns that the construction and subsequent usability testing reveal.

D4.2 will also look at different trust models – principally those trust models that place less dependency on a single entity like a community operator, and thus return greater control to the individual. This 'low trust' trust models respond to the concerns that many users of social network have, which in general lack the inherent trust that we see in the reference Angling, Taxi Driver and Gamer communities that are attracting PICOS's immediate attention.

The split in workload between D4.1 and D4.2 is roughly 60:40, and the effort devoted in D4.1 to the needs of WP5 and WP6 is higher than is expected in D4.2. Thus D4.2 will be able to look to a broaden horizon. This can be visualised as follows:

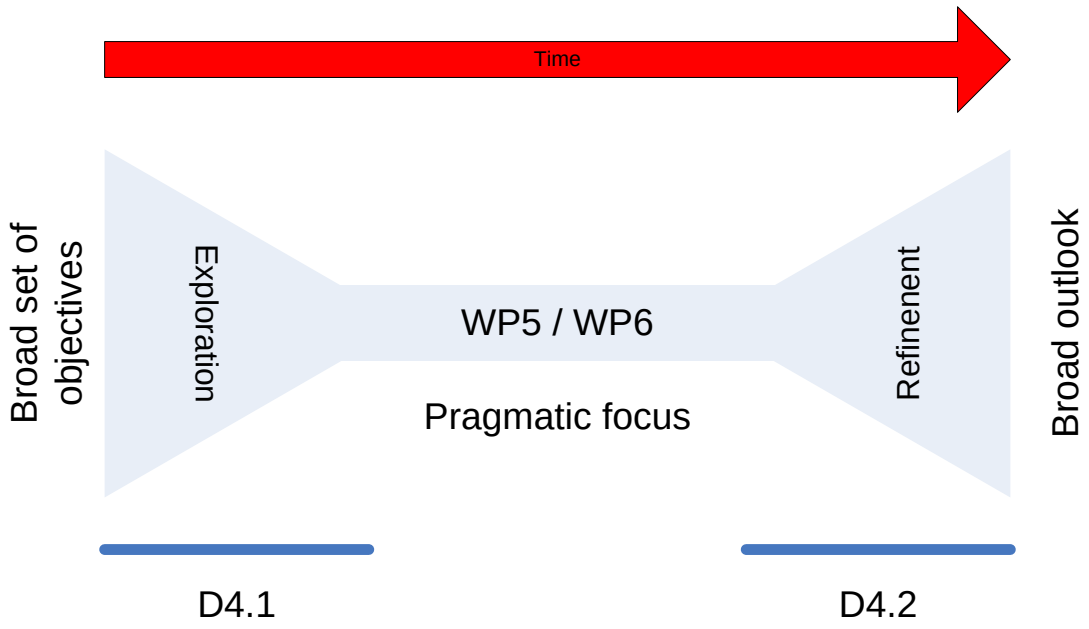


Figure 3 – Relationship between D4.1 and D4.2

3 User scenario

3.1 Relationship between scenario, the architecture and the prototype

Note: In this section we describe an artificial scenario – a user experience – which tells a typical day-in-the-life story of someone who we anticipate would benefit from using a PICOS enabled community. Studying the angling community has helped us gain a good understanding of what PICOS should deliver, and allowed us to scope the project. At this stage in the project no decisions have been taken on the first prototype, so it may not be based on this scenario at all, and it would be misleading to suggest that the features that John the angler uses will be prototyped. However, as far as possible this first version on the architecture tries to capture and address the tensions that this scenario predicts.

Throughout the user scenario we have attempted to show how the features that John desires can be realised by the PICOS community. We have done this by inserting references to PICOS Use Cases (PUCs) that we develop later in Section 13. A reference {PUCn} refers to Use Case n.

3.2 John, the Angler

An angling holiday:

John is an experienced angler, especially with the fish that live in the North Sea and adjacent waters, such as the North Atlantic and the Baltic Sea. On the occasion of his 40th Birthday, his fishing buddies decided that John should try fly-fishing, so they gave him a basic fly-fishing set as a Birthday gift. However, John has never attempted fly-fishing before, but believes that this fishing method is really worthwhile trying. In order to get a comprehensive idea of what fly fishing could be like, he decides to spend his next vacations in the mountains of Bavaria, where he can expect a number of promising sites for fly fishing. In preparation of his vacation, he found a fly fishing simulator on the Internet, which he considered useful for getting an idea for what fly fishing is like. Playing with the simulator gave him an idea; Fly-fishing could be a lot of fun, but it is also a fishing technique which needs a lot of skill and knowledge about the ecology of the target fish and their environment.

John's concerns about technology:

John is not a technology expert, but he is reasonably comfortable using the Internet; he sends emails and checks his bank balance online regularly. He is aware that the Internet presents a risk, but beyond knowing not to respond to Spam emails with his bank account details, and to watch out for viruses, he isn't sure what he needs to worry about or how to protect himself. He knows that people steal personal information and that hardly a day seems to goes by without there being a report in his newspaper saying that another government department or company has lost personal data.

Besides the membership in a regular fishing club, 'ASV Nordseekant' which is located in the city where John lives, he is a member of an online marine angler's community. He noted that a group of anglers in this community indicated a while ago that they are experience in fly fishing. So John logs in to his angling community and is asked to register with the group. On this occasion John is accessing

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

the website from his home PC, but since the community also supports access from mobile devices he expects to use his mobile phone to gain access once he leaves on his trip.

John registers with a community

John provides the requested personal information, including an angling credential that states that he is entitled to fish at his chosen location {PUC1}. John's angling community provides a service which allows members to sign in and apply online for a rod license for marine fishing, which is issued by the governmental fishing authority. This is a credential which is endorsed, and which provides evidence of John's right to fish in the said waters {PUC1}. The information provided is authenticated, and John is subsequently granted access to the thematic groups in his community about fly-fishing and he was searching for information he needs to plan his fly-fishing trip {PUC2}.

John joins a group

When John registered, he created a profile that defined what information about John other members can see {PUC1}. He can also create a group – a buddy list – in which to list the other anglers he will interact with on a regular basis {PUC9}. John decided to join the existing discussion group on fly fishing. However, since he was afraid, that he can be blamed because of its little knowledge, he wanted to discuss fly fishing issues also only with known buddies and he decided to create his own group on this topic {PUC9}. Initially, he invited just his known friends to become members of this private room, but he was already considering opening this group to all other community members when he would be a bit more experienced. This is all handled by the social relations facility, which is responsible for managing and graphing John's connections to other community members {PUC7}.

John sets his privacy preferences:

The profile also permits John to set privacy preference settings which, among other things, allow John to automatically disclose his social presence management component which shows John's online presence, an indication of his online status and location. His buddies or other anglers can check if John is online and available to chat, just as John can check the status of other users {PUC8}.

John searches for recommendations:

Before John starts his fly-fishing vacations he would like to get recommendations for promising angling spots and other necessary infrastructure such as restaurants, tackle shops and licensing rules and regulations around the anglers hotel he has booked for his vacations. John considered becoming a temporary member of one of the local Bavarian online angling communities which was recommended from his friends in his marine angler's community {PUC4}.

John logs in:

With his angling vacations approaching, John frequently logged in into the Bavarian online angling community and tried to find the respective information posted from other members of the community. As a registered member, John can upload photographs and download angling information, and communicate with other users {PUC7}. When logging in, John will need to prove his identity, by using his chosen authentication mechanism from the set of mechanisms that the angling community supports {PUC2}. John did not need to register again for temporary membership in the Bavarian online community on fly-fishing, since with his membership in the marine anglers community, he is automatically and transparently granted access to all other angling-related communities and web sites that he wants to visit when preparing the fishing trip. The facility, known as federated access, also

allows John to use online services for which he has not registered. This is because through mutual agreement, John's registrations credentials are accepted by other service providers.

This is especially useful, since some of this information that John needs is provided by third parties, for example weather information or qualified biological information from FishBase about the local fish fauna including identification tools and field guides he can print or he can use from his mobile device {PUC6}. If John ever decides to fish in another area it means he does not have to register with a new community every time. John simply takes advantage of a federated access service that allows him to automatically gain access to the new community.

John checks reputation:

While searching for local information, John also wants to see user-generated recommendations for the results of the search. A recommendation is only useful if he can get additional information on the person who made the recommendation, e.g. their profile, their reputation in different communities and their relationship to John {PUC5}. This is only possible because the community offers an identity management component allowing the federation of partial identities, and thus cross-community reputation and recommendation.

John configures location and privacy settings:

John suspects that at this time of the day two fishing family members and another Swiss friend, Jean-Paul, may also be logged in to his main online community. He checks his buddy list for their status and location {PUC8}. Of course, this is only possible if John's family and friends have granted John right to see this information, which they will have configured using their own social relations facility.

John can see his family members, but his friend, Jean-Paul, is currently blocking access to this information because he probably has privacy concerns. So, John decides to communicate directly with Jean-Paul using the community's instant messaging service. John writes to ('texts') Jean-Paul, who fortunately is logged in. He is alerted by his vibrating Smartphone and reads the message from John asking for access to Jean-Paul's status (presence) information. Attached to John's message, Jean-Paul receives a digitally signed statement issued by the Reputation Management component, which convinces Jean-Paul that he can trust John {PUC5}. In response, Jean-Paul also grants John access his social presence information, simply by updating his privacy preferences {PUC8}.

As John and Jean-Paul are holidaying together in the Alps later in the year and want to go fishing together, Jean-Paul also grants John access to his location information, but only during the days he knows they will be in the same holiday area. This is again managed by the privacy preferences and social presence facility.

John immediately sees the new information and has a great idea. He sets up a group using the Group Management facility so that he and Jean-Paul can share specific information to help with planning the trip; e.g. he would be able to deliver his exciting experience to this group straight from the watercourse, using his Smartphone {PUC9}. Although personal information will be shared, both friends are confident because they know that no one outside the Group will have access {PUC7}. They also realise that during and after the planned trip they can share photos and other (multi-media) information showing the great catches that they expect to make.

John accesses another community:

Finally having arrived in the angler's hotel, John decides to join another angling community, which is suggested from a member of the Bavarian online angling community as a group of fly fishing

specialists who knows the fishing sites where John wants to go fishing in his vacations very well. Normally this community is restricted to register members because they want to hide their special knowledge within a limited number of members. Only users of good repute can access the community. Because John has not been member before, his reputation is unknown {PUC5}. Fortunately, John has been a good member of another community and he can transfer that reputation to this new community as proof that he can be trusted.

Authentication:

The new angling community needs to verify John's identity and that John is indeed the member of the other angling communities. This is possible because of a federated identity management system which provides community membership management and trust management across communities. (Of course, if the local community does not know the other communities then the local community must decide dynamically how much to trust John). Once John's identity has been validated and he has been granted temporary access to the members-only section of the community site, other members of that community can see John's profile, and can see the reputation that he has established in the other communities before and has chosen to disclose.

Upon arrival at the water course, John realised that he has forgotten a number of items necessary for a successful fishing day, such as flies which are due at this time of the year. Thus he searches for a local tackle shop on his Smartphone, and he sees recommendations that have been made by members of the local angling community {PUC5}. Those will be presented as recommendations coming from fellow members of the community that he now belongs to. John can also see if any of those members are currently online, and if so he has the possibility of communicating with them directly, i.e. he can ask for advice in real-time. Any community member can control who can see their status. Members can also control who can contact them directly. For example, some members may prefer to only accept messages from registered members, as opposed to guests.

John posts feedback:

Having visited the tackle shop and gathered the needed equipment based on excellent advisements of the tackle shop owner, John decides to share (post) his own recommendation about the shop on the shop's website. Rather than posting as an anonymous user, he decides to post it using his local angling community identity. The shop website verifies that John is indeed that member of the local angling community.

This recommendation enables John to purchase items using his mobile device from the online shop. This is very convenient for John, since the shop delivers the chosen items to John's hotel. However, the shop needs John to first supply some sensitive personal data.

John wants to be anonymous:

John is excited about finally going fishing, but in the back of his mind there are those concerns about security and privacy. He wonders why he should trust the community to look after his information. Has he made a terrible mistake that he will live to regret? But then he remembers he also joined the local community because he wanted to get to know more local anglers who can help him to make the fishing trip successful, so perhaps he needs to be more relaxed about all this privacy stuff. After all, it's probably all hype to get people to buy credit card insurance! He decided then to share photographs of the fish he catches, the location, the date and time caught, and his experience with successful baits. However, he thinks that he doesn't mind telling his new angling friends from the local online

community, who he already knows well, but he doesn't want the whole world to know about his special experience.

John makes a payment:

Still a little concerned, John decides to investigate further, and discovers something called anonymisation and pseudonymisation, which apparently means that John can interact with others without telling them his real name. Sounds like the ideal solution. When John makes a payment or provides evidence of entitlement, non-essential personal information is obscured.

John also discovers that he can restrict who can see his information through something called access control {PUC7}. This is really easy to do since he only needs to set a few options in his personal profile and that's it.

John terminates his membership of the community:

At the end of the fly fishing vacations, John can choose to cancel his membership in the special local online community, or he can wait for it to expire automatically {PUC3}. However, even though he has left the community, the history of his membership, messages that he posted, and any reputation that he established, is maintained by the local angling community. Before leaving the community John decided to post photographs of his trip including the fish he caught. As an acknowledgement of their useful tips and trips which promoted John's success as a fly fisher, other members can still see the photos of the catch, even though John is no longer a member of the community. Since he behaved according to the rules of that community and since he provided content, he was rated from community members as a trusted fishing buddy and his reputation score (which never expires and can be transferred to other communities) was increasing which may facilitate to access special groups in other online angling communities in the future {PUC5}.

4 Community topologies

The PICOS architecture is intended to be topology agnostic. Put another way, it is designed to be easily implemented on a range of interconnect or communication configurations.

The prototype that follows D4.1 will be a specific configuration, and though the aim is for the architecture to be flexible, the reality is that at this stage of the project the priority is to focus on the anticipated needs of the prototype. This means that the architecture will lean towards a client-server implementation, and that this will necessarily be evident in the use cases and example implementations that we later describe.

Nevertheless, it is still useful to reflect on the likely topologies that PICOS should support. Beyond D4.1 there will be opportunity to explore these options in more detail, and to examine the privacy and trust issues that different topologies give rise to. In certain situations, different topologies will address concerns that the initial choice of topology gives rise to.

In our ideal model, PICOS functionality is delivered as a service. Services can be hosted locally or centrally, and can be for the direct benefit of the member or of the community as a whole. As we develop our understanding we refer to these groupings as My Services, Our Services and Community Services respectively, whether hosted locally or centrally.

4.1 The single entity model

Our model is based around the concept of an entity that in principle can supply all PICOS services. In practice this is unlikely to be the necessary, but the principle helps us envisage more complex configurations. A single entity, which could be client or server, is depicted thus[1]:

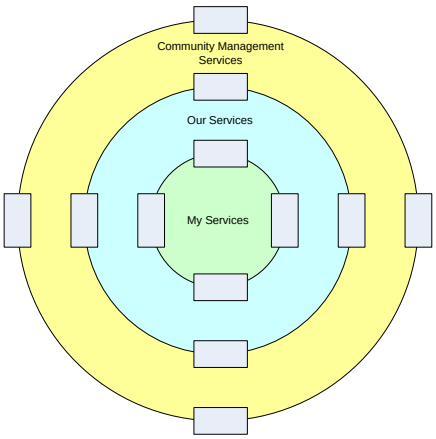


Figure 4 – Single entity model

[1] The small gray boxes simply show where the different 'layers' of services interact (or communicate). There's no reason for there being four gray boxes, except that it makes it easier to illustrate interconnected entities.

4.2 Client-server model

One option to interconnect clients is to use the client-sever topology. In this topology clients (e.g. smart phones) are represented by the inner green circles which host 'local' services. The client can process local service but relies on the community for shared services and services that are too demanding (in terms of computing and storage resources) for the client to host.

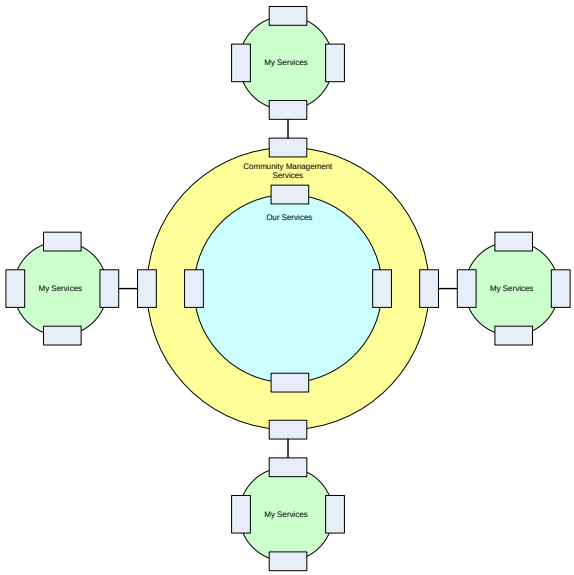


Figure 5 – Client-server model

To give an idea of how such an idealised model might be implemented, we include the following figure which shows a typical mobile community.

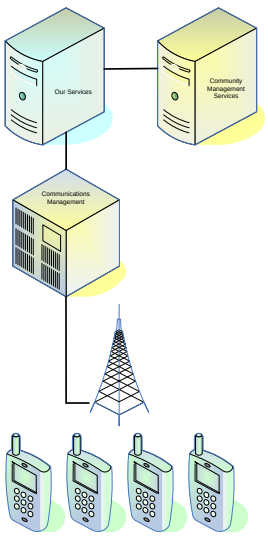


Figure 6 – Client-server implementation

4.3 Client-server architecture – conjoined communities

We envisage that communities will wish to interact with one another. In the following diagram we show two PICOS communities connected together. Member Services (Our Services) might be shared between the two communities.

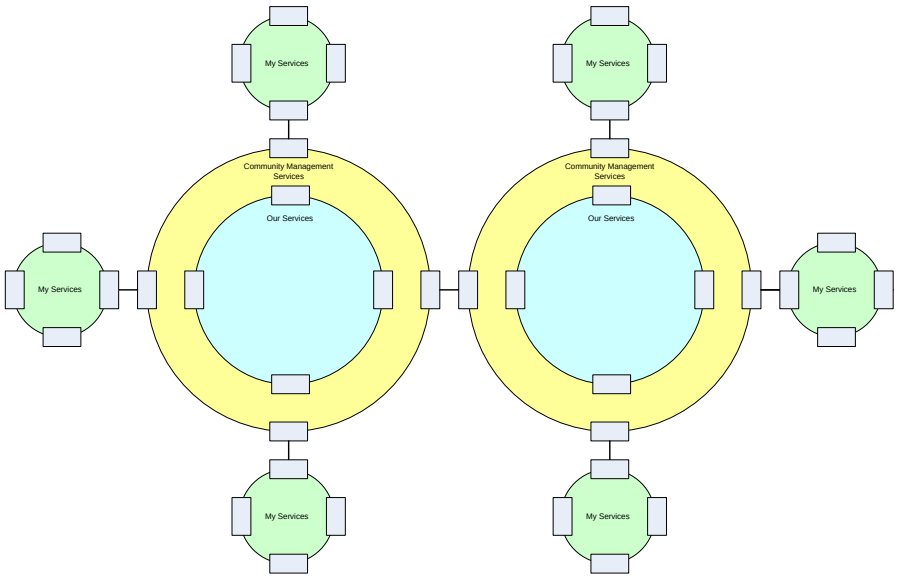


Figure 7 – Conjoined communities

It is also likely that one community will provide services for the other, for example where one offers a specialised feature. This corresponds with the general view that some services will be provided by independent third parties which do not have any members themselves. Such a situation might look like this:

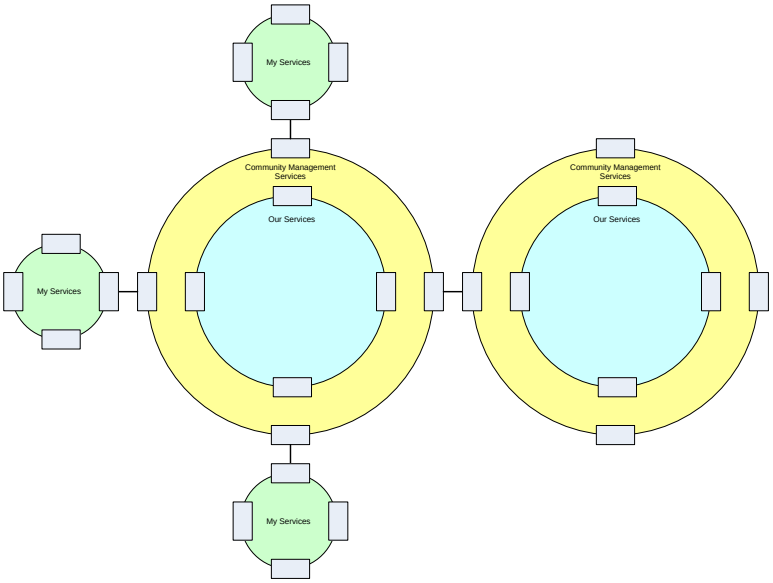


Figure 8 – External services

4.4 Peer-2-Peer architecture

The second topology that we consider is peer-to-peer (P2). In this model all services are distributed amongst members, and there is no need for a central service provider. Services (Our Services) are shared between members, with perhaps one member having a more powerful mobile appliance that can run the more demanding service.

This model is particularly attractive for members who feel uneasy trusting a community operator. However, it should be noted that a P2P service can be provided 'through' a centralised community provider. This can be a more suitable option if the community is build on top of conventional communication technologies, e.g. a mobile phone network.

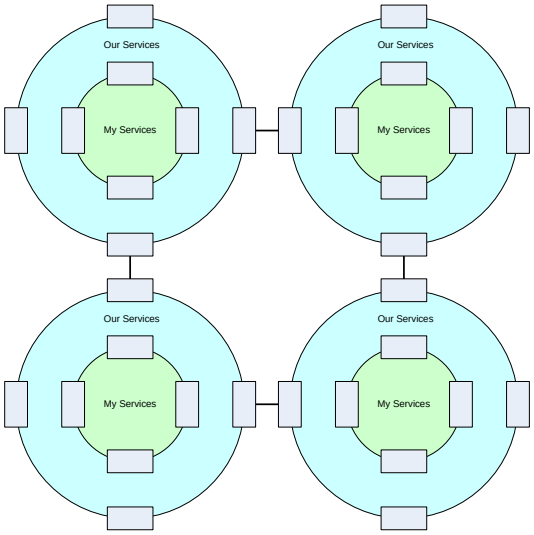


Figure 9 – P2P architecture

4.5 Dumb terminal architecture

A third topology is the thin client model. Here a 'dumb terminal' (web services portal) is used to access the community. This could be a smart phone that runs a simple client application that provides access (only) to the community, e.g. a browser. Apart for the access application, no other services are hosted on the client appliance; everything else, including My Services, is hosted centrally.

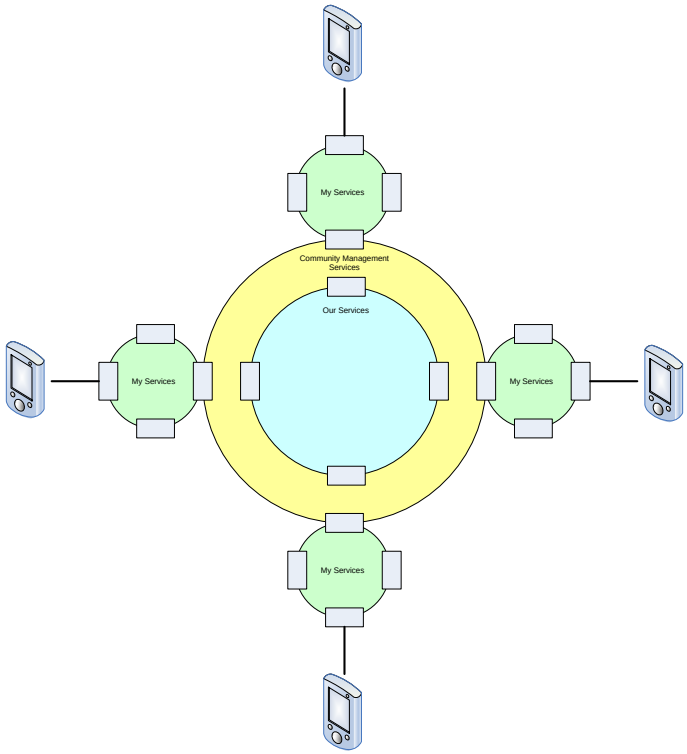


Figure 10 – Dumb terminal architecture

Such a configuration might be implemented as follows:

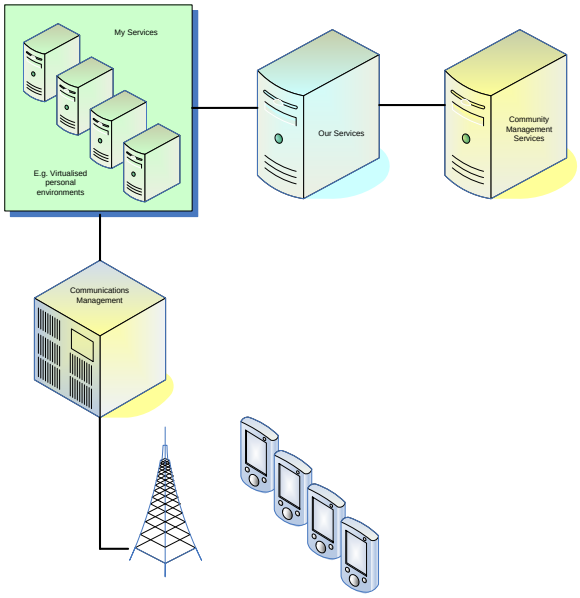


Figure 11 – Dumb terminal implementation

4.6 Conclusion to discussion on community topologies

In this section we present an idealised model of the PICOS architecture. Our aim is to look beyond the popular implementation of communities which tend to mimic communications topologies. There are two reasons for taking this approach:

- In the near future we anticipate personal networks becoming more common. The reference communities that we have examined could benefit from such technology, for at least part of their community services. For example, anglers might form a local ad hoc community while fishing.
- Different topologies create and address different trust models. For example, P2P (no community operator) may appeal to members who trust teach other, but not a central authority. We saw this with our Taxi Driver community[2], where drivers had a high degree of trust in each other (formed in the real world) and saw no need to trust anyone else.

Different topologies present interesting research challenges, some of which PICOS would like to investigate further. However, the reality is that in order to test the technologies that will enable a PICOS community, we need to work with today's technologies, and that means client-server topologies. As WP5 and WP6 develop the architecture further we will see more focus in this direction.

[2] The Bristol (UK)-based taxi drivers led by Sally Hoare (Hambrook Cars) who acted as a PICOS reference community during the requirements gathering phase of the project.

5 Community trust models

5.1 User attitudes towards trust and risk

Every community, and every community member, has a different attitude to risk. Some are risk accepting, while others are risk averse.

For those members who do worry about the risks of using a community, a range of situations are possible. Some members will want to take control of the situation and minimise the risk through their own personal actions. Other members will look for assurances from the community that the actual risks are not serious enough to worry about, or that there is a path to restitution should something go wrong.

In fact, there is a spectrum of possibilities, from high trust (low personal control) to low trust (high personal control), as the following diagram shows.

Figure 12 – Trust spectrum

Somewhere between these two extremes lies the community that PICOS will target for its first prototype.

We use the term 'trust model' to describe the trust that members have in each other and/or in the community. At one extreme we have complete distrust in everything, and this is when it may become necessary to move all the sensitive processing to the 'trusted' client. At the other extreme everyone trusts the community operator (but not necessarily other members).

5.2 The member perspective

Some of the different attitudes that members might express are:

- Complete distrust: No one trusts anyone else. Member authentication takes place at the client; Everything is signed at the client; Reliance on third party to endorse identities and signatures. Third party handles disputes and law enforcement.
- Some trust in Operator: Members trust Community Operator to manage community (reputation, privilege enforcement, profiles, etc), _BUT NOT_ manage member content. Member authentication takes place at the community, but signing takes place at the client and is endorsed by a third party. Third party handles disputes and law enforcement.
- Trust Operator but not each other: Members trust Community Operator to manage community _AND_ manage member content: Member authentication takes place at the

community. Content signing takes place at the community and is endorsed by the Community Operator. Community Operator handles disputes and law enforcement.

- Trust all but Operator: Members trust each other, perhaps because they know each other before the on-line community was formed, but don't trust the Community Operator who they see a simply a service provider. Member authentication takes place outside the community.
- Members rely on legislation and the obligation of a Community Operator to provide proof of compliance, to address personal concerns. Proof is underpinned by technology employed by the Community Operator.

It is important to understand the trust model. Managing risk can call on a variety of techniques, processes and attitudes. The approach for dealing with low trust will most likely be more demanding than for high trust. Members with low trust typically demand guarantees or isolation from the areas of high risk. Responses to this situation include:

- Today, in most communities, it is a case of 'fingers crossed and hope for the best', i.e. high trust. (High trust means that members are forced to trust the community; low means they are not.)
- Members might decide that they can never trust the community, and instead decide to protect their information in some other way, e.g. anonymise or encrypt it before it reaches the community. This is low trust option.
- Members might expect the community to be built in such a way that the weaknesses that concern them are eliminated. Again, this is low trust option.
- Members might make the community 'publicly' accountable for its actions. This is anything from high to low trust, depending on how much the members (can) rely on the accountability mechanisms.

Members worry about the content that they contribute to the community. By content we mean information that members contribute, or information that the community holds, that is personal and relates to a member.

5.3 The community operator perspective

So far we have discussed the concerns of the member. The community operator is likely to have similar concerns, where lack of trust in the community affect the ability to deliver a widely accepted service. For example, a community operator may worry that they cannot easily:

- Demonstrate that they are trustworthy
- Demonstrate, and be able to comply with the law
- Differentiate their community from existing communities

This is certainly the situation for a community operator that is essentially honest but wants to demonstrate their honesty to the (potentially highly distrusting) members of that community.

5.4 Target community for first prototype

The community that PICOS will target in the first prototype is exemplified by the angling community. This community is particularly interesting because it possesses several characteristics that social networks generally lack:

- It has a well defined purpose
- Members have a shared interest and shared values
- It has a co-ordinating entity that shares the same values
- It existed in the real world

Compared to a social network community, where trust is high and personal control low, the angling community looks for a balance of increased personal control and reduced need for trust. By contrast, a highly distrusting member would, compared with today's standard community offering, look for much greater control and reduced need to trust. The following diagram summarise this situation, and shows where we believe the first prototype of PICOS community will focus.

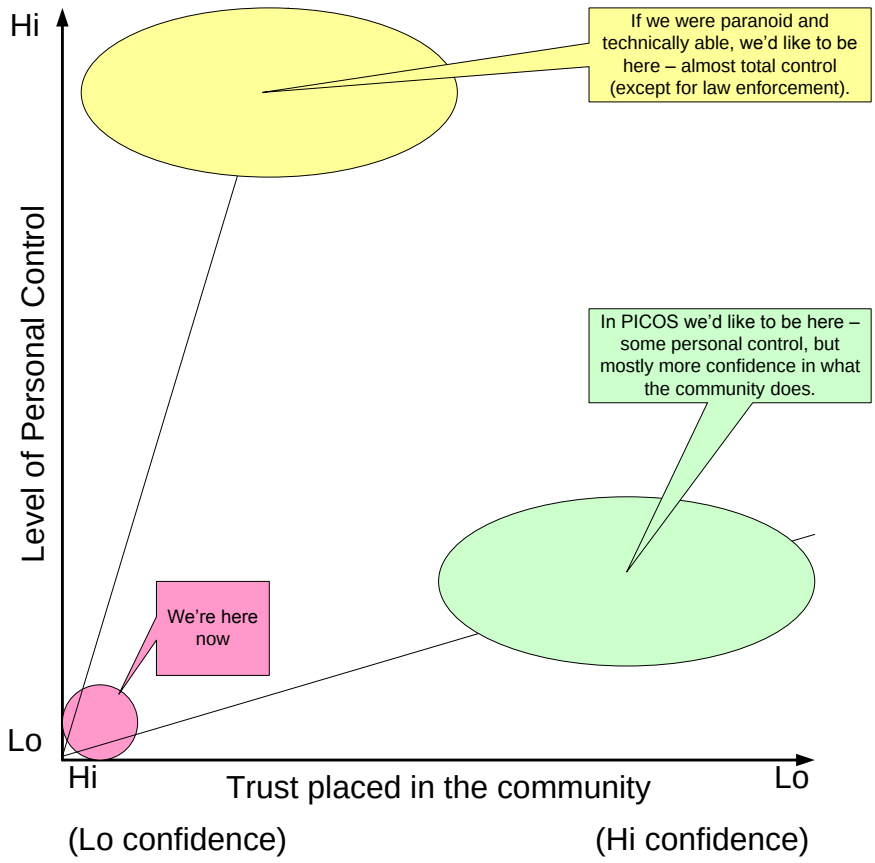


Figure 13 – Balancing trust and control

We can further define the target community by stating the desires of the membership. Members:

- Are interested in greater control over how their information is shared
- Want features that are not present in today's communities, e.g. Address books, groups, greater confidence in the identity of other members (OpenID, ID Brokers), evidence that their information has been accessed, assurance that the community complies with the law, a reputation system, feedback, profiles and privileges
- See the greatest threat coming from other members, not from the community operator, and from hackers outside of the community.
- Would trust a community that employs the latest (PICOS) technology to manage trust and privacy
- Believe that the community operator will be willing, or obliged (by law), to protect their data
- Do not require absolute guarantees, and consider 'after the event', or retrospective, control adequate
- Would not check the technology
- Use a community's reputation to decide how trustworthy it is
- Are more concerned about the authenticated identity of another member
- Look for strong authentication in certain situations, but third party endorsed identities was not a priority but would accept a mobile device that provides a trusted identity
- Are less concerned about integrity or provenance of content (they accept that content from an authenticated member is genuine)
- Want secure storage, but only for the more sensitive information
- Accept that law enforcement requires access to protected information, and would trust the community operator to perform the role of trusted intermediary
- In general, trust the community operator to perform the role of a trusted intermediary

A useful analogy can be drawn with a banking community:

- Customers have confidence that the bank process their information ‘safely’
- This confidence comes from the perceived professionalism of the bank, the technology they employ and the nature (purpose) of the interaction
- The main difference between banks and our community is that members share information between each other (or with groups)
- Another way to look at this is to consider the community operator as a data custodian

The fact that members communicate, share and trust one another does change the balance of trust in the banking analogy slightly, and certainly calls for a new set of controls and trust-enabling features.

5.5 An alternative trust model

The target angling community helps to define the needs of the prototype community with respect to privacy and trust. However, beyond the first prototype PICOS is interested in understanding other types of community.

A low-trust trust model, in which members do not (or at least should not) trust the community operator, is typified by today's social network communities. These communities do not have the long-established trust values that the angling community possess.

In the near future, more activities will be performed online using computers and the Internet. People will book a taxi using a computer; they will be registered at work using a computer; they will buy tickets for the theatre using a computer. When they enter the theatre at a given time they will be monitored (recorded) using a computer. When they talk with friends it will be by using a computer. When they interact with colleagues by means of online communities; when they book a hotel; when they enter a building; when they shop; when they pay their taxes – it will be through the use of computers.

Their whole life will be observed by computers. Therefore, the Internet, and the computers to which it connects, becomes the biggest surveillance system ever devised. Computers will have vast amounts of information about individuals. The technology makes this possible, and there is motivation – convenience, profit, competition – that makes it increasingly likely.

Evidence already exists of how easy it is for control over information to be lost. To date the widely publicised examples are accidental, but there is increasing concern that controls are not sufficient to prevent hacking. Examples include hackers who stole 17 million personal records from Deutsche Telekom, in Italy where Tax data for 40 Million people was accidentally published on Internet.

Incidents like these lead the concerned member to call for greater personal protection, and the approach promoted is one based on the *minimal disclosure* and *unlinkable transactions*.

Solving the trust problem that social networks present requires a different attitude. Essentially, the trust in the community operator is removed and distributed to one or more trust domains that are accepted by the member. In addition, sensitive process that might otherwise be carried out within the community is now performed in an isolated (probably local, e.g. smart phone) environment that the member trusts.

Fortunately, the changes mentioned are in principle all that is needed to change a solution for a high-trust community into a solution for a low-trust community. By creating trust domains and introducing online (or local) security (cryptographic) processing, the mechanisms deployed to protect privacy and engender trust become highly effective.

For example, a solution based on cryptographic primitives like Group Signatures or Traceable Signatures, provides:

- Anonymity
- Compliance with legislation and community operating practices
- Non-repudiation
- Personal accountability

- Reputation
- Strong identification and authentication

The downside of such solutions is the greater demand on infrastructure, which in practice takes time to develop and roll-out, and the increased inconvenience (in terms of additional complexity at the client device) and need for greater understanding by the member.

In designing the architecture for the target community, we have kept in mind that a stronger solution may be required for alternative communities / trust models, and have tried to keep the option to extend the solution in D4.2.

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

6 Legal and regulatory enforcement

PICOS aims at the creation of a fully legally compliant architecture and consequently platform. The current legal framework on privacy, data protection and identity management has already been analysed in PICOS Deliverable D2.3 "Contextual Framework" and has been complemented with specific legal requirements contained in PICOS Deliverable D2.4 "Requirements", which assist the developers in creating a fully legally compliant architecture. The legal requirements appear as principles in Appendix B of the D2.4 and serve as guidelines to the developers of the architecture.

The architecture reflects one of the fundamental positions of PICOS: that PICOS aims at the creation of a legally compliant platform. In all phases, from registration through to revocation, legislation is catered for.

While various components that form the overall architecture contain functionalities that enable legislation to be enforced, there is not one single part of the architecture that has sole responsibility. Instead, compliance with legislation is a design philosophy that permeates throughout the design process. The setting out of the legal requirements and their translation into clear principles for the developers at a very early stage of the PICOS project, as well as the continuous cooperation with the legal team during the designing phase of the architecture, follows the "privacy by design model". The privacy issues and in particular the processing of personal data (with the further implications regarding identity management) are taken into account at the earliest stage of the creation of the architecture.

The legal requirements, which are translated into principles in Appendix B of D2.4, could be expressed as policies, and the policies then interpreted and acted upon by each component. Equally, components could report back how effectively they have complied, and all the reports could be collated and presented as evidence. One of the six key areas, into which the PICOS architecture principles are divided, is Law. "Compliance with legislation" (see below PP1, section 7.1.2), "Data Controllers" (see below PP2, section 7.1.3) and the "Trusted Intermediary" (see below PP3, section 7.1.4) are three PICOS architecture principles, which are classified as having direct relation to law and more specifically to the data protection legislation. This classification shall however not be considered as exclusive. Several architecture principles that fall under one of the other key areas (i.e. trust, privacy, control, identity, other) have some relevance to law.

For instance the "Data Minimisation" principle (see below PP8, section 7.1.9), which is classified under "privacy" is also a core legal principle, according to which the processing of personal data should be limited to data that are adequate, relevant and not excessive.[3] According to this principle, data controllers are obliged to store only a minimum of data sufficient to run their services. While recognising that data minimisation is a principle adopted in European law, PICOS also appreciates that data is required in order to allow a community to grow. Technical tools and Privacy-Enhancing Technologies in particular, should be available to contribute to the effective implementation of the data minimisation requirement.

[3] Art.6(1)(c) Data Protection Directive.

Similar thoughts can be made on the "Audit" principle (see below PP14, section 7.1.15). Besides technical audits, this principle also refers to the legal/privacy audits that are needed in order to ensure compliance of the system with the data protection legislation and the relevant obligations that derive from it. Moreover two principles that ensure the exercise of control on data are the "Use of personal information" (see below PP3, section 7.1.4) and the "Protection of personal information" (see below PP4, section 7.1.5). The former relates to the control the data subjects (in most cases the users) have on their data and the latter aims at the protection of personal data, allowing the user to differentiate between non-personal data, personal data and sensitive data. The importance to differentiate between personal and sensitive data has already been highlighted in D2.3.[4]

With regard to automatic checking for compliance, automated checking of policies and regulations would be required. This represents a significant amount of research and development, on areas of new languages to universally express and process such legal restraints; security indicators to monitor infrastructures, real-time workflows to enact actions upon alerts are needed, etc. Legal compliance within PICOS developed tools and services are of paramount importance. The relevant rules are taken into account for the design of many of the core components of our architecture, especially all those dealing with identity management, privacy, reputation, content sharing, etc.

For instance the "Consent management" component (see below section 9.7.4) is closely related with the legal provisions of the Data Protection Directive on consent. It allows the members to modify or withdraw their consent, and it invokes the community-specific procedures that are applicable when consent is withdrawn, e.g. deletion of data, change to access rights to data, restrict access to community operator role only. The relation between the "data minimisation" component (see below section 9.8.4) and the data minimisation legal principle has already been implied above, during the discussion of the PICOS Architecture principles. The "Location sensor" component (see below section 9.5.14) is also closely related to legislation as it provides an interface to retrieve the current location of a member. The processing of location data is allowed according to the provisions of the ePrivacy Directive, as described in the PICOS D2.4 "Requirements" deliverable. Furthermore the "Audit" and "Accountability" (see sections 9.6.4 and 9.6.3 respectively) deal with the compliance of the user actions with, among others, his legal obligations.

PICOS needs to ensure that all personal data are kept in a form that permits identification of the data subjects for no longer than is necessary for the purposes for which the data were collected or for which they are further processed.[5] However, PICOS does not need to comply with the Data Retention Directive[6] and retain specific categories of data for law enforcement purposes. As it has already been discussed in PICOS D2.3 "Contextual framework" the Data Retention Directive applies only to providers of publicly available electronic communications services or public communications networks. Consequently the relevant obligations will cover only the telecommunications or mobile operator who enables some of the PICOS functionalities and not PICOS itself.

[4] Article 8 of the Data Protection Directive describes special categories of data, i.e., "personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, trade-union membership, and the processing of data concerning health or sex life", commonly known as sensitive data. The processing of the aforementioned data is prohibited, unless one of the specific grounds described in the same Article is fulfilled.

[5] Art.6(1)(e) Data Protection Directive.

[6] Directive 2006/24/EC of the European Parliament and of the Council on the retention of data generated or processed in connection with the provision of publicly available electronic communications services or of public communications networks and amending Directive 2002/58/EC, Official Journal L105, pp. 54–63 (15.03.2006).

As PICOS aims at the creation of a fully legally compliant architecture, it has embedded the legal requirements into the PICOS architecture principles and the PICOS components. The setting out of the legal requirements and their translation into clear principles for the developers at a very early stage of the PICOS project, as well as the continuous cooperation with the legal team during the designing phase of the architecture, follows the "privacy by design model". Although the PICOS architecture does not contain any fully legal components, it has been clearly illustrated that the whole concept of the PICOS architecture, its principles and components respect the data protection legislation and takes into account the needs of the law enforcement in the future.

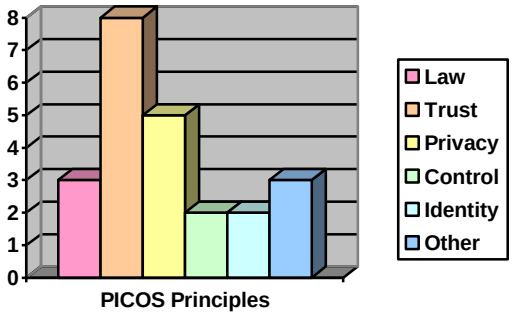
7 Architecture Principles

In this section we establish the PICOS Principles (PPs) that guide our architecture design. Each principle is derived from earlier work in which we gathered requirements from real-world and potential online community members. Selection of the principle is also influenced by our experience in the fields of communications, security and social values in trust and privacy.

Each Principle is assigned to one of six key areas, namely:

- Law
- Trust
- Privacy
- Control
- Identity
- Other

The distribution of the 23 principles is as follows[7]:



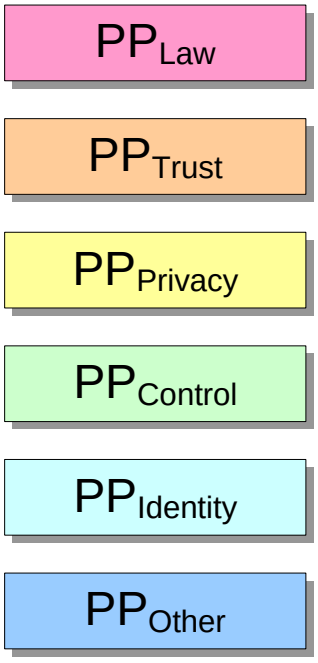


Figure 14 – Distribution of principles

[7] If viewed/printed in 'black and white, the columns read from left to right: Law, Trust, Privacy, Control, Identity and Other.

7.1 PICOS Principles

The 23 PICOS Principles are described in full in the following section. Each Principle is marked with the category that best describes the contribution that the principle makes to the architecture. It is accepted that some principles contribute to more than one category, but we only record the main principle at this time.

The definition of the principle is shown in italics. Supporting information follows each definition.

7.1.1 Overview of PICOS Principles by category

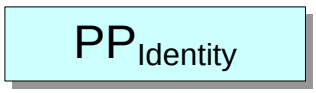
PP_{Law}	PP_{Trust}	$PP_{Privacy}$	$PP_{Control}$	$PP_{Identity}$	PP_{Other}
PP1: Compliance with Legislation	PP5: Openness and transparency	PP8: Data minimisation	PP3: Use of personal information	PP2: Data Ownership	PP7: Topology agnostic
PP15: Data controllers	PP6: Trust between communities	PP9: End-to-end privacy	PP4: Protection of personal information	PP11: Use of pseudonyms	PP10: Offline working
PP22: Trusted intermediary	PP12: Provenance	PP17: Authentication			PP20: Resilience
	PP13: External services	PP18: Multiple persona			
	PP14: Audit	PP19: Sub-groups			
	PP16: Objective and subjective trust				
	PP21: Diversity				
	PP23: Trust				

7.1.2 PP1: Compliance with legislation

The PICOS Architecture must be compliant with all legislation, regulation and best practices that exist in the geographical regions in which the Community operates

As a minimum, information is handled and processed according to the EU Data Protection Directive.

7.1.3 PP2: Data ownership



The PICOS Architecture must recognise that personal information belongs to the Member that the information uniquely identifies

Members explicitly grant others, including the Community Manger (if one exists), the right to store and process their data according to the Member's stated privacy and data handling preferences. The PICOS Architecture ensures that processing is proportional to the stated purpose, and that the Principle of data minimization is respected. For example, members may grant permission to the community operator to store and process their data according to the member's stated privacy and data handling preferences.

7.1.4 PP3: Use of personal information

The PICOS Architecture must provide members with the facility to state how their personal information can be used by others and, as far as is technically, legally and operationally possible, uphold the member's wishes w.r.t. information flow and processing

Members state conditions that dictate how their personal information can be used by other Members. Conditions are enforced by the Architecture.

The degree to which control can be enforced is probably limited to within the community boundary, unless Digital Rights Management technology is deployed at the client or at third parties who are authorised to process member data.

7.1.5 PP4: Protection of personal information

PP$_{Control}$

The PICOS Architecture must at all times protect personal information to the level selected by the Member

Three classes of data are supported: non-personal data, personal data and sensitive-personal data. Personal data might include home address, telephone number, while personal-sensitive data include medical records.

The classification of each data item is decided by the owner of the data (typically the Member). Classification is subjective and difficult to define, but the approach just suggested (non-personal data, personal data and sensitive-personal) is in widespread use.

7.1.6 PP5: Openness and transparency

PP$_{Trust}$

The PICOS Architecture must offer services to Members in an open and transparent way

Members will be more trusting if they fully understand the implication to their privacy of using services that handles their personal information.

7.1.7 PP6: Trust between communities

PP$_{Trust}$

The PICOS Architecture must recognise trust as a common currency when exchanged between PICOS communities

Members may belong to several PICOS communities. They will expect a seamless experience when interacting with Members across community boundaries, recognising that different communities focus on different 'themes', and different member's values, rules, behaviours. Portability of trust (or reputation) is highly desirable.

7.1.8 PP7: Topology agnostic

PP$_{Other}$

The PICOS Architecture should not be topology-specific

The Architecture is designed to be implemented on a range of interconnection (network) topologies, recognising that not all features are applicable in some configurations.

7.1.9 PP8: Data minimisation

PP$_{Privacy}$

The PICOS Architecture must support the concept of data minimization. Only data absolutely necessary for the provision of the Service should be collected.

While recognising that data minimisation is a principle adopted in European law, PICOS also appreciates that data is required in order to allow a community to grow. For example, Web 2.0 services are data-rich. A challenge for PICOS is to achieve an acceptable balance between these two demands. A solution may lie in the formation of trust, which allows greater use to be made of information in the knowledge that it is unlikely to be misused. Also, over time, we know from past research that trust itself grows, and community members are more comfortable sharing personal information.

7.1.10 PP9: End-to-end privacy

PP$_{Privacy}$

The PICOS Architecture must support end-to-end privacy

Where interacting Members are concerned that a central authority may be able to compromise (read/modify) their private interaction, the architecture should offer Members the option for end-to-end privacy (encryption?) subject to, and in compliance with, any legal obligations placed on the community operators (e.g. communication interceptions).

End-to-end privacy will not be required in all situations, and the trust that is placed in a community operator will be sufficient for most member needs (and reinforced by legislation). Also, providing end-to-end privacy makes law enforcement difficult or impossible, and therefore needs to be conditional.

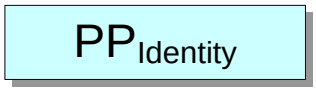
7.1.11 PP10: Offline working

PP$_{Other}$

The PICOS Architecture must support online and offline working, and easy transfer between the two states

For a variety of reasons, Members may need to operate when disconnected from the Community, or be able to connect with a subset of the Community. Members will wish to be able to protect and process personal information belonging to them or to other Members when offline, and be assured the same level or protection as when connected (online) to the Community.

7.1.12 PP11: Use of pseudonyms



The PICOS Architecture must present Members with the facility to be anonymous, to use pseudonymous identities or to use identities that are legally binding to that Member

Members will wish to interact with other Members and services, while still able to restrict how much identifying information is shared. They may vary the information shared for each interaction or vary it during an interaction. This is to allow Members to express their opinions with greater freedom, and to 'experiment' while they build confidence and trust in services and other Members. Note: While experimenting in isolation may be acceptable, using anonymising technologies when interacting with other Members may breach community operating practices.

Members may choose to be anonymous when providing feedback, or may have no choice if this is the default operating policy for the community. There are several possibilities. For example, reputation could be provided anonymously, but feedback intended to improve the community may identify the contributing member, and thus affect the contributing member's own reputation.

7.1.13 PP12: Provenance

The PICOS Architecture must ensure that Members can rely on the provenance of information that they receive from other Members / PICOS communities, subject to the Member choosing to state the provenance and there being no conflict or risk of undermining other privacy principles.

While it is probably too difficult to guarantee that information shared between Members is accurate, being able to rely on the source of the information is important for trust and reputation services. Note: This does not necessarily imply that the receiver of the information must be able to identify the originated, since the information alone may be trustable, e.g. because the source (not necessarily the sending member) is known and/or the content has been independently verified.

Where the source is not explicitly stated, the PICOS community may be able to give additional information about the level of trust (e.g. the reliability of the source, its profile or reputation rating.

7.1.14 PP13: External services

PP$_{Trust}$

The PICOS Architecture must ensure that externally hosted services are delivered in as trustworthy a way as an internally hosted Service, or that Members are aware when an external service is (potentially) less trustworthy than an internal service

Members may use services hosted by the Community to which they are currently connected, and Service provided by other PICOS communities. Ideally, all communities would operate at the same trust level, but in practice this is unlikely. Members should be able to determine from the nature of the Service, and not from a dependence on the hosting environment, how much trust to place in a Service.

Where it is hard to determine the trustworthiness of an external service, an indicator explaining to members that the service is provided externally may be sufficient.

For example, a contract (i.e. SLA) may certify that that the hosted service provider uses specific security technologies, trusted infrastructures, standardised procedures, etc.

7.1.15 PP14: Audit

PP$_{Trust}$

The PICOS Architecture must allow all services to be fully auditable by an entity trusted by all Members

If something goes wrong, Members will expect to be able to recover and prevent a repeat of the event. Members will also expect accountability, both at Member and (if applicable) Community Operators level. There may be Legal or regulatory requirements to provide auditing for some community applications.

7.1.16 PP15: Data controllers

PP$_{Law}$

The PICOS Architecture must identify the controlling entity(ies) who are obliged to fulfil Legal obligations concerning the Community

For example, the Police may need to serve a Legal notice that obligates the Community to supply data about Members. Other entities will have similar statutory rights, e.g. the removal of copyright protected, illegal or defamatory information.

7.1.17 PP16: Objective and subjective trust

PP$_{Trust}$

The PICOS Architecture should support both objective and subjective methods for assessing trust

Subjective methods include reputation management services. Objective methods include trusted computing bases and reputation management systems that are based on hard facts, e.g. system measurements, attributable actions and evidence of event fulfilment.

7.1.18 PP17: Authentication

PP$_{Privacy}$

The PICOS Architecture should support multiple forms of Member authentication, while continuing to respect privacy

Authentication should be possible using one, two and three factor (know/possess/are) methods. Health-related information must be adequately protected, treated as personal-sensitive information and respected according to the conditions stated by the Member.

7.1.19 PP18: Multiple persona

PP$_{Privacy}$

The PICOS Architecture should allow Members to have multiple persona

Members may want to operate within their PICOS Community, and between PICOS communities, under different identities (partial identities). One justification for this is to enhance privacy, for example by limiting linkability.

7.1.20 PP19: Sub-groups

PP$_{Privacy}$

The PICOS Architecture must support the creation of sub-groups within the Community

If you take the Taxi Driver Community, the three taxi drivers operated independently of their drivers and (if included within the PICOS Community) their passengers.

7.1.21 PP20: Resilience

PP$_{Other}$

The PICOS Architecture must not have a single point of failure

For example, it should not have a centralised information store, single authentication point or single management function.

7.1.22 PP21: Diversity

PP$_{Trust}$

The PICOS Architecture should be designed in such a way that no single entity can act in a way that might compromise the trust and privacy of the community

This does not relate to the general day-to-day management of the community, where placing the responsibility with a single entity does not represent a significant risk to the community, Member privacy or trust.

Keeping the community operational is something that a single entity could be responsible for, in line with the Service Level Agreement. Revealing anonymised members identities for law enforcement purposes might be something that requires the community operator to liaise with a TTP.

7.1.23 PP22: Trusted intermediary

PP$_{Law}$

The PICOS Architecture permits several trusted intermediaries (including external TTPs) to co-operate and link partial to real identities

This will almost certainly be required for legal purposes, and may be necessary for other purpose, e.g. to enhance reputation through external assurance or split responsibilities, and to provide non-repudiation services.

7.1.24 PP23: Trust

PP$_{Trust}$

The PICOS Architecture should ensure that Members are accountable for their actions while a member of the Community

All activities within the community are logged, and contributions are linkable to a real-world identity (by the community operator under a split role).

8 PICOS Features

8.1 Introduction

Based on earlier requirements gathering exercises, notably deliverables D2.3 and D2.4, we have identified a short-list of key features that we believe members of a PICOS community will value most. The features are derived from our examination of all our reference communities, but in keeping with the focus of this first architecture deliverable, we aim to satisfy the needs of the angling community as fully as possible.

We begin by expressing these features in terms of the benefit that they offer members. Later in this section, we describe each feature in greater detail, indicating the implications for an implementer of a PICOS community.

8.1.1 Key to features

Each feature is categorised according to the contribution that it makes to a PICOS community with respect to communities in existence today. Each is assigned an appropriate icon as follows:

PICOS introduces the new community feature

PICOS enhances this traditional community feature

PICOS enables mobility through this feature

8.1.2 Features most valued by members

We believe that the best way to express member requirements is in terms of what they want from their community.

We believe members would say, “We want to”:

- Share information (content) with other members
- Send messages to other members and, in general, have access to a range of different communication services, including real-time interactive ‘instant messaging’ and push-pull notification (e.g. voice/text messaging)
- Search for 1) members with similar interests and 2) information on specific topics
- Create or join sub-groups of members within the community

- Build trust in other members through the use of 1) reputation, 2) non-repudiation and 3) closed membership (registration)
- Access external services that offer specialist functionality not normally expected of my community
- Mark (tag) documents in such a way that I can restrict how they are used and who has access to them
- Receive notifications that will help me understand when I am at risk or need to perform an action to protect myself or my information
- Interact with other communities on the same basis as I interact with member of my own community
- Have essentially the same experience whether I am mobile or static
- Personalise my experience and expectations of the community based on my particular requirements and values.

These eleven high-level features are expressed in a way that makes sense to the membership. From this list we now identify a set of system features that would need to be implemented to satisfy these higher level goals.

8.1.3 Main system features

The system features that we believe will satisfy the needs of a PICOS community are derived for the earlier list of feature most valued by members:

Feature	Description	PICOS$_{mobility}$
1	Reputation	
2	Content sharing	✓
3	Registration	
4	Personalisation	
5	Messaging	
6	Searching	
7	Sub-communities	
8	Presence	✓
9	External services	✓
10	Content tagging	
11	Communication services	✓
12	Notification	✓
13	Inter-community interaction	
14	Mobility	✓
15	Non-repudiation	

The features marked with a check mark (✓) are considered to have particular, possibly unique significance for a mobile community.

In the remainder of this section we examine each feature in detail, and explain how PICOS will address the privacy and trust concerns that naturally arise.

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

8.1.4 Summary of PICOS features

PICOS$_{enhancing}$	PICOS$_{distinguishing}$
PF1: Reputation	PF10: Tagging
PF2: Content sharing	PF14: Mobility
PF3: Registration	PF15: Non-repudiation
PF4: Personalisation	
PF5: Messaging	
PF6: Searching	
PF7: Sub-communities	
PF8: Presence	
PF9: External services	
PF11: Communication services	
PF12: Notification	
PF13: Inter-community interaction	

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

8.2 PF1: Reputation

PICOS$_{enhancing}$

History: Feature contributor: UMA

8.2.1 Description

Reputation covers rating and feedback. PICOS enables a community to keep track of user behaviour by computing a reputation indicator (typically a single value) for each individually identified member. This reputation indicator is produced for each pseudonyms generated under a member's real identity, and shared with other members without revealing the real identity in a way that might link pseudonyms.

In addition, a reputation component is able to aggregate reputation indicators from external communities and combine with 'local' community indicators, assuming that a meaningful association can be made. This process is initiated 'on demand', solicited by the requesting member.

Since different communities may express reputation in different ways, a process of normalization may be required. This process also 'weights' the reputation indicator associated to a given identity, and then aggregates the normalised values to obtain a single personal (i.e. subjective) reputation indicator. This computation will be performed without revealing any private information about the identities being evaluated.

When members provide feedback to other community members, or rate community-related activities, their reputation indicator will be linked, and consequently may affect how other members value their contribution. For example, a rating coming from a member with low reputation could have little credibility. In fact, members may choose to filter out certain contributions by setting a threshold that defines minimum (lower bound) reputation for them to be accepted. The same technique can be applied to items (as opposed to members) to assist when searching for information within the community.

Reputation indicators are influenced by the feedback that members provide, and based on personal experience. Thus, positive feedbacks will increase member reputation, while negative feedbacks will decrease it.

To ensure that the reputation system is reliable, only registered/authenticated members are permitted to provide input that influences reputation indicators. This is required to avoid the possibility that the reputation system becomes devalued by false, incorrect or malicious feedback. It is likely that other checks will be required that strongly associate members with the actions that they choose to comment on.

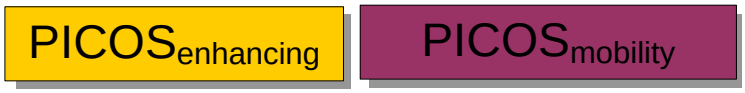
We talk about reputation indicators without precisely defining what form an indicator will take. This is because we want to allow for member to be able to customise the output to meet their particular purpose and social values. Customisation will be achieved using the member's personal profile information where, for example, weights may be specified and applied to the reputation computation.

8.2.2 How PICOS will address the privacy/trust/IdM concerns

PICOS will maintain privacy and build trust by:

- Accepting and properly handling cross-community reputation, and whenever possibly valuing transitive trust indicators
- Ensure that feedback originates from an identifiable source
- Ensuring that feedback can be provided anonymously
- Demonstrating that the reputation system is trustworthy by building on open (transparent), robust design principles
- Within the community, creating a culture that encourage constructive feedback, and eliminating non-constructive influences

8.3 PF2: Content sharing



History: Feature contributor: GUF

8.3.1 Description

Content sharing comprises various activities concerned with the exchange of different types of information within a community (inter-community) and across communities (inter-community). Sharing involves several distinct phases that contribute to the exchange process, namely contribution, storage, administration, manipulation, communication (notification) and distribution of content. In practice, a specific mechanism would be required for each function.

Content is a general term that we use to refer to generic information, which may be represented as text, graphics (pictures), albums, videos and audio data, personal messages. It may also be encrypted, with decryption being possible at a system or an individual member level.

- Contribution: The process of making content available to other community members, involving mechanisms to perform the uploading of information (files).
- Administration: Administering previously contributed content within the community, including tasks for (re-)structuring and managing content. Administration also enables members to set privacy requirement on content contributed, thus they can control access to content by other members.
- Manipulation: Provides mechanisms for the manipulation of contributed content, including the partial editing of content, renaming, tagging, and deletion.
- Communication: Allows the mode of sharing to be specified, e.g. direct member to member, member to group of members and member to forum, and indirect exchange via a central repository using a push-pull procedure.

8.3.2 How PICOS will address the privacy/trust/IdM concerns

PICOS will maintain privacy and build trust by:

- Providing mechanisms for members to control how their content is shared with other, including compliance with any regulatory or legal constraints that may apply
- Allow content to be tagged in various ways, and for tags to be evaluated by other members
- Enable sharing to be controlled according to member attributes, including context (e.g. location) and recipient properties (e.g. reputation), including intra-community sharing
- Provide recommendation on the possible risks associated with sharing, by 1) taking into account preference, personal profile, the profile of recipients, context, the nature of the information being shared and 2) helping members identify (search for) other members who

have similar interests and match the contributor's acceptable trust profile (where the trust profile includes a reputation threshold).

8.4 PF3: Registration

PICOS$_{enhancing}$

History: Feature contributor: ATOS, UMA

8.4.1 Description

Registration is the first point of contact for individuals who wish to use a PICOS community. It is where information about the individual is collected, where roles and privileges (rights) are assigned, and where information associated with the authentication of the individual (subsequently known as member) is assembled (e.g. passwords, cryptographic keys). Registration encompasses authentication, identity management and de-registration. It represents the first step in the lifecycle management of community members.

Once membership is confirmed, members can create different identities (pseudonyms) so that they can represent themselves in different ways within the community. For example, a given member may choose a different pseudonym for a specific context, and with that context wish to operate under different privileges or profiles.

Members act under different pseudonyms to protect their privacy, possibly simultaneously. Pseudonyms are not linkable, and to all other member each pseudonym appears as a distinct, unique member. Reputation is based on unique identities (real or pseudonym), enabling other members to establish (track) specific contributions and related activities. For convenience, members may choose to share/transfer profiles/privileges between their identities, but PICOS will ensure that unlinkability is never compromised.

In addition to having real and pseudonymous identities, members can choose to act anonymously. Anonymous identities should not be confused with members' identities being anonymised by the community, for example when contributing feedback to the reputation component.

Membership of the community can be revoked at any time, for example:

- When a subscription expires,
- If the member freely decides to 'resign' membership,
- If the member behaves dishonestly or breaches to terms and conditions under which membership was accepted

8.4.2 How PICOS will address the privacy/trust/IdM concerns

PICOS will maintain privacy and build trust by:

- Offering an authentication mechanism that guarantees that members are who they say they are, and consequently have rights to the claimed privileges
- Establishing a sound, reliable identity to which reputation can be built.

- Through identity, supporting non-repudiation, which ensures that members are accountable for their actions, even if performed under a pseudonym.
- Protecting identity by allowing pseudonymous (and to a lesser extent anonymous) transactions, which when correlated do not allow disclosure of personal, identifying information. This is achieved while still building trust by tracking member activities as part of reputation management.

8.5 PF4: Personalisation

PICOS$_{enhancing}$

History: Feature contributor: HPL

8.5.1 Description

Every community member will possess a personal profile. The profile will be partly public and partly private. It will describe members' unique characteristics and their shared interests, but it will also provide members with an opportunity to state on what basis they are willing to interact with other members and generally make use of the community.

We understand from earlier (e.g. D2.4) deliverables and previous research (e.g. Trustguide[8]) that community members value choice and the ability to express personal preferences when interacting with others members within the community.

The profile is more than just a list of requirements. If forms the basis of the Privacy Advisor which is designed to help members to determine trust and maintain privacy in a way that is personal and unique, befitting their personal, continually changing requirements.

8.5.2 How PICOS will address the privacy/trust/IdM concerns

PICOS will maintain privacy and build trust by:

- Managing personal profiles by enabling members to create, modify and delete their personal profile
- Controlling visibility by enabling members to define exactly what information placed in the profile can be viewed by other members, groups, the community operator and external service providers.

[8] Trustguide. Final Report. Research carried out by HPL for the UN Government, investigating Trust in ICT. www.trustguide.org.uk

8.6 PF5: Messaging

PICOS$_{enhancing}$

History: Feature contributor: ITO

8.6.1 Description

Note: We use the term 'messaging' to refer to the exchange of information between an identifiable originator and recipient(s). We exclude what might be best described as broadcast systems, under which we including blogging, wikis and message boards. We accept that in a closed PICOS community, where membership is reliant on registration, and consequently everyone is identifiable, this constraint does not hold. However, we believe that where privacy is concerned there is an expectation that all parties in an exchange will already be known ('a posteriori') by name or personal characteristic to the originator.

Messaging is the exchange of information over a distributed and potentially unprotected medium, e.g. the Internet. We identify three distinct forms of relationships between members:

- One-to-one (1:1), e.g. via Instant Messaging (IM), private chat room
- One-to-Many (1:n), e.g. one member sends a message to which many members respond

The third case, which we exclude from the discussion on messaging, is:

- Many-to-Many (n:m), e.g. a public chat room where many members create messages simultaneously which receive many responses.

Originators of messages decide who can read, modify and/or forward their message to other members within the community. Originators remain the owner of messages sent, and can view records (logs) that show who has accessed a message, and can request to be notified automatically.

Logging of message is limited to message routing information only. Message content is not recorded unless specifically requested by the originator. Respondents to messages must be alerted to the fact that content logging is occurring before they reply.

Owners set access rights and apply other restriction, e.g. no print or local storage permitted. Special conditions may apply to messages sent to third parties.

To maintain privacy, message transmitted over unprotected medium are encrypted, thus preventing messages from being read or altered by anyone not authorised to do so.

8.6.2 How PICOS will address the privacy/trust/IdM concerns

PICOS will maintain privacy and build trust by:

- Encrypting information that passes over unprotected mediums
- Ensuring that recipients are identifiable to the satisfaction of the originator

- Observing originator preferences which may include reputation threshold requirements
- Logging events relating to message access and optionally logging content

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

8.7 PF6: Searching

PICOS$_{enhancing}$

History: Feature contributor: IfM-Geomar

8.7.1 Description

Searching for information on the Internet is an exciting issue. It seems that there is nothing that users want that cannot be found, and there are billions of users who benefit everyday from the Internet. However, when searching in the entire Web, a specific website or an online community, users leave behind traces which allows their visit to be tracked. Their behaviour and interests can be observed, and potentially (mis)used by third parties. For example, simply browsing an online retailer can lead to the creation of a personal profile (interests, related products, etc.) which may result in users receiving unwanted advertising when they next visit the website, or unsolicited e-mails.

A quote from the New York Times reads, 'It may be easy to forget that there are people who want to remain anonymous on the Web while the online world is full of those who happily post pictures of themselves and their navels for all to see. But interest in software that allows people to send e-mail messages that cannot be traced to their source or to maintain anonymous blogs have quietly increased over the last few years, say experts who monitor Internet security and privacy.'[9]

Searching includes searching the community both for members and for content. It can relate to a specific, unique target or to targets that satisfy a set of conditions. It is an 'internal' service, that helps PICOS function, and it is a member service that supports many of the features that are offered to members.

8.7.2 How PICOS will address the privacy/trust/IdM concerns

PICOS will maintain privacy and build trust while searching by:

- Protecting against traffic analysis: developing measures that allow individuals to share information over public networks without revealing their privacy.

  Examples of techniques which PICOS could exploit include an adaptation of 'onion routing' (e.g. TOR), where users transactions are distributed across the Internet so that transactions cannot be easily linked to the member identity.

- Cookies management: HTTP cookies have been, and continue to be widely used when searching the Internet. They are used for authenticating, session tracking and maintaining specific information about users, such as site preferences or the contents of their electronic shopping carts. Cookies have brought personalization, commerce, and convenience to website browsing but have disadvantages which PICOS will address, e.g. unauthorised intercepting (cookie hijacking), unauthorised modification (cookie poisoning) during

[9] Citation from: "Privacy for People Who Don't Show Their Navels", by Jonathan D. Glater for the New York Times.

transfer between user and service provider; unauthorised cross-site ‘cooking’ (preventing a cookie for one website being transferred/used at another website) and providing user control over cookie expiration.

- Implementing, disposable, temporary identities which allow members to receive responses to search requests without needing to reveal their permanent or regularly used identity.
- Not breaching non-linkability rules

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

8.8 PF7: Sub-communities

PICOS$_{enhancing}$

History: Feature contributor: BRNO

8.8.1 Description

Functionality to create and manage sub-groups is one of the valued characteristics of community services and also one of the PICOS key features. Sub-groups allow easier communication and content sharing and form an important factor in the fine grained access control to data. Sub-community feature covers the creation and management sub-groups, e.g. buddy list, family & friends.

8.8.2 How PICOS will address the privacy/trust/IdM concerns

PICOS will maintain privacy and build trust by:

- Ensuring that only community member with appropriate rights can create, manage and delete sub-groups
- Ensuring that a community member with appropriate rights can remove unwanted members from a particular subgroup
- Supporting both global (e.g. projects, special interest groups) and personal subgroups (friends, family, etc.)
- Allowing Sub-groups to be categorised as open, restricted (membership by owner approval) and invitation-only. Selection can be made by member identifier or other characteristics, e.g. context or reputation
- Restricting visibility of sub-group membership to the owner and other members, as agreed with the specific member concerned. Such restrictions may form part of the member profile or preferences. Thus, membership may be visible to all members, only approved sub-group members or specified sub-group members.

8.9 PF8: Presence

PICOS$_{enhancing}$ PICOS$_{mobility}$

History: Feature contributor: ATOS

8.9.1 Description

PICOS defines Presence as a combination of online status and context information, e.g. location. Presence is tightly coupled to the communication services such as chat (Instant Messaging – IM) and indicates a member's availability for conversations with other members. Members can control both community-wide visibility or choose to restrict/permit visibility to specific members.

Presence can affect how trust and privacy is provided. For example:

- Privacy (more accurately referred to in this situation as confidentiality) is respected in many aspects of a presence system: Members may choose not to reveal to the community that they subscribe to certain services; Members may not want to reveal that they reveal their status to certain members; Members can conceal from others the information that they retrieve from information providing features like Presence.
- Confidentiality is provided through a combination of hop-by-hop (point-to-point) and end-to-end encryption. The hop-by-hop mechanisms provide scalable confidentiality services, disable attacks involving traffic analysis and hide all aspects of presence messages. However, since they typically operate on the transitivity of trust, they may cause message content to be accidentally revealed to proxies. The end-to-end mechanisms do not rely on transitivity of trust, and only reveal information to the desired recipient. However, end-to-end encryption cannot hide all information, and is susceptible to traffic analysis.
- Strong end-to-end authentication and encryption can be achieved using asymmetric (public key) cryptography, while end-to-end encryption is easier achieved with symmetric (secret key) cryptography. Hop-by-hop and end-to-end mechanisms are required to address privacy concerns that the Presence feature may give rise to. For example, the SIP protocol (used to access services offered by most mobile phone operators uses hop-by-hop encryption, whereas TLS (an end-to-end scheme) is the default (and often only) option on Web-like servers, leaving TLS as the obvious choice. An alternative is to encrypt SIP messages using S/MIME.

8.9.2 How PICOS will address the privacy/trust/IdM concerns

PICOS will maintain privacy and build trust by:

- Providing members with choice over who can see their Presence information
- Protecting information provided by the Presence service using strong encryption

- Strongly authenticating members (or their appliance) when requesting Presence information, thus the authentication may vary with presence (e.g. location).

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

8.10 PF9: External services

PICOS$_{enhancing}$ PICOS$_{mobility}$

History: Feature contributor: TMO

8.10.1 Description

Communities benefit by the introduction of external services and resources, for example advertising, licensing, and traffic info. The overall functionality, from the member's perspective is enhanced and new application, use cases and business models can be supported, overall increasing the attractiveness of the community.

By providing information about members to advertising partners, in a privacy respecting way, members can receive information and offers that closely match their particular interests. This information can be integrated in existing or new services. For example, traffic information would greatly benefit one of the PICOS reference communities, namely Taxi Drivers, enabling them to optimise tours and find the quickest way to a destination or pick-up point.

PICOS will provide open, generic and flexible interfaces to external service providers, catering for both new and legacy features.

8.10.2 How PICOS will address the privacy/trust/IdM concerns

PICOS will maintain privacy and build trust by:

- Protecting member data delivered to external partners based on member privacy preferences profiles, and by not identifying members uniquely outside of the PICOS community
- Respecting the rights of members, as established in legislation (e.g. the EU Data Protection Directive) when personal information is shared with third parties
- Providing external services (especially advertising) to members on an opt-in basis, so that members can assess and decide on the added benefit that an external service may offer then
- Logging interactions (but excluding personal information) exchanged with external services
- Ensuring that external services are closely monitored and their operation is described to members in an open way so that members can make informed decisions on whether to subscribe to a service or use its features
- Handling complaints about external services without revealing more user data than necessary

8.11 PF10: Content tagging

PICOS distinguishing

History: Feature contributor: GUF

8.11.1 Description

Content tagging covers the association of 'type indicators' (or tags) to information processed by the PICOS community membership. For example, tags might describe information that provides a location or that indicates that a member is mobile. Tags are semantic text elements, or meta-information, that describes content. They can be applied to a wide range of content including documents, video, audio and community operational information. They may be used in order to identify information received from external third parties, e.g. advertisers. They typically describe content in greater detail, and can relate to specific items, e.g. names of people in a photograph and the setting (e.g. 'summer holiday 2008'). The semantic nature of tags serves as a basis for a PICOS community to efficiently manage many different types of content the members may generate, e.g. organising, searching and making recommendations.

PICOS adopts three types of tag to ensure that members have adequate control over the content that they manage:

- Personal tagging: Only the member who contributed the content can tag the content
- Restricted tagging: Only members specified by the content contributor can tag the content
- Unrestricted tagging: All members of the community can tag to the content.

Mobility is an important feature of the PICOS community, so naturally tagging of location based information is regarded as highly desirable and important. Location based tags are tags that include information about the current location, e.g. 'geo' coordinates; location information and additional context information, e.g. time, place and mobile appliance, could be offered to members so that they can tag content in a way relevant to their mobile lifestyle.

Tagging also has a role to play in access control and in expressing context sensitive privacy preferences. For example, a member may tag personal health information so that any medical professional can access it, but only when the member is located abroad or away from their own regular Doctor.

8.11.2 How PICOS will address the privacy/trust/IdM concerns

PICOS will maintain privacy and build trust by:

- Using tags to restrict access to specific content that they contribute to the community
- Allowing members to decide how tags are applied and who can subsequently alter these tags.

8.12 PF11: Communication services

PICOS$_{enhancing}$ PICOS$_{mobility}$

History: Feature contributor: HPF

8.12.1 Description

Communications services cover a wide range of member-to-member(s) real-time, video and audio (chat) services. PICOS assumes that mobile communities will (at least initially) make extensive use of mobile communication services, e.g. GSM. The integration of mobile communication services with a community service raises new privacy and trust issues, but also provides members with increased convenience when interacting with their community. For example, it will:

- Support anonymous communication, allowing members to interact using mobile communication services without exposing mobile phone numbers or other identifiers, e.g. by replacing a Caller ID with a member identifier (possibly pseudonymous, and ensuring that no personal identifying information is revealed in received call / missed call lists, or detailed in billing records, etc.
- Enforce privacy policies when sharing member-generated content within multi-media communication sessions. For example, a pre-defined set of members who are permitted to access/view photographs or video clips, would still apply when those same images are present over a mobile communication channel.
- Enforce member policies with regard to reachability, i.e. control how a member can be contacted when they are using specific communications mechanisms (messaging, voice, video), and when other constraints apply, context, location, 'buddy lists'.
- Provide privacy-aware caching of community information at the mobile device. Information might include contact information, and additional restrictions may apply after transfer, e.g. allow download but prevent subsequent forwarding to other members.
- Support "anonymous" communication, where members can communicate peer-to-peer using their community member identifiers (i.e. without disclosing phone numbers or other identifiers)
- Enforce privacy preferences, which are applied before sharing member-generated content over a real-time communication services

Ideally PICOS will embed (integrate directly into the PICOS architecture) most communications services, some may be offered by a third-party providers, e.g. a mobile operator. Tight integration is preferred because it means that all communication is triggered from within the community and consequently privacy policies can be instantly applied. Communication that takes place outside of the scope of the community platform raise questions about how policies can be applied and tightly coupled to content.

8.12.2 How PICOS will address the privacy/trust/IdM concerns

PICOS will maintain privacy and build trust by:

- Respecting privacy preference and content tags regardless on the communications service offered, or of the form in which the content is presented to the member
- Anonymising personal information that would otherwise reveal the identity of those engaged in the communication.
- Controlling the application of content communicated to members by enforcing preferences applied by the originator, e.g. read but no onward sharing.

8.13 PF12: Notification

PICOS$_{enhancing}$ PICOS$_{mobility}$

History: Feature contributor: HPF

8.13.1 Description

Previous deliverables (e.g. D2.3, D2.4) and other research (e.g. Trustguide) revealed that members are more likely to trust a community when they understand how services operate. Describing how personal content is processed, in an open and transparent manner, engenders trust. Thus, keeping members informed about the status of the community, their outstanding transactions or other information that relates to their personal use of the community, through the use of notifications and alerts increases confidence.

8.13.2 How PICOS will address the privacy/trust/IdM concerns

PICOS will maintain privacy and build trust by:

- Reporting status information covering:
  - System and service delivery status
  - Community status
  - Member status
  - Inter-community status
  - Activity status
- Enhancing decision making by providing information covering:
  - Context-based, risk indicators
  - Past performance derived form historic status information
  - Alignment with personal profile
  - Previous actions/decisions
- Offering advisory information covering:
  - Events arising from real-time status information
  - Significant changes in community structure
  - Member activity
  - Deviation from personal profile

8.14 PF13: Intra-community interaction

PICOS$_{enhancing}$

History: Feature contributor: BRNO

8.14.1 Description

The aim of this feature is to allow identities created in one community to be used in several other linked communities, while maintaining a coherent level of privacy and security. Inter-linking of communities may imply non-trivial organizational and implementation issues and will require a certain level of trust among the inter-linked communities.

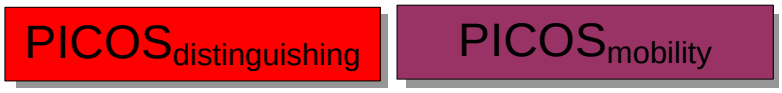
Other attributes can be similarly transferred across community boundaries, including privileges and reputation information, such that the identities in each community refer to the same real individual. This forms part of the federated identity management feature of PICOS, where members of one community may have automatic right to access another community based on their reputation. In such a case, registration is not required, except with the first community a member joins, and privileges are transferred automatically to any additional community. An example can cover situations where an angler based in Germany wants to access resources from an angling community in France without the need to first register on the French system.

8.14.2 How PICOS will address the privacy/trust/IdM concerns

PICOS will maintain privacy and build trust by:

- Only interacting with neighbouring communities if the Administrators of both communities agree on the inter-community linking
- Setting security level for all connected communities that meet certain minimal requirements, e.g. standard for the encryption of transmitted passwords, the level of security applied to servers
- Ensuring that the identity of a member can be used in other communities only if the user has expressed consent with such an identity sharing
- Allowing each member to choose whether they wish to explicitly approve each additional community linked to, or letting members configure their personal profile to automatically include all inter-linked communities (or some specific groups based e.g. on subject, location).
- Mutually respecting access restriction imposed by member of neighbouring communities
- Ensuring that exchanged content can only be used as instructed by the contributor
- Not linking a member to any community that they have expressly asked not to be connected to.

8.15 PF14: Mobility



History: Feature contributor: TMO

8.15.1 Description

PICOS supports mobility. Mobile devices present new challenges to an otherwise static community: different technologies and additional use cases.

The different technologies comprise the communication network and the devices. The network uses different methods to transport data (not always transparent to the application and user) and provides additional services including short messaging, network-based authentication, and location and presence services. Also, the devices are adapted for mobile use and typically have reduced size, limited usability, reduced power and less storage.

Mobility enables new use cases. Members now have access to the communities at any time and at any location. They might be using different devices for mobile and static access, and consequently want to synchronise the data held by their two appliances. Based on additional member data, e.g. current location, new services can be linked existing use cases offering enhanced features for the member.

8.15.2 How PICOS will address the privacy/trust/IdM concerns

PICOS will maintain privacy and build trust by:

- Securing connections over which messages pass between members
- Offering strong user authentication based on network attributes or temporary credentials
- Identity management and the inter-working of mobile devices and stationary devices
- Making full use of additional context attributes, e.g. location, presence, device capabilities
- Adapting the way that information is presented to members when using mobile devices to enhance usability.
- Responding to threats that are specific to mobile situations
- Controlling information flow between members via the PICOS community, to maximise member privacy

8.16 PF15: Non-repudiation

PICOS distinguishing

History: Feature contributor: UMA

8.16.1 Description

PICOS offers the capability to ensure that actions performed by members cannot be later disputed (repudiated). This is achieved by using digital signatures. Every contribution made by a member is digitally signed by a public key that is either directly or indirectly bound to the member's real identity. If a member contributes content using their pseudonymous or anonymous identity, then a continuous non-repudiated chain can only be constructed with the cooperation of one or more trusted third parties (TTPs), who together may be able to collude to compromise a member's anonymous or pseudonymous identity.

Note that non-repudiation can only be guaranteed if members individually control their private cryptographic key. It is crucial that any implementation provides sufficient protection to guarantee this requirement. Equally, it is important that bindings are not compromised registering unlinkable pseudonyms or performing anonymous actions.

If non-repudiation as provided supports a legislative obligation, then it is important that members are registered with the community using an externally provided non-repudiated mechanism, such as X.509 strong authentication and X.509 Public Key Certificates issued by a legal Certification Authority (CA).

8.16.2 How PICOS will address the privacy/trust/IdM concerns

PICOS will maintain privacy and build trust by:

- Enhancing member confidence by ensuring that member cannot repudiate contribution specific content.
- Ensuring that pseudonyms and anonymous identities cannot be compromised through weaknesses in the binding mechanism introduced by the non-repudiation mechanism.

9 PICOS Components

9.1 Introduction

Forty-nine actual components have been identified from the requirements gathering staged (as reported in D2.4) of the project, which are believed to be necessary to create the PICOS architecture. Each component is categorised according to one of five broad component headings, namely:

- Services and Applications
- Content Handling
- Member Administration
- Communication
- Audit, Control and Reporting

The five component groupings lead to a simple model for representing the organisation of the PICOS architecture, which we call the PICOS 5-layer Architecture Model.

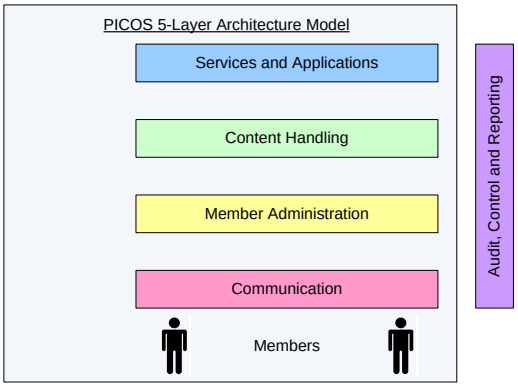


Figure 15 – PICOS 5-Layer Architecture Model

Components are assigned to 'tiers'. The component groupings are referred to in PICOS as Tier-0 functionality.

Individual components are described as either Tier-1 or Tier-2, depending on the breadth of functionality that they offer. In general, where a component relies on one or more other components for most of its functionality, i.e. the component coordinates interaction with other (subservient) components, or provides a coordinating function, it is called a Tier-1 component. The subservient components are referred to as Tier-2.

In addition to the five Tier-0 'grouping' components, there are nine Tier-1 components and four Tier-2 components.

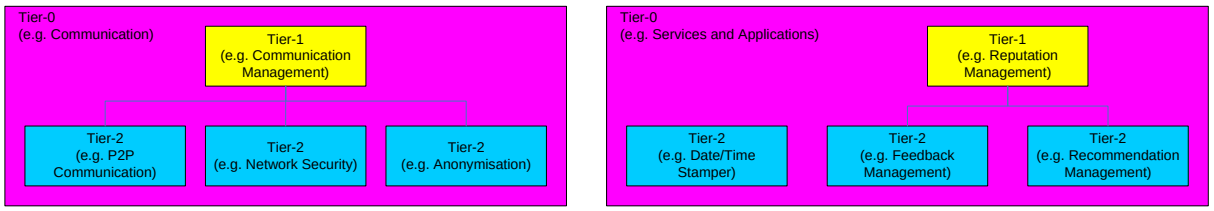


Figure 16 – Example of component Tier levels

In this section we present the purpose and description of each component, and show the 'first level' relationship (inter-dependency) between components.

Note that we do not show every connection to every component. In particular we do not show connections to components where it is obvious from the context that such a connection would exist in practice, e.g. to the Event Logging and Audit components.

9.2 Component categories

As previously mentioned, each component is categorised as Tier-1 or Tier-2 (Tier-0 refers to a component grouping). In addition, each component is categorised, and assigned an appropriate icon, according to the contribution that it makes to a PICOS community with respect to communities in existence today. Components that represent a research opportunity for PICOS are also highlighted.

T_0 Tier 0 Component Grouping

T_1 Tier 1 Component

T_2 Tier 2 Component

$PICOS_{distinguishing}$ PICOS introduces the new community component

$PICOS_{enhancing}$ PICOS enhances this traditional community component

$PICOS_{research}$ Research within PICOS required. (Components requiring research are unlikely to figure strongly in the first prototype.)

9.3 Overview of PICOS component by contribution

Title	Tier	$PICOS_{enhancing}$	$PICOS_{distinguishing}$	$PICOS_{research}$
Audit, Control and Reporting	0			
Communication	0			
Content Handling	0			
Member Administration	0			
Services and Applications	0			
Access Control	1	✓	✓	
Application Orchestrator	1		✓	
Audit	1	✓		
Communication Management	1			
Identity Lifecycle Management	1	✓		
Importer/Exporter	1	✓		
Intrusion Detection	1	✓		
Preparation Area	1		✓	
Sub-community Management	1	✓		
Accountability	2		✓	✓
Anonymisation	2		✓	
Authentication	2	✓	✓	
Authentication Method Selection	2	✓		
Authorisation	2	✓	✓	
Consent Management	2		✓	
Content Sharing	2	✓	✓	
Cryptography / Key Management	2	✓		
Data Minimisation	2		✓	✓
Date/Time Stamper	2			
Delegation	2	✓		
DRM	2	✓		
Event Logging	2	✓		

Title	Tier	PICOS$_{enhancing}$	PICOS$_{distinguishing}$	PICOS$_{research}$
Event Reconstruction	2		✓	
External Recommendation	2		✓	✓
External Service Delivery	2	✓		
Feedback Management	2		✓	
Identity Translator	2		✓	
Linkability	2		✓	✓
Location Sensor	2	✓		
Network Security	2	✓		
Non-repudiation	2	✓		
Notification	2	✓		
P2P Communication	2	✓		
Partial Identity Management	2		✓	✓
Payment Services	2	✓		
Policy Management	2		✓	✓
Privacy Advisor	2		✓	✓
Privilege Management	2	✓		
Profile Management	2		✓	
Recruitment	2		✓	
Registration	2	✓		
Reputation Management	2	✓		✓
Revocation	2	✓		
Scenario Management	2		✓	
Secure Repository	2	✓		
Service Selection	2	✓		
Social Presence	2		✓	
Trust Negotiation	2		✓	✓
TTP Management	2	✓		

9.4 Communication

The Communication component group contains the following components

9.4.1 Tier-1 Communication components

- Communication Management

9.4.2 Tier-2 Communication components

- Network Security
- P2P Communication

9.4.3 Communication Management

History: *Component contributor: HPL*

PICOS Principle (PP): 1, 4, 7, 10, 20

PICOS Feature (PF): 11

9.4.3.1 Purpose

The *Communication Management* component is responsible for providing and co-ordinating communication between members and the PICOS community.

9.4.3.2 Description

The *Communication Management* component provides a level of abstraction above other specific communication technologies. Whenever members or external services (service providers) need to communicate with the community, this component chooses the most appropriate set of mechanisms to establish the communication. This component is also responsible for managing the security of the communication, which is achieved by calling on the services of the *Network Security* component.

Example 1: An incoming communication is requested by a member via the *Service Selection* component. The *Communication Management* component detects the request and activates the appropriate communication medium, e.g. GSM, Wi-Fi, Bluetooth. The type of network security available may vary depending on the medium chosen, thus several steps may be required to establish a secure channel (encryption algorithm negotiation, key sharing, authentication, etc.). The *Communication Management* component is responsible for handling this detail, and will be expected to do so in a way that is transparent to other services.

Example 2: A member wishes to communicate directly with another member (P2). The *Communications Management* component will receive the request, and then established a secure channel using the *P2P Communication* component. As in Example 1, security will be achieved using the *Network Security* component under the direction of the *Communication Management* component.

A need may also arise to provide anonymous network connectivity, in which case the *Communication Management* component will call on the services of the *Anonymisation* component to support (for example) a TOR protocol.

9.4.3.3 Dependencies

Components that this component calls	Purpose
Anonymisation	To support use of a TOR protocol.
Network Security	To establish a secure, authenticated channel.
P2P Communication	To enable P2P communication between members

Components that call this component	Purpose
Service Selection	When requested by a member of another entity to establish a communication channel with another member or entity.

9.4.3.4 Drawing

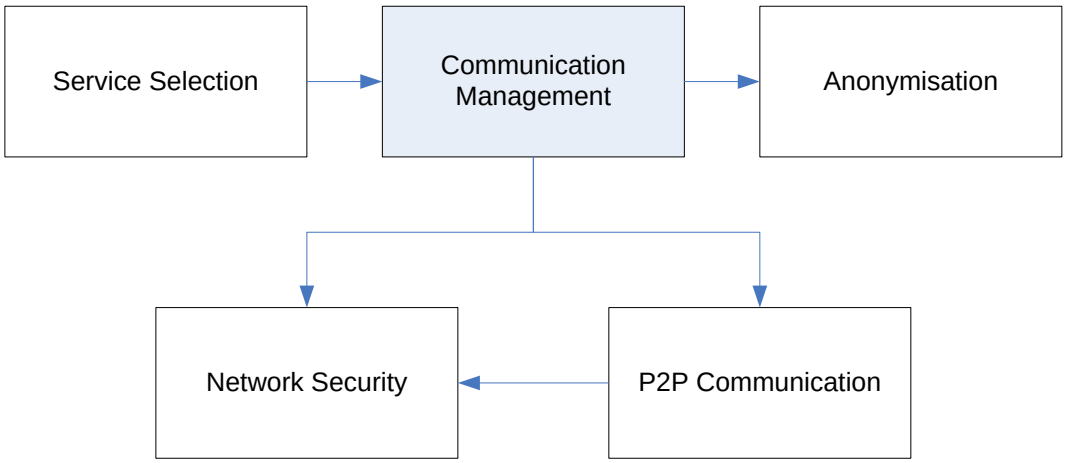


Figure 17 – Communication Management component

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

9.4.4 Network Security

History: *Component contributor: HPF, UMA*

PICOS Principle (PP): 1, 4, 7, 9, 10, 11, 13, 20

PICOS Feature (PF): 11

9.4.4.1 Purpose

The *Network Security* component is responsible for creating a secure channel between communicating entities. Security includes authenticating end points, choosing appropriate security mechanisms and handling support functions, e.g. key management.

9.4.4.2 Description

The common interpretation of network security is the execution and control of mechanisms designed to protect the confidentiality and integrity of information that passes over a communication network.

Security can be achieved using the application oriented layers of the ISO OSI reference model (i.e. layers 4 to 7) in a supporting protocol like TLS (layer 4) or S-HTTP (layer 5). Alternatively, security in the form of transmission security can be realised on the network layer (layer 3 of reference model) over an IPSec protocol.

The aim of network security is to ensure that messages in transmission cannot be read or altered without authorisation, e.g. by a third party or an adversary. The strength of protection is sufficient to deter the most determined adversary.

In addition to confidentiality and integrity of data, network security can also provide authentication of communicating parties and authenticity of data. Privacy is an increasing issue which network security can address, by obscuring identities (originator/recipient identities). Network security can also offer end-to-end confidentiality using encryption.

Network security is typically initiated by the originator of the communication, in relation to the intended target. The originator can be a member using a client device, or it can be a centralised service that wishes to communicate with another service, another community or an individual member. In addition, communication may be directly between members, who may involve peer-to-peer (P2P) network security, or perhaps more realistically, with current network topologies, a secure communication channel orchestrated by a centralised service (the spoke-and-hub communication model).

In a mobile scenario, where a secure authenticated channel is required between client and server, TLS is a possible though not necessarily efficient option[10], providing member-to-member (pseudo P2P) protection.

[10] TLS has the problem that compression is inefficient, which may be an important disadvantage for the mobile scenario

The role of the *Network Security* component, which in practice may be distributed at several 'control points', is to implement the mechanisms that provide confidentiality, integrity and authentication at the data transfer layer. For key management and other cryptographic services, the *Network Security* component will call on the services of the *Cryptography / Key Management* component. The role of the *Network Security* component may be extended to anonymisation where, with the help of the *Anonymisation* component, originator/recipient identities can be obscured. An example of how this can be achieved is the TOR[11] anonymisation technique.

A further function of the *Network Security* component concerns traffic analysis. While encryption protects the content of a message, routing and other message characteristics can reveal sensitive information. Where possible, the *Network Security* component will ensure that no information leakage is possible.

[11] TOR is a network of virtual tunnels that allows people and groups to improve their privacy and security on the Internet

9.4.4.3 Dependencies

Components that this component calls	Purpose
Cryptography / Key Management	For cryptographic mechanism and key management services in support of the request to secure a communication channel.

Components that call this component	Purpose
Communication Management	When setting up a communication channel that requires network security.
P2P Communication	When setting up a P2P communication channel that requires network security.

9.4.4.4 Drawing

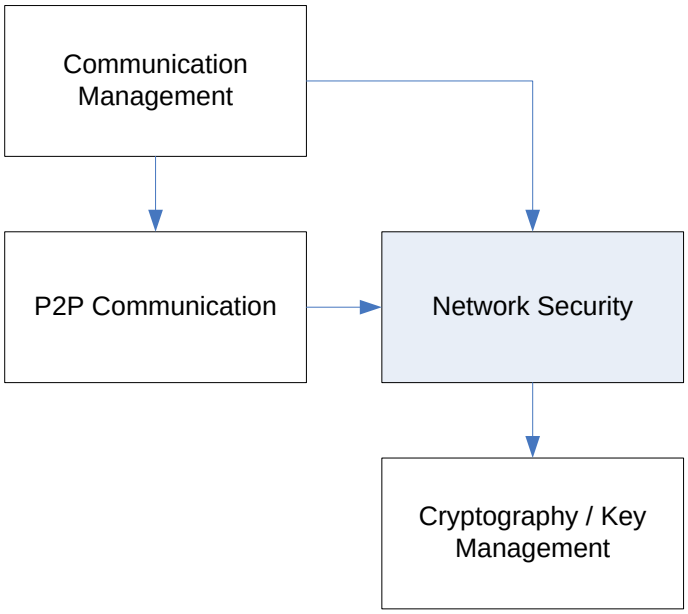


Figure 18 – Network Security

9.4.5 P2P Communication

History: *Component contributor: TMO*

PICOS Principle (PP): 1, 4, 7, 9, 10, 11, 13, 20

PICOS Feature (PF): 11

9.4.5.1 Purpose

The P2P Communication component is responsible for establishing a secure channel between two peer entities, typically two members. The objective is to provide communication when no centralised service is available (e.g. a remote location where there is no mobile network coverage) or to isolate the content of the communication from the community (members and community operator).

9.4.5.2 Description

There are basically two kinds of communication that a PICOS community might support

- between a member and a service provider (client-server (CS), or Hub-and-Spokes)
- between peers, usually members (peer-to-peer, or P2P).

The P2P topology is mainly used in offline situations where no centralised connection infrastructure is available. P2P communication helps to overcome the lack of network coverage in a mobile setting. When connection to the server is re-established, the users are synchronized again with the server. In addition, P2P communication can ease the setup of ad hoc communities. Two members can initiate a communication between their mobile devices, e.g. by using Bluetooth.

The P2P component is responsible for the setup and release of direct connections, and for the transfer of data between members. It may also have a role to play in the subsequent synchronization of off-line data with centralised resources. However, it must be recognised that one reason for using P2P (as opposed to centralised communication) is privacy, thus data shared between members should not consciously be exposed to the community. This includes routing and other message identifying information.

9.4.5.3 Dependencies

Components that this component calls	Purpose
Network Security	To establish a secure channel

Components that call this component	Purpose
Communication Management	In response to a request for an entity (member) for a direct P2P secure connection with another entity.

9.4.5.4 Drawing

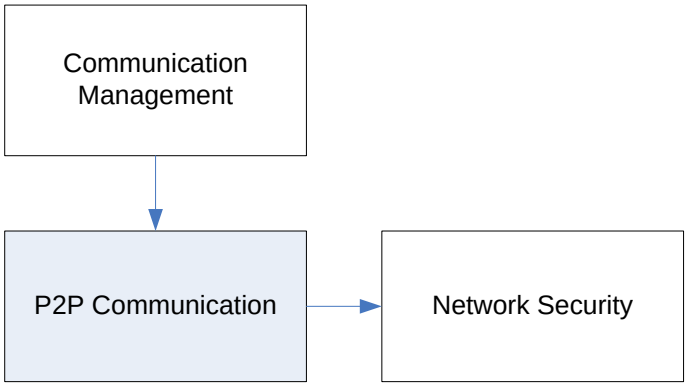


Figure 19 – P2P Communication

9.5 Services and Applications

The Services and Applications component group contains the following components

9.5.1 Tier-1 Services and Applications components

- Access Control

9.5.2 Tier-2 Services and Applications components

- Anonymisation
- Application Orchestrator
- Authentication
- Authorisation
- Date/Time Stamper
- External Recommendation
- External Service Delivery
- Feedback Management
- Identity Translator
- Importer/Exporter
- Location Sensor
- Notification
- Partial Identity Management
- Payment Services
- Preparation Area
- Privacy Advisor
- Recruitment
- Reputation Management
- Scenario Management
- Service Selection

- Social Presence
- Trust Negotiation
- TTP Management

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

9.5.3 Access control

History: *Component contributor: UMA, IfM-Geomar, HPF*

PICOS Principle (PP): 3, 17

PICOS Feature (PF): 3

9.5.3.1 Purpose

The *Access Control* component responds to a request for access to the community. Typically this will be from a member, but could also be for an external entity, e.g. a service provider.

9.5.3.2 Description

The Access Control component appears as the first point of contact for visitors to the community.

This component acts as the gatekeeper, controlling access to all community resources. It combines authentication and authorisation functionality, which are both provided as separate Tier-2 components. The *Access Control* component also handles Guest and Third Party access, and co-ordinates access from entities that claim membership of a community that has a mutual relationship with accessed community (federated access).

On receipt of a request to access the community, the *Access Control* component gathers identification and authentication information which is passed to the Authentication component for validation. If authentication is successful, the *Access Control* component forwards the access request to the *Authorisation* component where the level of access permitted is determined.

Access requests may also be received for entities other than member, or from members of other communities. In such cases, the access request is processed within the Access Control component.

For new (prospective) members, the *Access Control* component co-ordinates the registration process by directing the request to the *Registration* component.

Guest members are not required to register or be authenticated, but receive significantly reduced community functionality. Guests are processed by the Access Control component and passed directly to the Authorisation component, where limited authority is granted.

9.5.3.3 Dependencies

Components that this component calls	Purpose
Authentication	To validate the identity of an member (entity).
Authorisation	To assign access rights to the member (entity).

Components that call this component	Purpose
Communication Management	To apply access control to an incoming entity (member).

9.5.3.4 Drawing

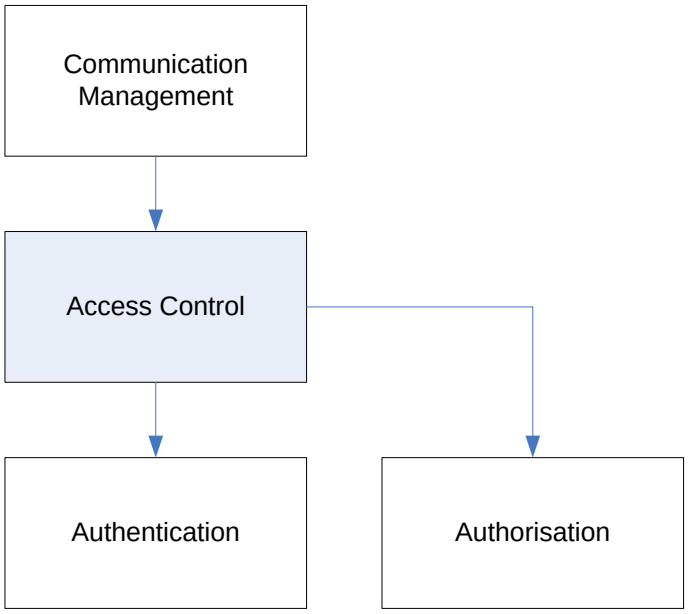


Figure 20 – Access Control

9.5.4 Anonymisation

History: *Component contributor: UMA*

PICOS Principle (PP): 1, 4, 8, 11

PICOS Feature (PF): 2, 5, 9, 13

9.5.4.1 Purpose

The *Anonymisation* component is mainly responsible for creating pseudonyms (anonymous credentials), and may have a role to play in providing or co-ordinating anonymous communication (e.g. TOR anonymous networking technique).

9.5.4.2 Description

The *Anonymisation* component provides anonymisation functionality at the application and network layers.

Application layer:

Anonymisation at the application layer allows members to create and register new pseudonyms, which can be used as partial identities. Thus, the *Partial Identity Management* component relies on the *Anonymisation* component. (Each partial identity appears to the community as a unique member with unique privileges and reputation.) Members can use pseudonyms to interact anonymously with the community, as well as to access external services anonymously.

In the situation where a member (client) operates independently of the community, the *Anonymisation* component provides members with private cryptographic keys so that they can be authenticated as the rightful holder of a pseudonym or anonymised privilege. Keys are generated by the *Cryptographic / Key Management* component.

Also at the application layer, the Anonymisation component is involved in anonymising (or pseudonymising) data that exists after a member has resigned from the community. This process is triggered by the *Revocation* component. After a period defined by community policy (see Policy component), identifying information associated with data left behind by the resigning member is first pseudonymised in a reversible way (e.g. encryption), and later in an irreversible way (e.g. hash) such that personal identifying references are totally erased.

Network layer:

The *Anonymisation* component also anonymises Internet communication endpoints (i.e. the IP address of the initiator of a transaction). The component is called when a member wishes to interact anonymously/pseudonymously, and is most likely to be used when interacting with external service providers. Anonymisation at the network layer prevents correlation between transactions and IP addresses, which could be linked to a member's identity.

This facility is not always necessary, and its usage is optional dependent on context and member preferences. By way of an example, TOR (a second generation onion routing platform) provides this facility, but requires an external TOR onion routing network and a TOR-component client.

9.5.4.3 Dependencies

Components that this component calls	Purpose
Cryptography / Key Management	To create an endorsed pseudonym.

Components that call this component	Purpose
Communication Management	To request assistance setting up network level anonymisation, e.g. a TOR solution.
Partial Identity Management	To request an endorsed pseudonym.
Revocation	To anonymise data belonging to a departing member, according to the policy of the community (Policy Management component).

9.5.4.4 Drawing

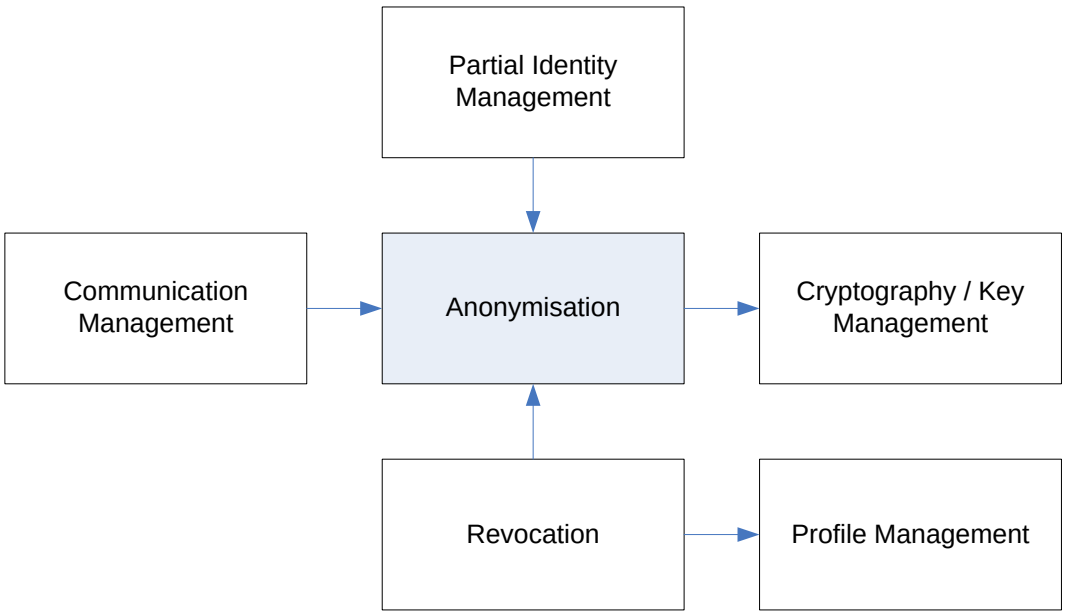


Figure 21 – Anonymisation

9.5.5 Application Orchestrator

History: *Component contributor:* ATOS

PICOS Principle (PP): 13

PICOS Feature (PF): 9, 13

9.5.5.1 Purpose

The *Application Orchestrator* component combines internal and external services to provide a richer set of functionality for members.

9.5.5.2 Description

Flexible services can help a community create privacy-respecting mash-ups (workflows) that combine information services and communication services, and content. This aggregation of services must be transparent to members. Aggregation allows the building of composite services by combining existing elementary or complex services available as part of the PICOS Toolbox or offered by external third parties. It achieves this by interacting with the External Service Delivery component.

For example, the taxi driver community describes a scenario where when picking up their child from school, parents would be happier if they could track the taxi in real-time and see an audit trail that confirms that the driver and child were in close contact at the prescribed time. Knowing that information about regular school pick-ups is protected, and being able to obtain reputation information about the driver on demand, would engender trust. In the angling community, a similar example might involve the aggregation of location, weather, a fish database and individual angling skills, to create a service that automatically plans a weekend fishing expedition.

The *Application Orchestrator* component serves two purposes:

- It provides a level of abstraction which presents aggregated services as single services, but hiding the implementation detail that is sometimes visible when interacting with multiply independent services. For example, services that may be provided as part of the PICOS Toolbox can be combined to create a richer set of functionality, while not exposing members to the fact that several services are involved. In time, this can lead to open standards.
- In a similar way, the orchestrating service provides a useful higher level of abstraction that benefits developers. Customising a community is simplified and bringing new services to members is quicker. Specific privacy preserving and trust enhancing features can easily be introduced, possibly in response to member requests provided by way of the feedback service. Greater flexibility would allow the developer to experiment and produce prototypes with ease.

9.5.5.3 Dependencies

Components that this component calls	Purpose
External Service Delivery	To access individual services.

Components that call this component	Purpose
Service Selection	To compose a set of services in response to a member request.

9.5.5.4 Drawing

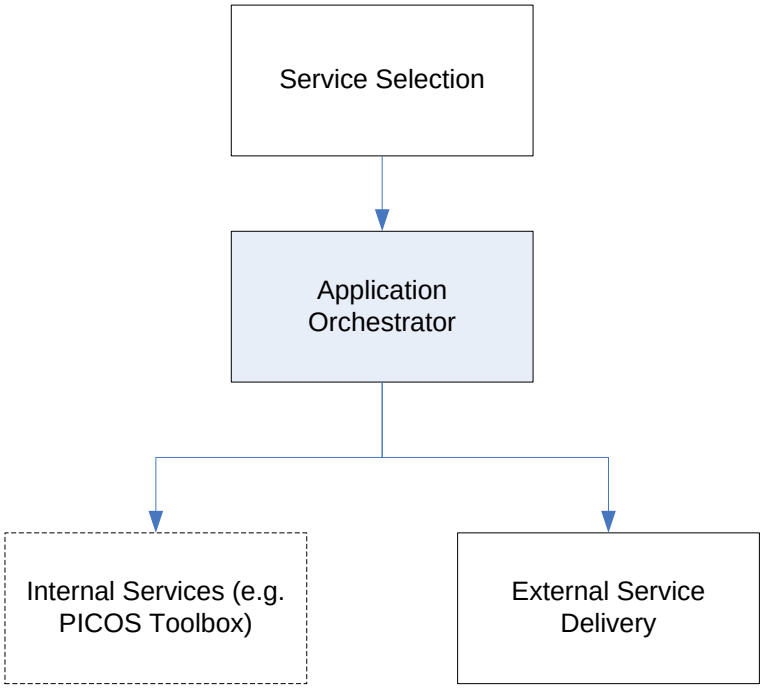
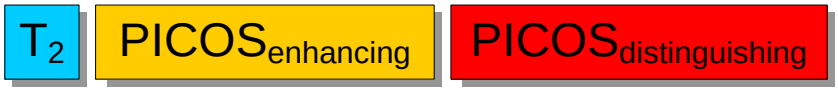


Figure 22 – Application Orchestrator

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

9.5.6 Authentication



History: *Component contributor: UMA, IfM-Geomar, HPF*

PICOS Principle (PP): 3, 17

PICOS Feature (PF): 3

9.5.6.1 Purpose

The *Authentication* component forms part of the access control process, and operates under the supervision of the *Access Control* component. The role of the *Authentication* component is to validate the information that a member provides when accessing the community and thus 'proves that they are who they say they are'.

9.5.6.2 Description

The *Authentication* component supports the community gatekeeper role, controlling access to all community resources.

All members are identified and authenticated before being granted access to the community. Identity is based of a previously registered partial identity. Authentication can be by credential, which includes an externally endorsed pseudonymous, personal identity token, platform identity biometric or traditional password. The means of authentication is set by the *Authentication Method Selection* component with respect to community Policy (determined by the *Policy* component). The *Authentication* component is supported by the *Cryptographic / Key Management* and *Secure Repository* components

After being authenticated, members are authorised to access the service to which they are entitled according to their membership privileges. Authorisation is determined by the *Authorisation* component.

Authentication information is protected during transmission between the member (client platform) and the community using an appropriate security/encryption protocol, e.g. TLS in the case of a mobile client.

Guest members are not authenticated, but only have access to a very restricted set of community services.

Third Party access, e.g. by an external service provider, would be subject to authentication unless other security checks are in place, e.g. access via a trusted channel or if subsequent checks are made on the authenticity of content submitted and shared with members.

The services of a TTP (CA) may be required in order to authenticate federated identities, accessed via the *TTP Management* component.

A useful description of how authentication is employed in PICOS can be found in Section 13 in:

- PICOS Use Case 2: Accessing the community

9.5.6.3 Dependencies

Components that this component calls	Purpose
Authentication Method Selection	To determine the set of acceptable means of authentication for the member, according to community policy (*Policy Management* component). Authentication may require the co-operation of a Trusted Third Party (TTP), e.g. where registration took place in another community (*TTP Management* component.)
Cryptography / Key Management	To support cryptographic authentication protocols.
Secure Repository	To retrieve sensitive authentication information.

Components that call this component	Purpose
Access Control	To verify the identity presented by the member. Note: The *Access Control* component subsequently calls the *Authorisation* component

9.5.6.4 Drawing

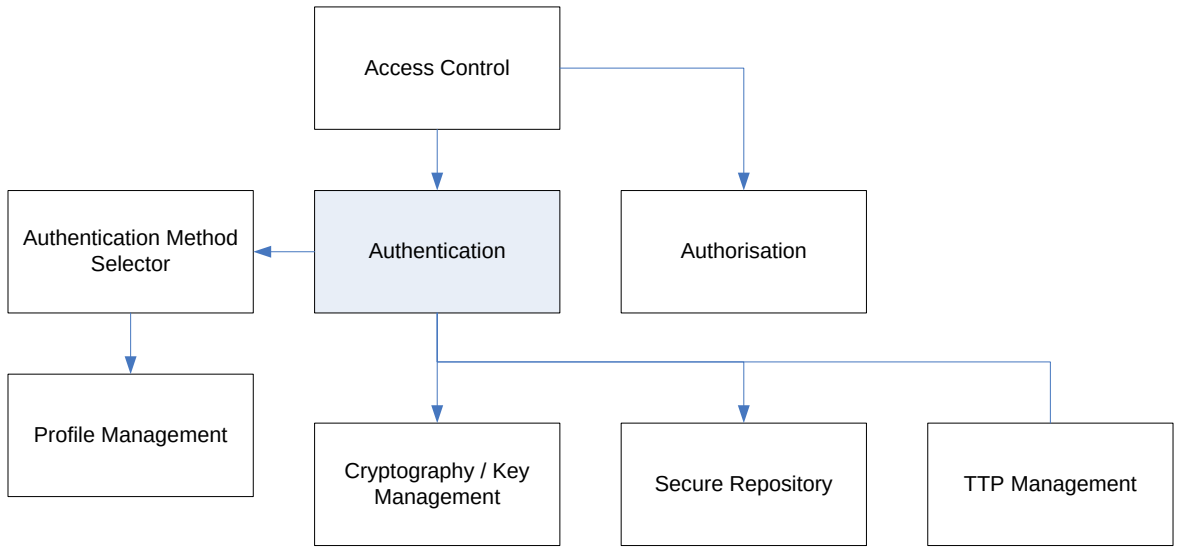
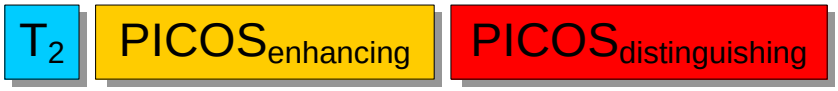


Figure 23 – Authentication

9.5.7 Authorisation



History: *Component contributor: UMA, IfM-Geomar, HPF*

PICOS Principle (PP): 3, 17

PICOS Feature (PF): 3

9.5.7.1 Purpose

The *Authorisation* component is responsible for assigning authenticated members the rights they have been granted.

9.5.7.2 Description

Members (and other entities) are authorised to access community services, subject to the following:

- Satisfying authentication requirements
- Being permitted by their profile to access the service
- Being permitted by their social presence (e.g. location) to access the service
- Receiving the consent of the service provider (particularly relevant in the case of an external service provider) or the content provider (in the case of requesting access to content provided by another entity.

The *Authorisation* component checks all of the above and, if the criteria are met, the member is allowed to proceed to the *Service Selection* component and presented with a set of available services as befits their role.

The *Authorisation* component is called whenever a member requests a service that has restricted access. For example, a member may be restricted for accessing a service for certain locations. Having accessed the community and received authorisation to import content, this privilege may subsequently be revoked if the member moves to another location (e.g. relocating from a work place to a public place). In such a case the *Service Selection* component might call the *Authentication* component before granting access to the service. A similar situation exists with reputation, which is another dynamic member attribute.

A useful description of how authorisation is employed in PICOS can be found in Section 13 in:

- PICOS Use Case 1: Registration
- PICOS Use Case 2: Accessing the community
- PICOS Use Case 4: Multiple Partial Identities

9.5.7.3 Dependencies

Components that this component calls	Purpose
Profile Management	To retrieve privileges assigned to the member.
Social Presence	To check current status of the member

Components that call this component	Purpose
Access Control	As part of the access control process governing access by members to the community. Note: Authorisation is called after calling the *Authentication* component.
Service Selection	When a member requests a service that is only available depending on current (real-time) social presence.

9.5.7.4 Drawing

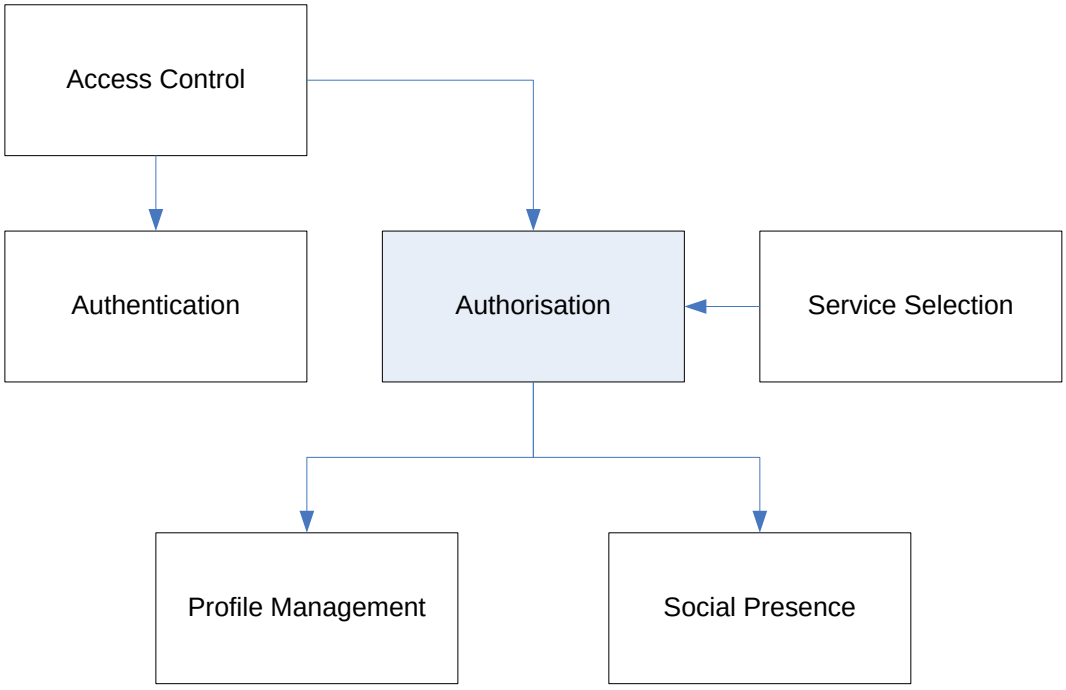


Figure 24 – Authorisation

9.5.8 Date/Time Stamper

History: *Component contributor:* IfM-Geomar

PICOS Principle (PP): 12, 23

PICOS Feature (PF): 1, 2, 10, 15

9.5.8.1 Purpose

The *Date/Time Stamper* provides an accurate and reliable date/time reference.

9.5.8.2 Description

Text, audio, picture and video in the online world are in digital form and easily modifiable. This gives rise to questions of how best to check when a document was created or last modified. Cryptographic processes, e.g. hash, MAC and digital signature functions allow changes to be detected, but they do not reveal the time of modification. For example, with intellectual property matters it is sometimes crucial to verify the date that the inventor first recorded the patentable idea, in order to establish its precedence over competing claims. PICOS can digitally time-stamp any documents so that it is infeasible for a date to be backdated or forward-dated.

Date and Time stamping must satisfy two fundamental requirements:

- It must be infeasible to timestamp a document with a date and time different from the present one
- It must be infeasible to change even a single bit of a time-stamped document without the change being apparent.

Many other components might find the Date/Time Stamper component useful, e.g. Feedback component, Event Logging component, Content Sharing (Importer/Exporter) component.

The Date/Time Stamper component adds a time-stamp to a document. It requires access to a stable date/time reference.

Example: RFC 3161: Time Stamping Protocol defines the entities involved (Requestor / Client and Time Stamp Authority (TSA) / Server), the message format and the transport protocol which permits communication between the entities. The following figures show the ‘time-stamp request’ and the ‘time-stamp verify’ phases.

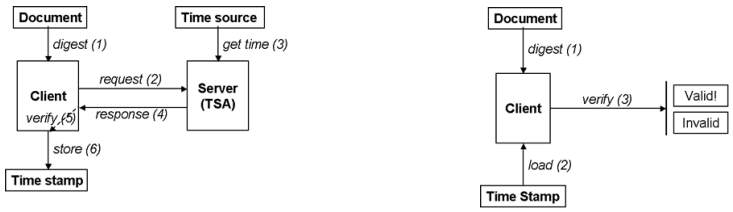


Figure 25 – Example Time/Stamp protocol

9.5.8.3 Dependencies

Components that this component calls	Purpose
None defined at present	

Components that call this component	Purpose
Content Sharing	To record when content is imported and shared.
Event Logging	To record when events are written to the event log.
Feedback	To record when feedback was provided.

9.5.8.4 Drawing

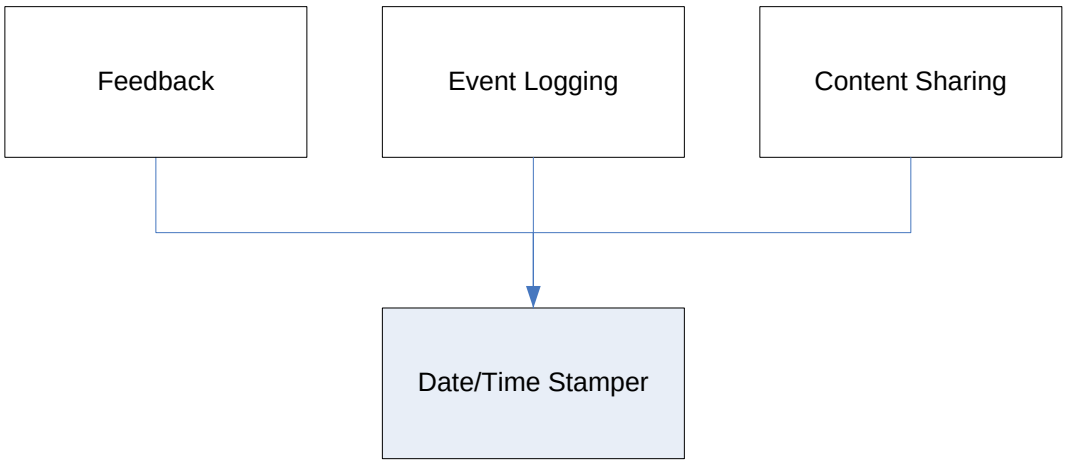


Figure 26 – Date/Time Stamper

9.5.9 External Recommendation

History: *Component contributor: HPF*

PICOS Principle (PP): 6, 16, 22, 23

PICOS Feature (PF): 1

9.5.9.1 Purpose

The *External Recommendation* component acts as a gateway for recommendations that come from other communities or from external / non-trusted sources.

9.5.9.2 Description

The *External Recommendation* component establishes a common language or common ranking system that allows external recommendations to be interpreted in a common way. It would most probably be called by the *External Service Delivery* component. It allows internal and external recommendations (probably just reputation to begin with) to be compared on the same scale.

In the case of external reputation, this is managed in the same way as for a member by the *Reputation Management* component. It will be anonymised and recorded against the partial identity of the external entity using the *Profile Management* component.

9.5.9.3 Dependencies

Components that this component calls	Purpose
Profile Management	To record the reputation of the external entity.
Reputation Management	For reputation management service as used with members.

Components that call this component	Purpose
External Service Delivery	To check recommendations on external services.

9.5.9.4 Drawing

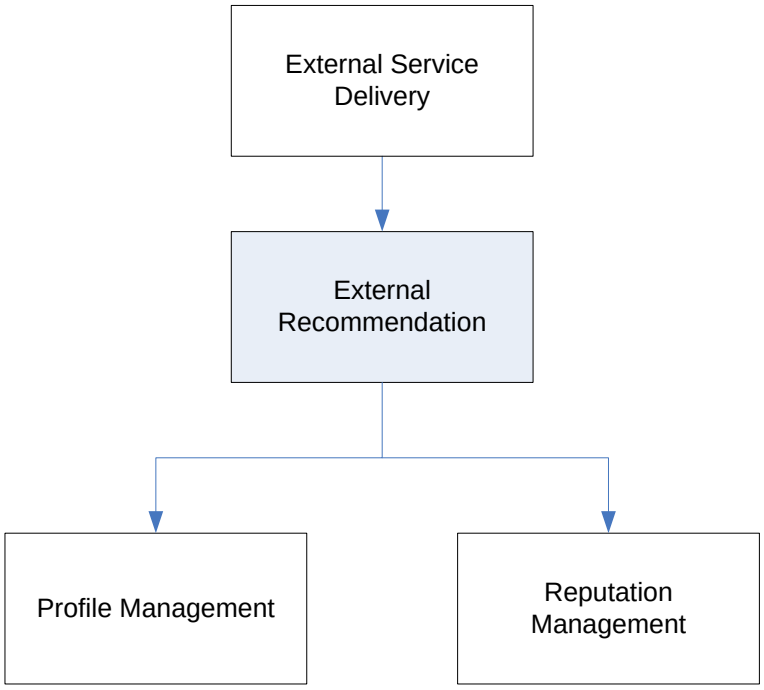


Figure 27 – External Recommendation

9.5.10 External Service Delivery

History: *Component contributor: HPL, HPF, ITO*

PICOS Principle (PP): 6, 13, 17, 22

PICOS Feature (PF): 10

9.5.10.1 Purpose

The *External Service Delivery* component is responsible for ensuring that external service is delivered according to the level and quality of service previously defined and agreed with community operator/members. This component can also aggregate service to provide richer services to members.

9.5.10.2 Description

The *External Service Delivery* manages the interaction with external service providers. It controls how members access external services and limits the amount of member personal information using the *Data Minimisation* component. It also controls the delivery of content and notifications from the service provider to community members using the *Content Sharing* component.

Service aggregation can cover both the aggregation of the content the aggregation of services. At the content level, aggregation allows members to merge documents and live feeds to provide a common source of information. At the service level, aggregation takes internal and external service, ranging from full-scale applications to simple functions (code fragments), that can be combined into larger services. Web-based aggregation is also a possibility, e.g. Google Reader.

To enhance privacy, external service can be accessed anonymously using a partial identity specifically created by the *Partial Identity Manager* component.

A useful description of how external service delivery is employed in PICOS can be found in Section 13 in:

- PICOS Use Case 6: External services

9.5.10.3Dependencies

Components that this component calls	Purpose
Content Sharing	To allow content received from any external service provider to be shared with other members.
Data Minimisation	To reduce the information that a member shares with an external service provider.
Partial Identity Management	To anonymise the identity of the member accessing the external service.

Components that call this component	Purpose
Service Selector	By members requesting access to an external service.

9.5.10.4Drawing

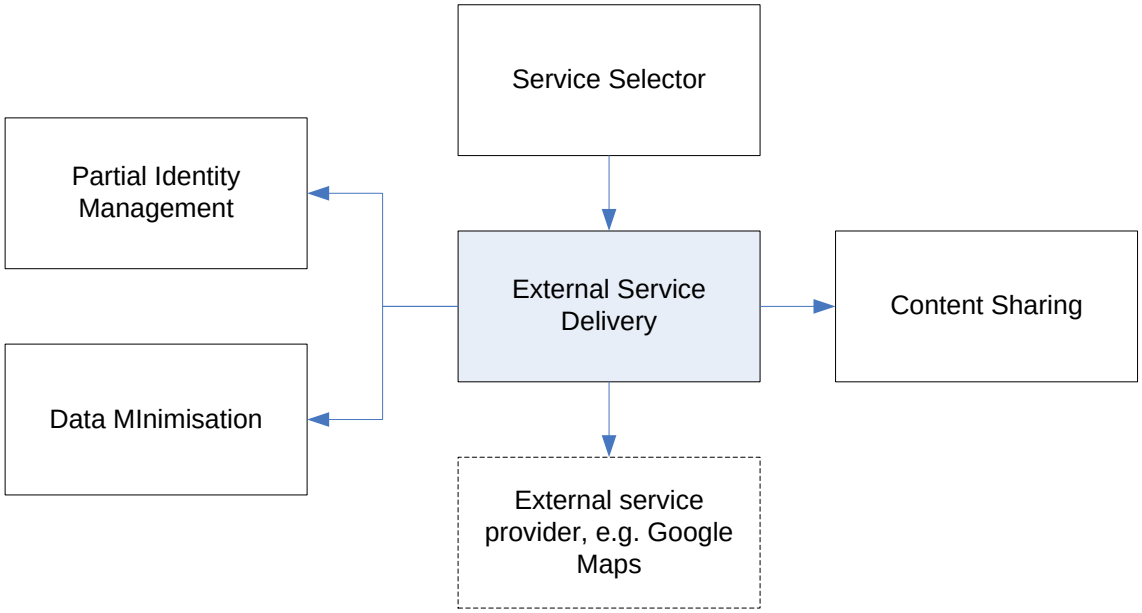


Figure 28 – External Service Delivery

9.5.11 Feedback Management

History: *Component contributor:* TMO, UMA

PICOS Principle (PP): 5, 16, 23

PICOS Feature (PF): 1

9.5.11.1 Purpose

The *Feedback Management* component provides a route for members to supply feedback to the community.

9.5.11.2 Description

Feedback from members is vital in online communities, whether the community is provided for professional or leisure-time activities. Creative contribution from the membership is vital to the growth and success of any community, large or small. The more members are engaged (i.e. providing feedback) with the community, the stronger the community becomes. Members are also a valuable source for ideas. Capturing member feedback can lead to new, innovative community services.

The *Feedback Management* component is a service that:

- Collects feedback from contributors
- Creates and facilitates fellowship and one-to-one communication among community members
- Shares feedback and information provided by other members, and report on progress implementing new features which resulted from member user suggestions
- Allows customisation/innovations to meet the needs of community members
- Provides input into the reputation system, so that members who support the community through action or contribution can be rewarded with positive remarks

Feedback can take various forms, from simple 5-star ratings, karma rating through to personal recommendation. Feedback in this form is also referred to as reputation. Feedback is especially helpful in large communities where feedback can enhance trust in other members and the community as a whole. Reputation is automatically adjusted when feedback is received and can lead to increased privileges for the member concerned.

Privacy is always a concern, so the *Feedback Management* component restricts feedback to specified sub-communities or ensures that feedback is anonymous but accountable (subject to control to filter inappropriate feedback). Feedback is tagged to indicate the partial identity of member who provided the feedback.

The feedback process should ideally be self-managing, or managed by a ‘leader’ elected by the community (and the election could be based on feedback or reputation). Centralised monitoring of

content, the censorship of entries and control over the focus and activities of sub-communities should be avoided.

9.5.11.3Dependencies

Components that this component calls	Purpose
Reputation Management	To record reputation information.

Components that call this component	Purpose
Service Selection	To allow members to provide feedback.

9.5.11.4Drawing

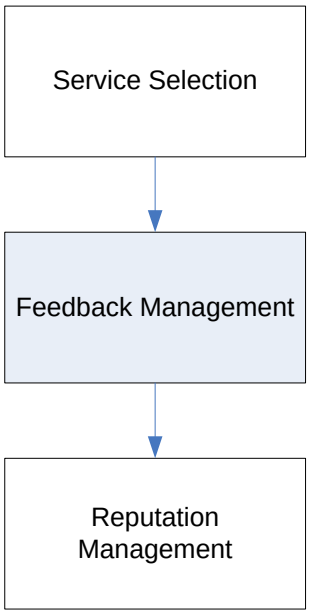


Figure 29 – Feedback Management

9.5.12 Identity Translator

History: *Component contributor:* BRNO

PICOS Principle (PP): 6, 13, 18

PICOS Feature (PF): 9, 13

9.5.12.1 Purpose

The *Identity Translator* component is an extension to the Identity Management component which deals with special situation concerning external identities.

9.5.12.2 Description

The *Identity Translator* component is an extension, i.e. a separate service, to the *Identity Management* component. Its purpose is to 'reconfigure' an identity to take account of a special situation surrounding the access to a service or external community, using the *External Service Delivery* component. For example, a member may have a preference set that states that their identity must be anonymised or reduced in 'richness' (i.e. some personal information removed) when they interact with any part of the community outside of their designated sub-community. The reason for this is that they do not want to excessively expose personal information in an environment which they do not consider trustworthy.

The *Identity Translator* component would most likely call on the Anonymisation component or the *Partial Identity Management* component to perform this request. Exactly what action the *Identity Translator* takes will depend on the policy set by the member and the community, the context of the situation and possibly other factors like member and community(ies) reputation.

Another reason for invoking the *Identity Translator* component is because part of the community cannot support the format of the identity used elsewhere. This may be true with a legacy system or possible a mobile client which has limited functionality. It may also be required during ad hoc interaction with other communities (possibly as a guest member).

9.5.12.3Dependencies

Components that this component calls	Purpose
Data Minimisation	To reduce the 'richness' on personal data sent to the service provider.
Partial Identity Management	To create pseudonyms as alternative to personal identifiers, which should be less revealing of personal information. Note: the *Partial Identity Management* component may utilise the *Anonymisation* component.

Components that call this component	Purpose
External Service Delivery	To access an external service.

9.5.12.4Drawing

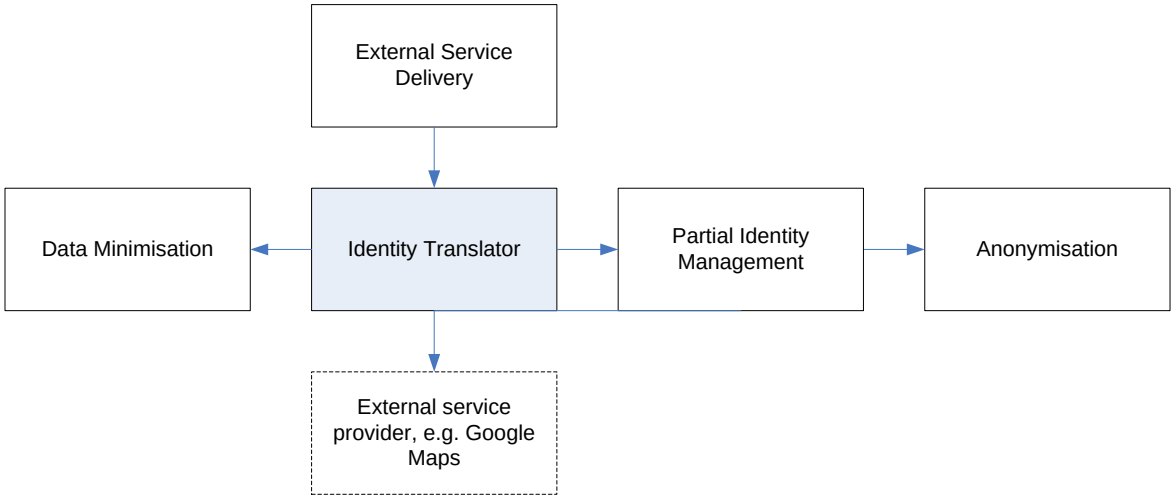


Figure 30 – Identity Translator

9.5.13 Importer/Exporter

History: *Component contributor: GUF*

PICOS Principle (PP): 2, 3, 4

PICOS Feature (PF): 2, 5, 6, 10, 12

9.5.13.1 Purpose

The *Importer/Exporter* component is responsible for (mainly) allowing members to upload/download content.

9.5.13.2 Description

The *Importer/Exporter* component represents the interface for data exchange between the community and member. It provides for the synchronisation and backup of personal data, and the up-/downloading of content. By uploading contact data from an address book, e.g. Microsoft Outlook, it is possible to expand the PICOS address book or PICOS buddy list to include members from the mobile community. Additionally, an interface is available which allows the exchange of data with other communities, e.g. GoogleMail, Facebook, etc.

This *Importer/Exporter* component also supports the import/export of media (e.g. picture, video and sound files) and document (e.g. Microsoft Word documents, pdf files, etc.), so that content can be shared between members. Furthermore, the *Importer/Exporter* component supports the import/export of personal data from a mobile client, including address data and security settings. The component supports backup and migration of client devices.

Imported data may be manually or automatically 'tagged' with meta-data, and transferred to a predefined location for sharing with other members of the community, subject to access restriction being satisfied.

The *Importer/Exporter* makes extensive use of the *Content Sharing* component.

9.5.13.3Dependencies

Components that this component calls	Purpose
Content Sharing	To 'tag' content, make content available to (share with) other members and to apply restriction on access to the content.

Components that call this component	Purpose
Service Selection	In response to a member request to import/export content.

9.5.13.4Drawing

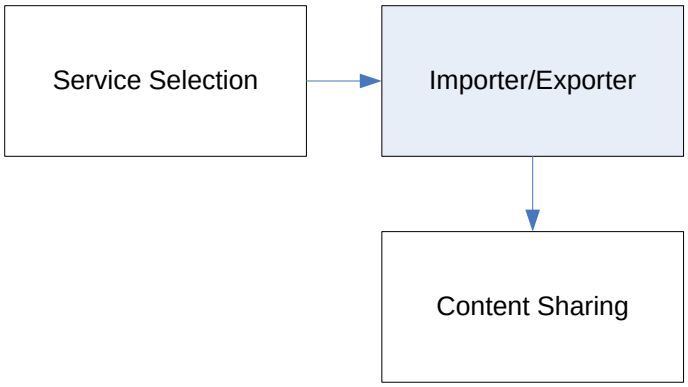


Figure 31 – Importer/Exporter

9.5.14 Location Sensor

History: *Component contributor: TMO*

PICOS Principle (PP): 10

PICOS Feature (PF): 4, 8

9.5.14.1 Purpose

The *Location Sensor* reports the current location of the member.

9.5.14.2 Description

The *Location Sensor* component provides an interface to retrieve the current location of a member. The location can either be determined by the member (client) device, e.g. using a GPS receiver, or by the network, e.g. cell-based location.

9.5.14.3Dependencies

Components that this component calls	Purpose
None defined at present	

Components that call this component	Purpose
Access Control	As part of authorisation during access control.
Authorisation	As part of authorisation during service selection.

9.5.14.4Drawing

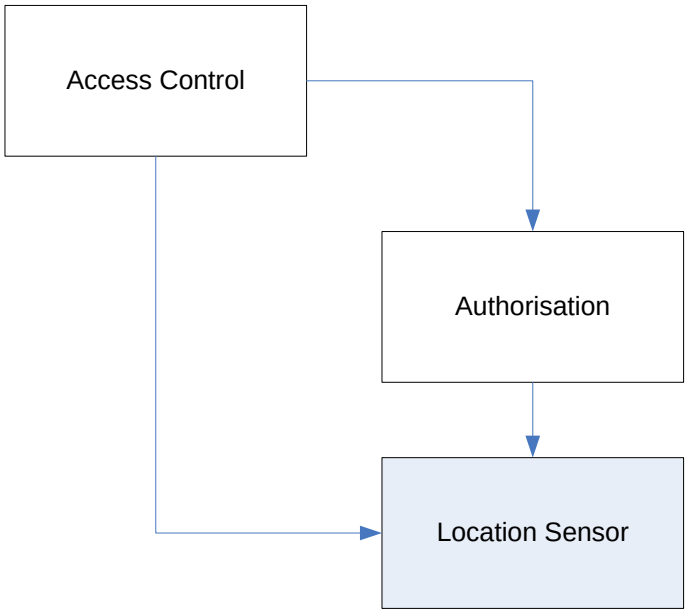


Figure 32 – Location Sensor

9.5.15 Notification

History: *Component contributor:* All

PICOS Principle (PP):

PICOS Feature (PF):

9.5.15.1 Purpose

The *Notification* component communicates with members in response to a member or community initiated event.

9.5.15.2 Description

For a variety of reasons, members or the community operator need to notify other members that something within the community has changed, or that an action must be taken. For example, when a member contributes content to the community they will want to notify all members who are permitted to see the content that the content is available. The *Social Presence* component may also need to call the *Notification* component to alert selected members to a change in status of another member.

9.5.15.3Dependencies

Components that this component calls	Purpose
None defined at present	Notification is currently considered to be an internal function of the community messaging system.

Components that call this component	Purpose
Social Presence	To alert a change in social presence, e.g. location.
Content Sharing	To alert members of an import of content that they are entitled to access.

9.5.15.4Drawing

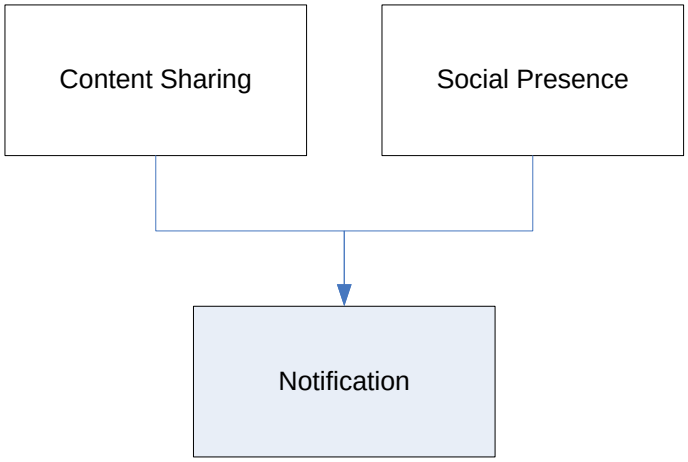


Figure 33 – Notification

9.5.16 Partial Identity Management

History: *Component contributor:* HPL

PICOS Principle (PP): 11, 18

PICOS Feature (PF): 1, 3

9.5.16.1 Purpose

The *Partial Identity Management* component creates partial identities that enable members to interact with the community.

9.5.16.2 Description

The role of the *Partial Identity Management* component is to enable members to utilise one or more partial identities as they interact with other community members. A partial identity is an identity that includes some but not all personal attributes. For example, a partial identity may consist of a name and telephone number, or more likely will be a pseudonym. The latter has the advantage of affording greater privacy.

For reasons of accountability and community management, it may be necessary to link all partial identities that relate to a single individual under a common identity, in PICOS called the root identity. The ability to link partial identities for an individual would be restricted, either to the community operator or an external trusted intermediary (or a law enforcement authority).

Every member has at least one partial identity, which is created when they register with the community and subsequently when they request for additional partial identities. The reason for requesting additional partial identities is so that members can interact with the community in multiple ways.

Every partial identity has a profile, preferences and a reputation, and to other members appears like a unique member.

Partial identities provide members with access to the community and community services.

A useful description of how partial identifiers are employed in PICOS can be found in Section 13 in:

- PICOS Use Case 1: Registration
- PICOS Use Case 2: Accessing the community
- PICOS Use Case 4: Multiple Partial Identities
- PICOS Use Case 5: Reputation

9.5.16.3Dependencies

Components that this component calls	Purpose
Anonymisation	To create the partial identity (a pseudonym) endorsed by the community.
Profile Management	To assign the partial identity a profile.

Components that call this component	Purpose
Registration	When registering as a member of a community, an initial partial identity is automatically created.
Service Selection	When a member wants to create an additional partial identity.

9.5.16.4Drawing

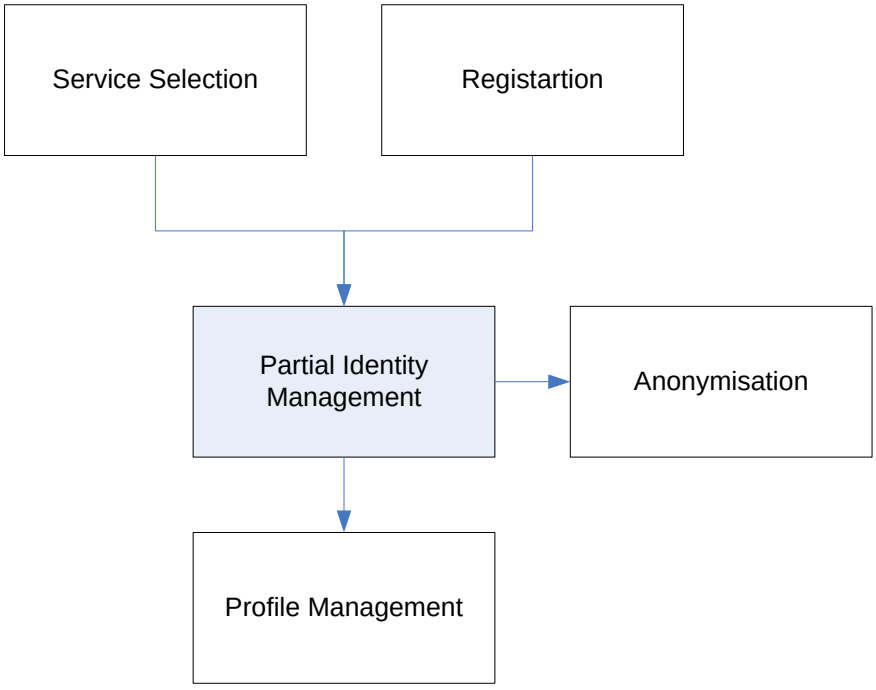


Figure 34 – Partial Identity Management

9.5.17 Payment Services

History: *Component contributor: HPL*

PICOS Principle (PP): 3, 4, 9, 13

PICOS Feature (PF): 9

9.5.17.1 Purpose

The *Payment Services* component provides access to external payment service, e.g. Visa, MasterCard, PayPal. It is specifically included in the PICOS architecture because PICOS addresses privacy issues that arise through advertising. In all other respects, the *Payment Services* component is similar to other externally provided services.

9.5.17.2 Description

The *Payment Services* component enables members to purchase services offered by the community operator, or offered by an external service provider that has advertised services to the community. Such a service is accessible the *External Service Delivery* component.

Several payment methods should be catered for, e.g. Visa, MasterCard, PayPal, thus the payment service is essentially outsourced. The community operator ensures a common user experience and integration between the community and the supplier of the service being purchased.

Payment problems are resolved between the member(s) concerned and the external payment service provider.

9.5.17.3Dependencies

Components that this component calls	Purpose
External Payment Delivery	To gain access to the payment service provider's system.

Components that call this component	Purpose
Service Selection	When a member wants to make a payment for a service (possibly including and external service).

9.5.17.4Drawing

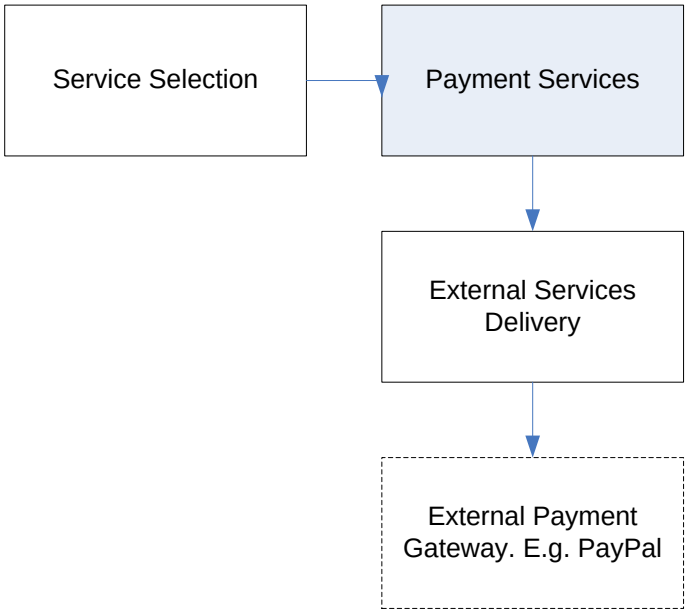


Figure 35 – Payment Services

9.5.18 Preparation Area

History: *Component contributor:* HPF, IfM-Geomar

PICOS Principle (PP): 5, 10

PICOS Feature (PF): 14

9.5.18.1 Purpose

The *Preparation Area* component provides a secure area for members to create and manage content before sharing with the community.

9.5.18.2 Description

The *Preparation Area* component, also referred to as personal space, provides an area where members can experiment, get used to the community and experience what the community has to offer without any personal risk. It enables members to build trust in the community and its services.

The preparation area is presented as a service, and may be available to Guest members if the community policy permits.

A parallel can be drawn with Web 2.0. Web 2.0 is designed to allow members to easily establish personal workspaces on the Internet. The same principle applies to a PICOS community. In Web 2.0, aspects of personal and public spaces are closely intertwined. Members are able to choose for themselves, and effectively trade privacy for greater social interaction.

Only the owner of the personal space has access; it is not visible to any other community member or operator. It is not audited and no history is maintained. However, it is possible for members to transfer personal information from the preparation area into the main body of the PICOS community, at which point the information is managed using the controls that PICOS provides for the community.

The functionality that supports the preparation area can be provided locally, on the client platform, or centrally by the community. The latter is potentially less secure but more convenient for client platforms with limited capabilities.

9.5.18.3Dependencies

Components that this component calls	Purpose
None defined at present	The preparation area can be considered as a duplicate PICOS community, possibly provided in a trusted location or by a Trusted Third Party (TTP).

Components that call this component	Purpose
Service Selection	To allow a member to experiment with PICOS functionality is a safe situation.

9.5.18.4Drawing

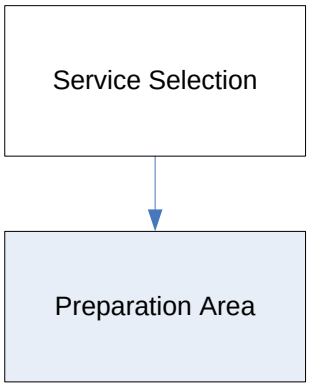


Figure 36 – Preparation Area

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

9.5.19 Privacy Advisor

History: *Component contributor:* HPL

PICOS Principle (PP): 3, 4, 8

PICOS Feature (PF): 2, 10

9.5.19.1 Purpose

The *Privacy Advisor* informs members if the action they are about to perform will place their privacy at risk.

9.5.19.2 Description

The *Privacy Advisor* is perhaps best thought of as a member's best buddy. It is designed to provide guidance of privacy related matters that may affect a member as they interact with the community. Privacy (and trust) is subjective, and it is often difficult to find a single 'right answer' to questions and concerns about privacy. One of the challenge is one of understanding what information a member values most. The role of the *Privacy Advisor* component is to present facts about the community that have a bearing on privacy. Along with the member's privacy preferences and profile, it should be possible to offer advice to the member. Therefore, *Profile Management* and *Reputation Management* components are likely to be involved.

This is probably best described as an 'advanced component', where further research is necessary. For this first version of the architecture, it is important to create a 'place holder' for this type of functionality, and to offer a simplified service consisting of perhaps general community information (number of member, recent activity) and events that relate directly to information that a member has contributed to the community for the benefit of others.

The *Privacy Advisor* component may be activated for a variety of reasons, e.g. by the *Service Selection* component, *External Service Delivery* component and *Scenario Management* component. It may also play an important role in negotiating trust, i.e. the *Trust Negotiation* component.

9.5.19.3Dependencies

Components that this component calls	Purpose
Profile Management	To obtain information about the entity involved.
Reputation Management	To obtain information about the entity involved.

Components that call this component	Purpose
External Service Delivery	To advise the member on the action that they are about to perform.
Scenario Management	To provide context information to help members make decisions about the exposure of an identity (or partial identity), the sharing of information and the use of community services.
Service Selection	To advise the member on the action that they are about to perform.
Trust Negotiation	To advise the member on the action that they are about to perform.

9.5.19.4Drawing

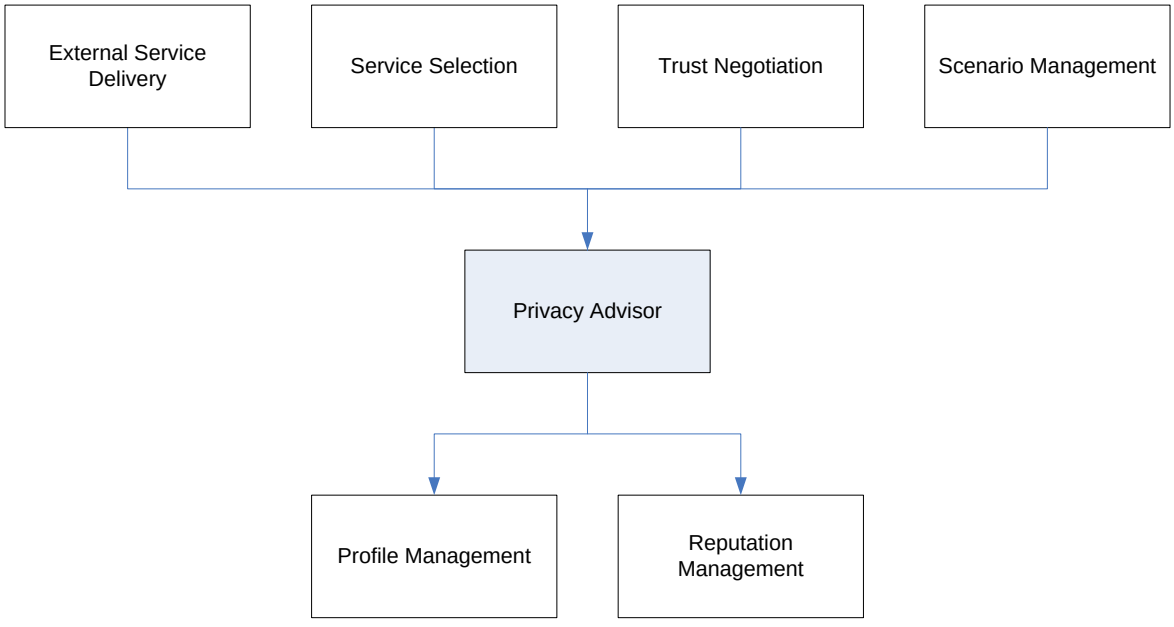


Figure 37 – Privacy Advisor

9.5.20 Recruitment

History: *Component contributor:* HPF

PICOS Principle (PP): 21, 23

PICOS Feature (PF): 1

9.5.20.1 Purpose

The *Recruitment* component provides a way for existing members to recommend prospective members for membership of the community.

9.5.20.2 Description

The *Recruitment* component provides functions to enlist new members, based on recommendations from existing members and reputation. Another source of recommendation is a Trusted Third Party (TTP) or intermediary, via the *TTP Management* component, which would vouch for the prospective member. Thirdly, the recommendation may come from another community, perhaps via the *External Recommendation* component.

One of the criterion that the Authentication component accepts as evidence is a completed application form. An option is for this form to be endorsed (similar to a sponsor) by and existing member. The *Recruitment* component could support this process.

9.5.20.3Dependencies

Components that this component calls	Purpose
Authentication	To support a prospective member's application form.

Components that call this component	Purpose
External Recommendation	To recommend a prospective member for membership.
TTP Management	To recommend a prospective member for membership.

9.5.20.4Drawing

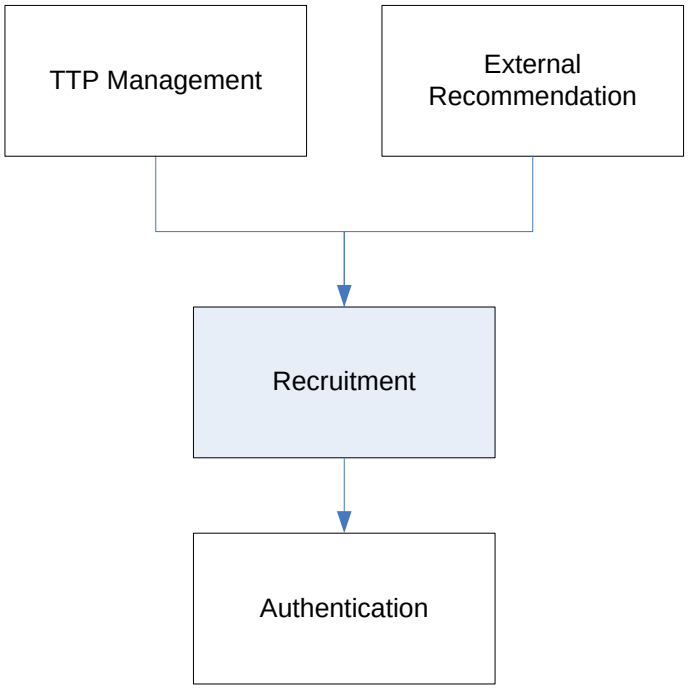


Figure 38 – Recruitment

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

9.5.21 Reputation Management

History: *Component contributor:* TMO, UMA

PICOS Principle (PP): 6, 22, 23

PICOS Feature (PF): 1

9.5.21.1Purpose

The *Reputation Management* component is used to provide an indication of the trustworthiness of an entity (typically a member).

9.5.21.2Description

Reputation is an important mechanism for building trust between community members, and forms the basis for making recommendations. Reputation is based on member performance and typically derived from feedback and recommendation from other members. Recommendations are transitive, in that member A recommends member B to member C, but member C has no firsthand experience of member A. (Trust is often said to be transitive too, i.e. A trusts B, and B trusts C, therefore A trusts C). From this principle, a hierarchy of member recommendations can be created and maintained as a basis for trust between the members.

The *Reputation Management* component is responsible for handling reputation received from members. The exact process requires further research, but one possibility is for the *Reputation Management* component to maintain a recommendation graph, which represents links between members. Reputations can be added/removed from the graph. In order to build trust it may be necessary to maintain a history showing the 'lifetime of a reputation', so that members can observe how it has evolved.

Reputation is not only concerned with the reputation of other members. It is equally concerned with reputation of subjects/topics/items/activities, in fact anything relevant to the community. For example, in the angling community reputation might include fishing location, tackle, bait, conditions and external angling services.

Reputation information is stored in the profile of the entity (member) to which it relates, using the *Profile Management* component. It may be requested by various components, but in particular the *External Recommendation*, *Privacy Advisor* and *Trust Negotiation* components.

9.5.21.3Dependencies

Components that this component calls	Purpose
Profile Management	To record reputation information on the entity concerned.

Components that call this component	Purpose
External Recommendation	To obtain reputation information on the entity concerned.
Privacy Advisor	To obtain reputation information on the entity concerned.
Trust Negotiation	To obtain reputation information on the entity concerned.

9.5.21.4Drawing

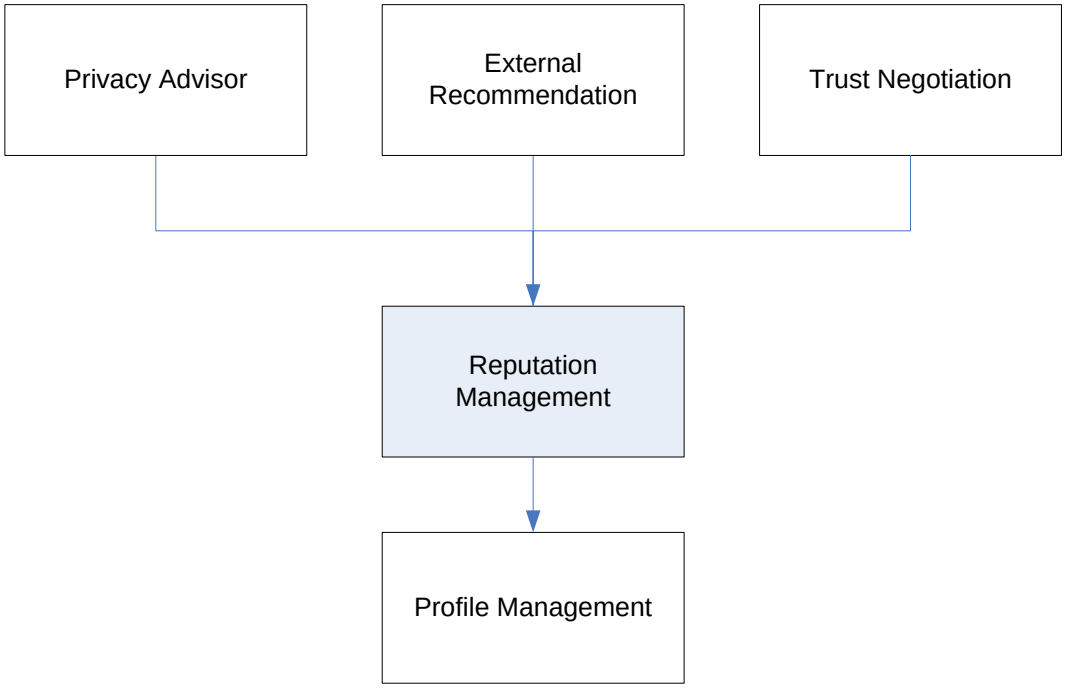


Figure 39 – Reputation Management

9.5.22 Scenario Management

History: *Component contributor:* ATOS

PICOS Principle (PP): 8, 11, 23

PICOS Feature (PF): 2

9.5.22.1Purpose

The *Scenario Management* component provides context information to members, sufficient to help them make decisions about the exposure of an identity (or partial identity), the sharing of information and the use of community services.

9.5.22.2Description

The *Scenario Management* component determines the current context and assesses its relevance to maintaining privacy. It also assesses the impact of a changing context and the implication of trust policies intended to protect sensitive information.

In an online community it is often difficult to determine context (i.e. determine or understand a scenario), since members are not aware of other members, services or third parties who might be observing or collecting data, e.g. to analyse virtual behaviour. Therefore, mechanisms are required that detect and communicate something about the environment in which members operate.

There are (at least) four metrics that can be observed in a typical scenario, and which could be used to indicate risk (and therefore impact on trust and privacy):

- The complexity of the relationship between members (including members of other communities), and the use of services (especially third party provided services) can suggest the level of control, or the ability to enforce personal privacy policies
- The sensitivity of the data being shared or processed is another indicator. The more sensitive and extensive the data, the greater the impact of exposure and the need for tighter control.
- A third metric is the reputation of other members involved in the scenario. The strength of authentication may also have a bearing on the trustworthiness on those involved in the scenario. This can be extended to the reputation of services that support data processing and sharing.
- A fourth metric is the communication medium over which information is shared.

9.5.22.3*Dependencies*

Components that this component calls	Purpose
Privacy Advisor	To provide input to the privacy advice process.

Components that call this component	Purpose
External Service Delivery	To advise the member on the action that they are about to perform.
Service Selection	To advise the member on the action that they are about to perform.
Trust Negotiation	To advise the member on the action that they are about to perform.

9.5.22.4Drawing

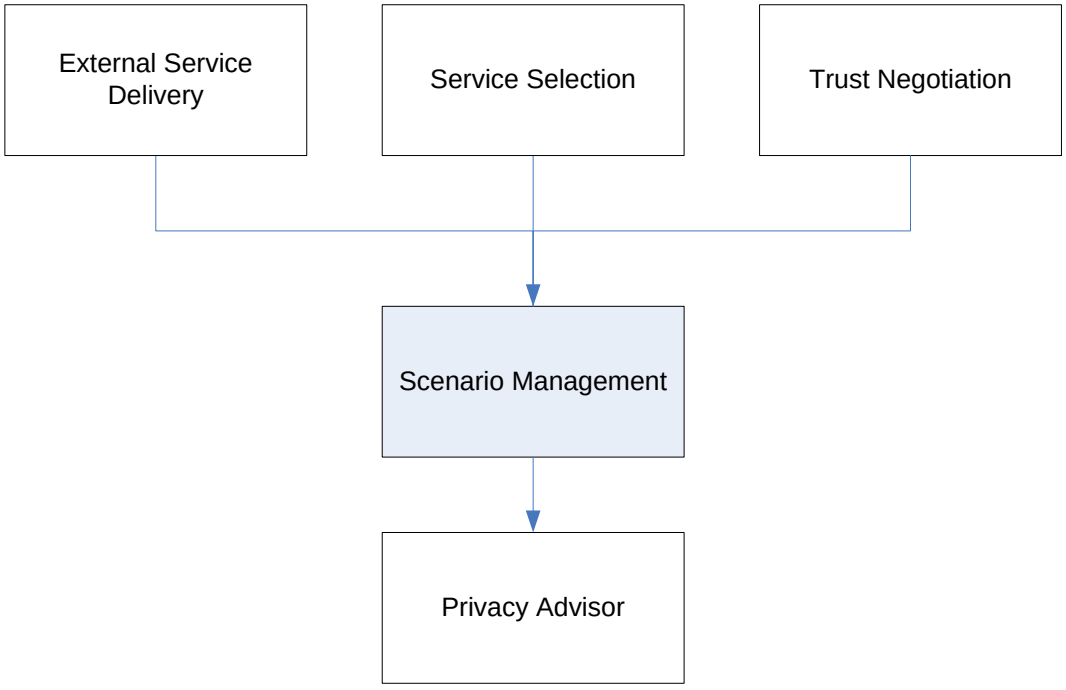


Figure 40 – Scenario Management

9.5.23 Service Selection

History: *Component contributor:* All

PICOS Principle (PP):

PICOS Feature (PF):

9.5.23.1Purpose

The *Service Selection* component presents the available service to the member.

9.5.23.2Description

Once a member has gained access to the community via the *Access Control* component, they are presented with the set of service that they can access according to their privileges. Privileges are set in their profile by the *Profile Management* component.

In addition, restrictions on the service available to the member may be imposed by the *Social Presence* component, and by the community policy as defined by the *Policy Management* component.

9.5.23.3Dependencies

Components that this component calls	Purpose
Policy Management	To determine the services that the member can access
Profile Management	To determine the services that the member can access.
Social Presence	To determine the services that the member can access.

Components that call this component	Purpose
Access Control	To access a service.

9.5.23.4Drawing

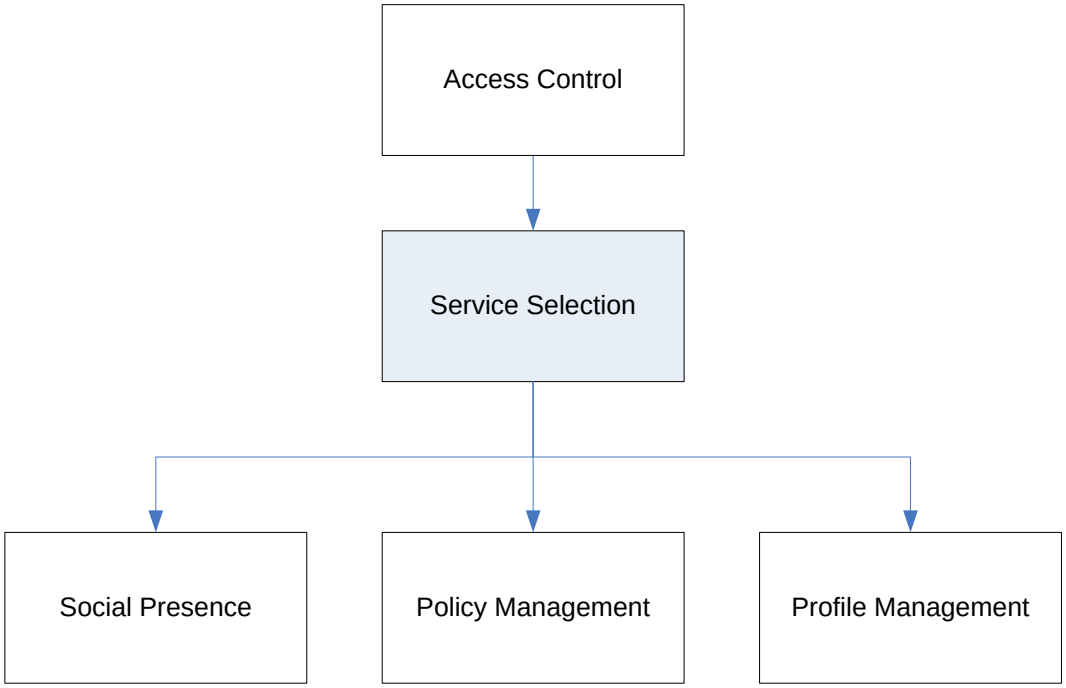


Figure 41 – Service Selection

9.5.24 Social Presence

History: *Component contributor:* ATOS

PICOS Principle (PP): 10

PICOS Feature (PF): 8

9.5.24.1Purpose

The *Social Presence* component controls the visibility of a member to other members in the community.

9.5.24.2Description

Social presence is defined as the willingness and ability of a member to communicate with other members in the community. Social presence also expresses a member's reachability and willingness to share current status information.

The *Social Presence* component accepts, stores, and distributes social presence information to other members who are interested.

Example: A presence service can be built using several protocols (models), e.g. SIP, RPC, RMI. Taking SIP as an example, and noting that in SIP members are referred to as 'watchers', then the main entities involved in a social presence service would be:

- Watcher: A member (Client_A) that wants to know the presence of another member. In order to obtain this information, the watcher creates a SUBSCRIBE request, and as long as the watcher subscription state is active, a NOTIFY message will be received any time there is a status change of the watched member (Client_B).
- Presence User Agent (PUA): A Presence User Agent manipulates presence information to extract a presence (for a member). This manipulation can be the side effect of another action (e.g. sending a SIP REGISTER request to add a new Contact) or can be done explicitly through the publication of presence documents.
- Presence Agent (PA): A Presence Agent is a SIP User Agent which is capable of receiving SUBSCRIBE requests, responding to them, and generating notifications of changes in presence state. A Presence Agent must have knowledge of the presence state of the member. This means that it must have access to presence data manipulated by PUAs for member. One way to do this is by co-locating the PA with the proxy, as shown below as P-CSCF).
- Presence Server: A presence server is a physical entity that can act as either a Presence Agent or as a Proxy Server, responding to SUBSCRIBE requests. When acting as a PA, it is aware of the presence information of the member. When acting as a Proxy Server, the SUBSCRIBE requests are 'proxied' to another entity which may act as a PA.

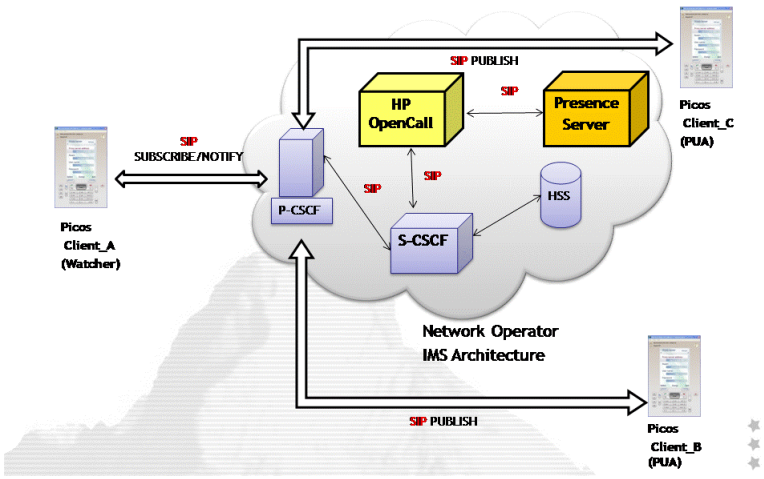


Figure 42 – Example of Social Presence implementation using SIP

In the figure above, when an Client_A wishes to know the social presence of another member (e.g. Client_B), it sends a SIP SUBSCRIBE request . This request identifies the watched member in the Request-URI (Client_B URI). This request eventually arrives to the Presence Server, and is first authenticated and then authorised.

Once the Presence Server has authorized the subscription it sends an immediate NOTIFY message containing the state of the watched member (Client_B) and the subscription. The presence state may be bogus, in the case of a pending subscription (indicating offline). This is to protect the privacy of the watched member, who may not want to reveal that they have not provided authorisation to the watcher. As the state of the watched member changes, the Presence Server generates NOTIFY messages containing the new state, and notifies all subscribers (and authorised) watchers members subscriptions.

A useful description of how social presence is employed in PICOS can be found in Section 13 in:

- PICOS Use Case 7: Presence

9.5.24.3Dependencies

Components that this component calls	Purpose
Consent Management	To determine is the member wishes their social presence to be made available to other members, and if so then which members (or all).
Location Sensor	To obtain the current location of the member.
Profile Management	To obtain other social presence information about the member.

Components that call this component	Purpose
Service Selection	To obtain the social presence of a member.

9.5.24.4Drawing

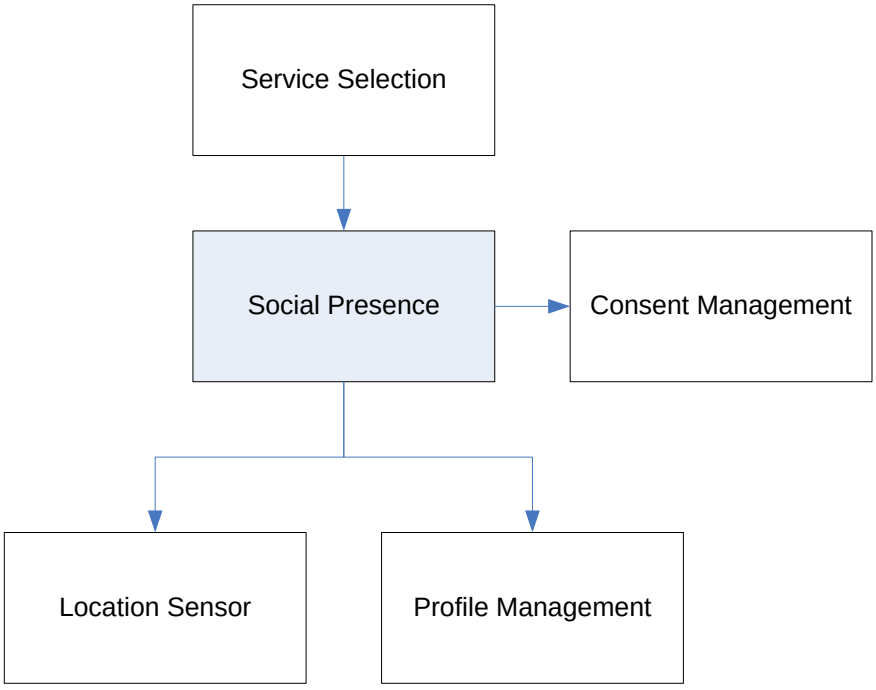


Figure 43 – Social Presence

9.5.25 Trust Negotiation

History: *Component contributor: HPL*

PICOS Principle (PP): 6, 16, 22, 23

PICOS Feature (PF): 1, 15

9.5.25.1Purpose

The *Trust Negotiation* component facilitates the establishment of a feeling of trust between two members.

9.5.25.2Description

When members engage with others they do so as a conversation, where one gathers information about the character (desirable qualities) of the other. This forms the basis of trust. Initially trust is low, but is built up over time as more 'personal' information is exchanged. Whether it is correct to call this a negotiation is not clear, but clearly a protocol exists which governs the transfer (or not) of information.

The knowledge built-up about the other member consists in part of reputation, a matching of profile (a profile is intended to express personal attitude to privacy) and personal preferences. A goal might be for members to relax their preferences as they become more comfortable with those whom they interact. PICOS could encourage this to encourage broader interaction across the community.

The *Trust Negotiation* component establishes a shared level of trust between members. The *Trust Negotiation* component may be used when forming a new sub-community to establish membership, or to identify members with a similar trust profile who may be willing to interact with one another.

The component may be called as a member service via the Service Selector component, or by the *Privacy Advisor* component.

9.5.25.3Dependencies

Components that this component calls	Purpose
Profile Management	To examine mutual trust.
Reputation Management	To use reputation as a basis for trust.
Sub-community Management	To create sub-community in which further trust can be established.

Components that call this component	Purpose
Privacy Advisor	To use the trust negotiation process to discover level of trust and then advice on privacy exposure.
Service Selection	When a member wishes to develop a trust with another member.

9.5.25.4Drawing

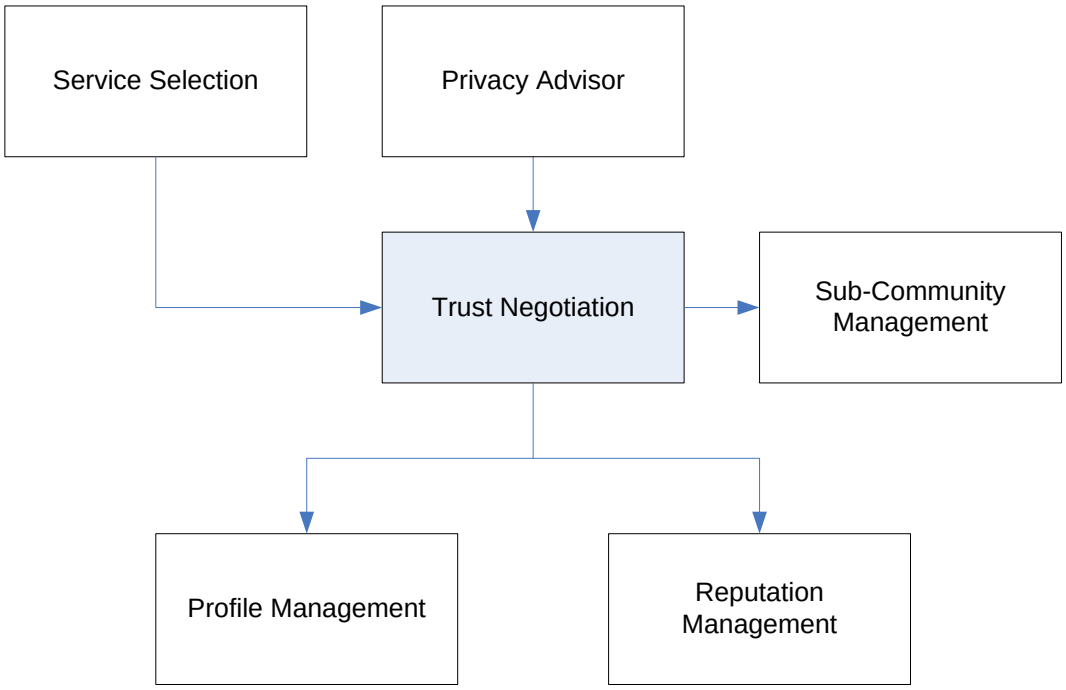


Figure 44 – Trust Negotiation

9.5.26 TTP Management

History: *Component contributor: UMA*

PICOS Principle (PP): 1, 2, 6, 12, 22

PICOS Feature (PF): 1, 3, 9, 13, 15

9.5.26.1Purpose

The *TTP Management* component provides the interface between the community and a trusted third party. It is most likely to be called as a result of a member selecting an external service, but may also be called for federated access or single sign-on by the *Access Control* component.

9.5.26.2Description

A community may need the services of an external trust authority to endorse identities. For example, an external TTP (e.g. a Certification Authority (CA)) binds a real identity to a public key, having first proved that the member has proved 'ownership' of the corresponding private key. Other TTPs may provide law enforcement, non-repudiation or system checking services. Access to the *Cryptography / Key Management* component may be required.

Communities that offer non-repudiation with legal consequence must enlist the support of TTP that follows a standard of legal protocol for certificate issuance. They are likely to be regulated, and have the power (but not necessarily the authority) to 'break' that anonymity of community members. Normally, CAs honour the wishes of the member, but if the member is involved in an illegal activity or breaches community policy, the TTP may be required to reveal the member's real identity.

The TTP Management component therefore provides the connection to TTPs for operation (use by members) and administrative purposes (use to create and exchange endorsement information).

Since the TTP services the needs of the whole community it will operate according to policy set by the *Policy Management* component.

A TTP is a useful source of recommendation for recruiting new members, thus the *TTP Management* component provides a link to the *Recruitment* component.

9.5.26.3Dependencies

Components that this component calls	Purpose
Cryptography / Key Management	To manage private or shared keys, and for key generation.
Policy Management	For community-wide operating practices relating to TTP.

Components that call this component	Purpose
Authentication	To validate federated identities.
External Service Delivery	Following a request to administer the TTP interface.
Recruitment	For external recommendation of new member for recruitment.

9.5.26.4Drawing

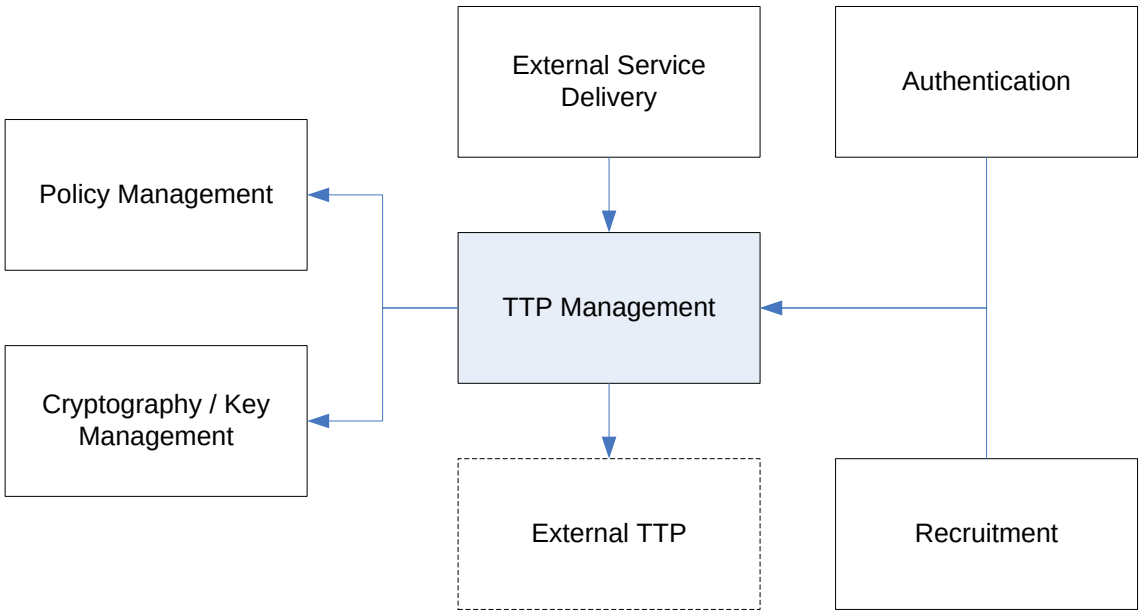


Figure 45 – TTP Management

9.6 Audit, Control and Reporting

The Audit, Control and Reporting component group contains the following components

9.6.1 Tier-1 Audit, Control and Reporting components

- Intrusion Detection

9.6.2 Tier-2 Audit, Control and Reporting components

- Accountability
- Audit
- Event Logging
- Event Reconstruction
- Policy Management

9.6.3 Accountability

History: *Component contributor:* UMA

PICOS Principle (PP): 1, 17

PICOS Feature (PF): 3, 15

9.6.3.1 Purpose

The *Accountability* component holds members accountable for their actions.

9.6.3.2 Description

The *Accountability* component monitors the behaviour of members to build trust confidence in the community, by attempting to detect dishonest activities. It can be thought of as the social conscience of the community. It is specifically engineered to detect activities that indicate fraudulent or inappropriate activity.

Information is collected from a variety of sources within the PICOS community. This information is analysed against predefined behaviour profiles. The results assist with community management and law enforcement, and feed into the reputation Management. On the basis of collected information (not defined yet) they are assessable.

The consequence of dishonest behaviour may be limited to the scope of the community, or may entail legal consequences. Identifying members in a community that aims to preserve privacy and protect identity has additional challenges. Where an action is performed under a pseudonym (or anonymously), the co-operation of an external Trusted Third Party (TTP) may be required in order to resolve the real identity behind the pseudonym. However, sometimes it is not necessary to discover the real identity of a pseudonymous/anonymous member in order to rectify an action or reprimand a member.

9.6.3.3 Dependencies

Components that this component calls	Purpose
Service Selection	For monitoring purposes.

Components that call this component	Purpose
Audit	To gather evidence.
Reputation Management	To gather evidence.
Event Logging	To gather evidence.

9.6.3.4 Drawing

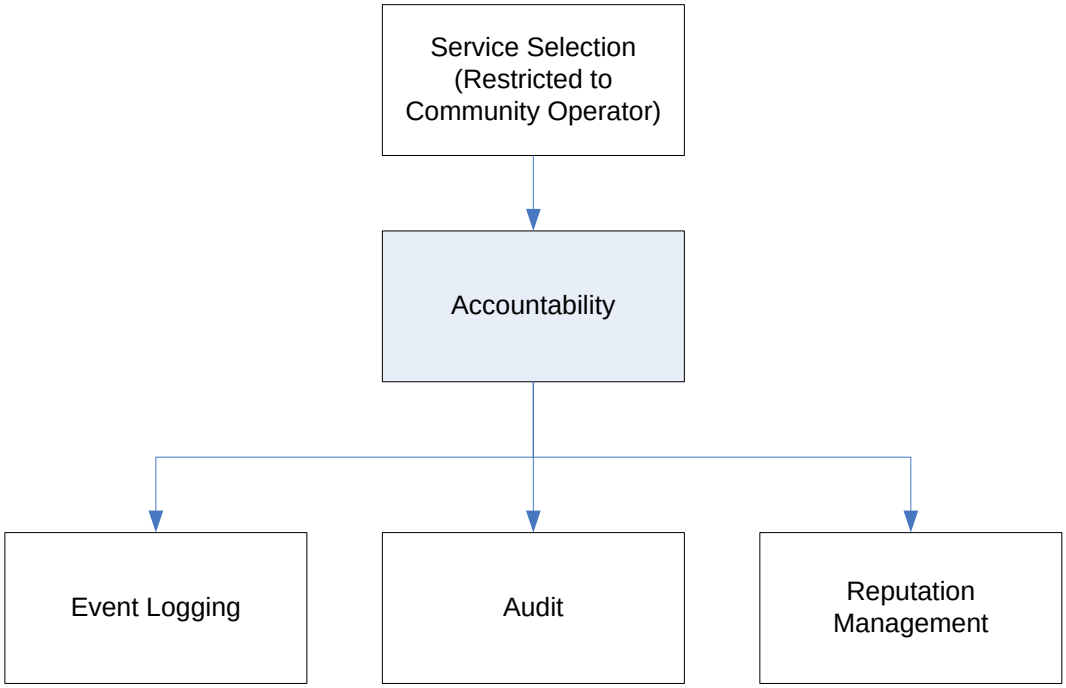


Figure 46 – Accountability

9.6.4 Audit

History: *Component contributor:* HPL

PICOS Principle (PP): 1, 14

PICOS Feature (PF): 1, 15

9.6.4.1 Purpose

The *Audit* component provides easy access to information that may need to perform an internal or external audit of the community.

9.6.4.2 Description

The *Audit* component works alongside the Event Logging component, creating a record on community activities that are required for community monitoring activities.

For example, it may be necessary to examine member accounts to check on authentication mechanisms, roles and rights. It may also be necessary to examine system logs to check compliance with legal and regulatory requirements.

The data examined is collected from many sources, e.g. membership (lifecycle) management (including registration and access control), use of sensitive functions (e.g. tagging, payment services), and the general administration of the community (event logging).

9.6.4.3 Dependencies

Components that this component calls	Purpose
Event Logging	To gather data about the community.

Components that call this component	Purpose
Service Selection	To gain access to Audit tools.

9.6.4.4 Drawing

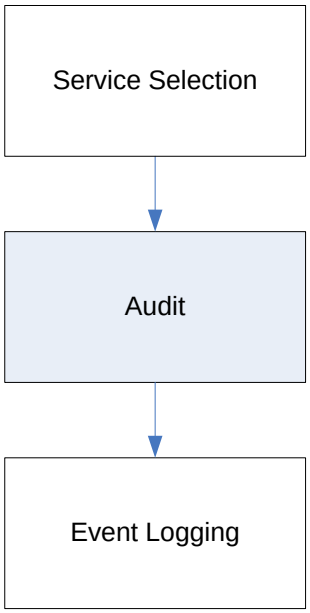


Figure 47 – Audit

9.6.5 Event Logging

History: *Component contributor:* GUF

PICOS Principle (PP): 1, 5, 14, 20

PICOS Feature (PF): 1, 15

9.6.5.1 Purpose

The *Event Logging* component maintains a reliable log of all community-related or member-related events.

9.6.5.2 Description

This *Event Logging* component plays an important role in establishing trust between a community member and the community platform. It documents all events (actions) which occur during the use of a community by members. Events comprise:

- Member related events: Actions performed by members, such as changing the current member location, uploading content, posting in forums, changing profile details, writing or receiving messages, etc. Other examples of events which are logged include which applications have accessed member profiles, what content has been submitted, which members have viewed that content and changes in privacy policies.
- Community related events: Events ranging from a new member joining the community or a sub-group, through to reporting technical or statistical events (e.g. number of members, average visiting time per user, reaction to particular advertisements, etc.)

Each member can decide which events are to be automatically communicated to other members. For example, uploading new pictures to a member's picture album could trigger a communication to all of his members of a sub-group or only to a smaller list of close friends.

Community related events benefit both members and the community provider. They provide information that helps manage provisioning, system availability and maintenance of the community, as well as indicate where to improve or adapt the services offered. They are also required to demonstrate that privacy policies are being respected.

The event log is available for members to inspect. Members can use this facility to verify that their profile data and content is being correctly managed, and to detect privacy breaches.

The *Event Logging* component collects event information from the other PICOS components. This information is archived to the secure (read-only) event log, where it is available for inspection, to prevent fraud.

Sometime it is not enough to simply monitor events. Events combine to form transactions, which can often reveal more about a community than individual events alone. The *Event Logging* component is able to associate events, based on its knowledge of the community, and thereby maintain a richer

record of the day-to-day use of the community. This can be of particular help when analysing the performance of services.

9.6.5.3 Dependencies

Components that this component calls	Purpose
None defined at present	

Components that call this component	Purpose
All components that give rise to event that affect the community.	

9.6.5.4 Drawing

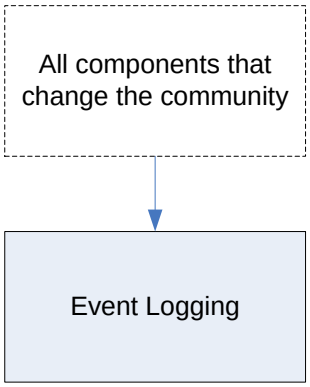


Figure 48 – Event Logging

9.6.6 Event Reconstruction

History: *Component contributor:* HPL, IfM-Geomar

PICOS Principle (PP): 1, 20, 23

PICOS Feature (PF): 1

9.6.6.1 Purpose

The *Event Reconstruction* component is responsible for assisting with the rebuilding of the community in the event of a catastrophic failure or should there be a need to create a duplicate community for investigative purposes.

9.6.6.2 Description

The *Event Reconstruction* component is concerned with the recovery of a system or of lost data. It also provides a means to gather evidence and test system functionality. Overall, event reconstruction creates greater member confidence.

Online communities are relied upon by millions of members to provide a reliable, always available resource. In reality, this is not the case for the Internet or web-based information, which is where these expectations are set. The average lifespan of a web page is 44 -75 days.

The PICOS community must be able to reconstruct itself in the event of failure. Website recreation utilities already exist[12], as do website and online services (Web services) that enable a website to be recreated reflecting its status at any point in history[13]. To a degree, it is possible to restore lost information by trawling the Web using one of the many search engines (e.g. Internet Archive, Google, Live Search, and Yahoo). However, despite the belief that 'no information is ever lost', reconstructing a community can be necessary and difficult.

The *Event Reconstruction* component works along side the *Event Logging* and *Audit* component to rebuild a community using details of transaction, events and archived data.

Retaining such extensive information obviously leads to concern about privacy. Access and operation of the *Event Reconstruction* component is thus tightly controlled.

[12] Frank McCown at Harding University created a tool called Warrick that helps the user to recover any lost website (or single web page) automatically (http://warrick.cs.odu.edu/).
[13] The 'wayback machine'.

9.6.6.3 Dependencies

Components that this component calls	Purpose
Event Logging	To gather information to facilitate the reconstruction.
Audit	To gather information to facilitate the reconstruction.

Components that call this component	Purpose
Service Selection	To gain access to Audit tools.

9.6.6.4 Drawing

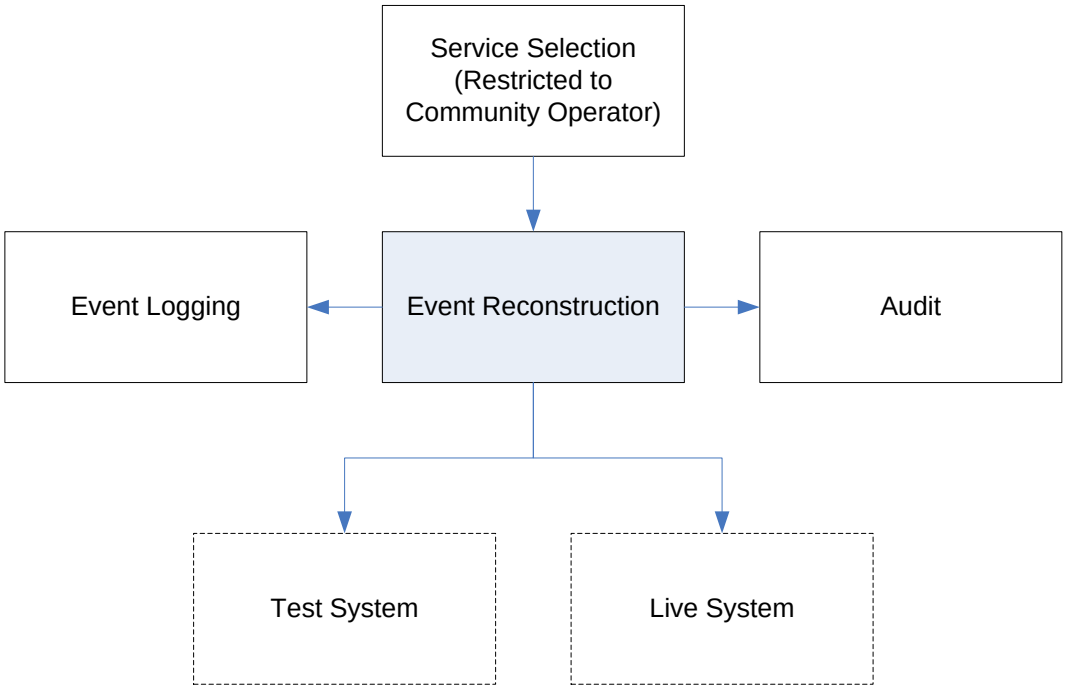


Figure 49 – Event Reconstruction

9.6.7 Intrusion Detection

History: *Component contributor:* BRNO

PICOS Principle (PP): 4

PICOS Feature (PF): 13

9.6.7.1 Purpose

The *Intrusion Detection* component is responsible for detecting attacks on the PICOS community.

9.6.7.2 Description

The *Intrusion Detection* component detects attempts to compromise the security of the PICOS system, with primary attention to the PICOS server. Security is usually understood in terms of achieving confidentiality, integrity/authenticity and availability of data and resources.

It is important to consider intrusion detection at various layers.

- Security of the network should be protected by firewalls and logs regularly analysed.
- Security of the operating system also plays an important role.

Intrusion detection is a complex task, and both of the above examples are beyond the scope of the PICOS project. The *Intrusion Detection* component in PICOS focuses on analysis of higher level events like authentication and registration attacks, events related to reputation management, attempts to access information protected with access control, and other unexpected behaviour (e.g. large number of posts, Spam, etc.). Thus, this component can be triggered by many of the other components, e.g. *Access Control*, *Communication Management* and *Service Selection components*, and probably *Content Sharing*, *Registration* and *Reputation Management* components.

A significant part of intrusion detection can be automated. However, it is very important to have the option to manually check log files in an easy manner and to verify the conclusions of the automated intrusion detection system.

Actions performed by the *Intrusion Detection* component include temporary and permanently blocking members and/or nodes, updating reputation and credential information (white/black lists).

It should also be noted that intrusion threats can come from insiders of a community, just as they come from outside. Their intent may be to subvert system integrity (information, reputation, logs, etc) or gain access to restricted data (private keys, bank accounts, VISA, PayPal, sensitive personal information, etc).

Tagging helps to identify data that is considered important/sensitive and thereby can facilitate the protection by focusing on the most sensitive data. For example private keys used for authentication must be strongly protected (e.g. hardware token, smart cards), because a breach could compromise the whole system.

It is important to recall that the law requests that the server stores sensitive information in a secure way to protect members' privacy from attackers.

9.6.7.3 Dependencies

Components that this component calls	Purpose
None defined at present	

Components that call this component	Purpose
Access Control	To trigger the intrusion detection response process.
Service Selection	To trigger the intrusion detection response process.
Communication Manager	To trigger the intrusion detection response process.

9.6.7.4 Drawing

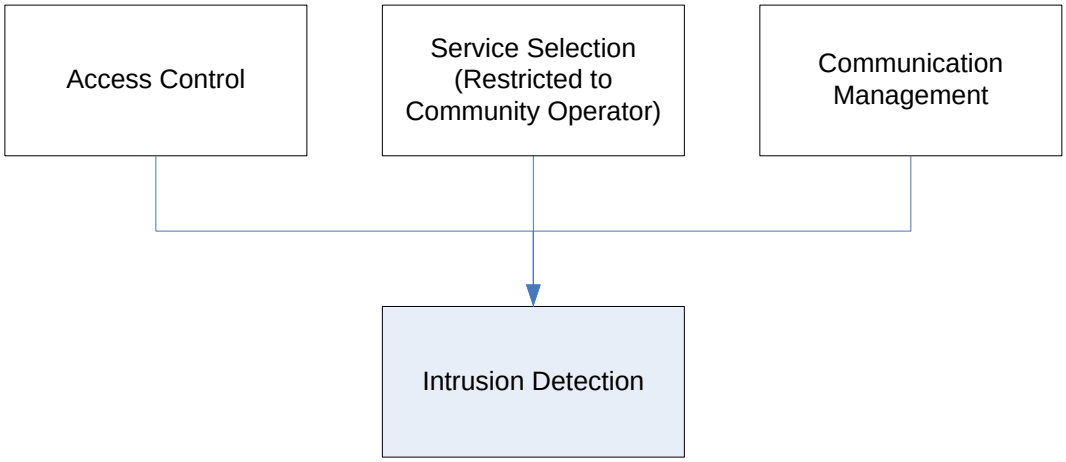


Figure 50 – Intrusion Detection

9.6.8 Policy Management

History: *Component contributor:* All

PICOS Principle (PP): 1, 3, 5,

PICOS Feature (PF): 2, 3, 11, 13

9.6.8.1 Purpose

The *Policy Management* component is responsible for policy that affects the whole community.

9.6.8.2 Description

Like many communities, policies play an important role in the PICOS community. Policies allow information to be communicated to members, demonstrating openness and transparency (and thus engendering trust), and allow broad 'default' operating practices to be established for large parts of the community.

Policy management falls into two main categories:

- Policy creation and sharing: This is where policies are recorded (edited) and made available to members. Often a standardised way of communicating policy will be used, and sometime this can be in machine readable form. Examples include W3C's P3P standard and IBM's EPAL proposal.
- Policy enforcement: A policy that establishes a standard, but which is not enforceable is arguably of little value. Enforcement can take many forms, but two popular ones are 1) proactive design of enforcement mechanisms and 2) reactive monitoring. Both approaches have their merits. At this stage in the project it is not possible to say which is preferable, or to explain how proactive enforcement would be achieved.

Policies remain a focus for the PICOS project and will be considered in more detail as the prototype evolves. Policies are also an area of research that PICOS might want to pursue.

Policy affects many aspects of a community. Two specific areas are authentication (and authentication method selection) and relationships with trusted third parties.

9.6.8.3 Dependencies

Components that this component calls	Purpose
Authentication Selector Method	To establish a community-wide set of approved authentication methods.
TTP Management	Where the chosen authentication method required the services of a Trusted Third Party (TTP), e.g. where an authentication token was issued by another community.

Components that call this component	Purpose
Service Selection	To administer community-wide policies.

9.6.8.4 Drawing

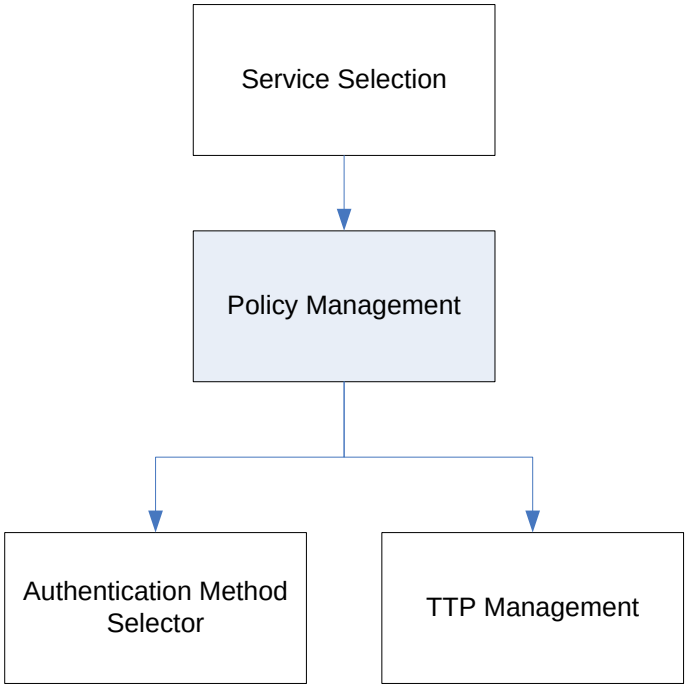


Figure 51 – Policy Management

9.7 Member Administration

The Member Administration component group contains the following components

9.7.1 Tier-1 Member Administration components

- Identity Lifecycle Management
- Sub-community Management

9.7.2 Tier-2 Member Administration components

- Authentication Method Selection
- Consent Management
- Cryptography / Key Management
- Delegation
- Personal Profile Management
- Privilege Management
- Registration
- Revocation

9.7.3 Authentication Method Selection

History: *Component contributor:* ATOS

PICOS Principle (PP): 17, 18

PICOS Feature (PF): 3

9.7.3.1 Purpose

The *Authentication Method Selection* component enables the selection of authentication method.

9.7.3.2 Description

Several authentication methods will be supported by the PICOS community. Exact details are not yet available, and will depend on the capabilities of the client device, but could include password, biometric, token and credential. The choice of which to use depends on the situation and context, and on the sensitivity of the action being performed. For highly sensitive information or actions, strong authentication is preferred.

The *Authentication Method Selection* component responds to the request from the *Access Control* component and the *Authentication* component, and for a given authentication method. The choice will be decided through community policy (*Policy Management* component), member profile (*Profile Management* component) and member preferences (*Data Minimisation* component and *Privacy Advisor* component), all under the direction of the *Access Control* component.

Where authentication takes place at the client, it is possible that a local version of this component will be required, i.e. where the *Access Control*, *Authentication* and possibly the *Authorisation* component are located at the client.

9.7.3.3 Dependencies

Components that this component calls	Purpose
Policy Management	To identity the preferred method(s) of authentication for the community as a whole.
Profile Management	To identity the preferred method(s) of authentication of the member.
Privacy Advisor	Where the member has a choice, the Privacy Advisor helps to select the method that best achieves data minimisation.

Components that call this component	Purpose
Authentication	To indicate to the member the preferred method(s) of

	authentication, having been triggered by the *Access Control* component.

9.7.3.4 Drawing

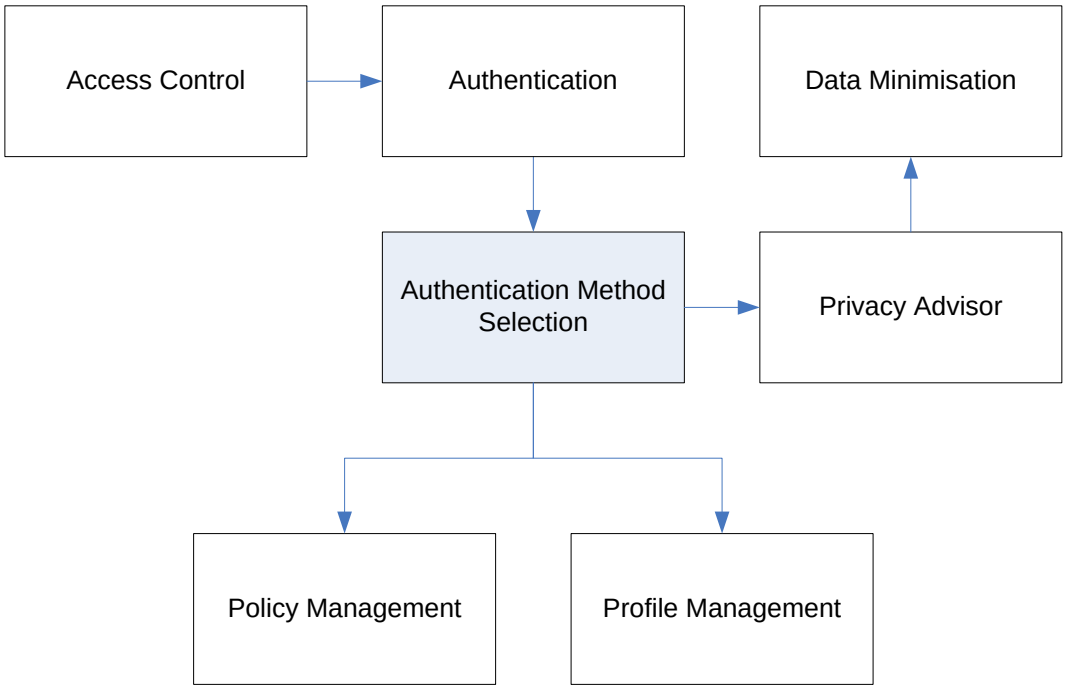


Figure 52 – Authentication Method Selection

9.7.4 Consent Management

History: *Component contributor:* HPF

PICOS Principle (PP): 2, 3, 4

PICOS Feature (PF): 2, 10

9.7.4.1 Purpose

The *Consent Management* component allows members to grant consent for their personal information to be used in the way members wish.

9.7.4.2 Description

The *Consent Management* component plays an important role in both privacy management and trust management. It stores and enforces user-defined policies with respect to the sharing of members' profile information (and other member data) with other members and with external services. It indicates if the member gave consent for this data to be shared with others and, if so, what terms and conditions apply.

The *Consent Management* component also allows member to modify or withdraw their consent, and it invokes the community-specific procedures that are applied when consent is withdrawn, noting that different communities may interpret consent changes in different ways, e.g. deletion, change to access rights which restrict access to certain roles only. The latter involves the *Policy Management* component.

9.7.4.3 Dependencies

Components that this component calls	Purpose
Policy Management	To determine community policy on managing member information.
Profile Management	To determine member preferences on sharing personal information.

Components that call this component	Purpose
Content Sharing	To respect member preferences when sharing personal information.
External Services Delivery	To check consent before sharing information with an external service provider.
Social Presence	To take into account member current status before sharing personal information.

9.7.4.4 Drawing

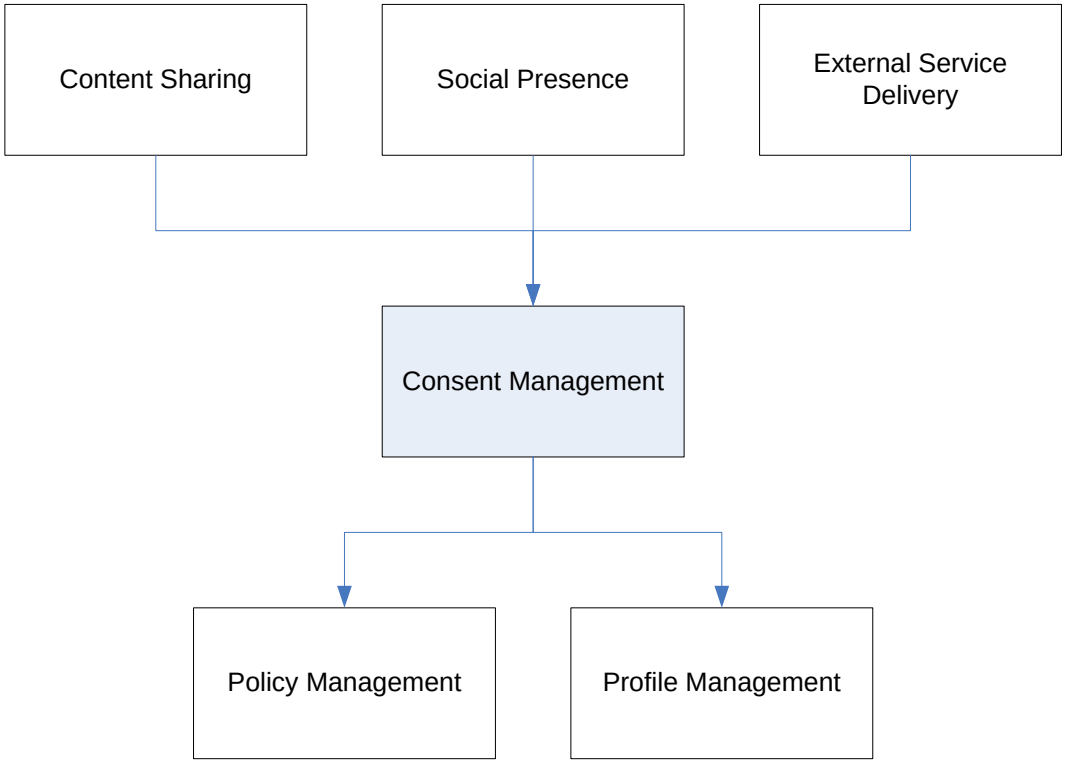


Figure 53 – Consent Management

9.7.5 Cryptography / Key Management

History: *Component contributor:* UMA

PICOS Principle (PP): 1, 4, 8, 9, 17

PICOS Feature (PF): 1, 3, 10, 15

9.7.5.1 Purpose

The *Cryptography / Key Management* component implements cryptography and key management mechanisms and services.

9.7.5.2 Description

The *Cryptography / Key Management* component provides support for symmetric and asymmetric public key cryptography, offering confidentiality and integrity mechanisms. The mechanisms supported fall into four categories:

- Confidentiality: RSA encryption/decryption and signatures
- Integrity: Hash, AES encryption/decryption, etc
- Non-Repudiation: DSA signatures, Schnorr signatures, ElGamal signatures & encryption
- Traceability: Group, FTMGS signatures, etc ...

Key management is an important part of any cryptography scheme. Keys need to be created, stored and generally managed securely. Retrieving the correct key to use with a mechanism for a particular purpose is also something that needs to be handled with care, especially where keys are transferred from one domain (e.g. server) to another (e.g. client) before use.

Keys are stored by the *Secure Repository* component, which may be implemented on either the server or client, or both.

Access to keys is indirectly controlled by the *Access Control* component, and access will be dependent on valid authorisation (*Authorisation* component). Access is granted where authentication is satisfied and the role is appropriate.

Many components may require access to this component, but the main components are *Network Security*, *Anonymisation*, *Secure Repository*, and *Authentication*.

9.7.5.3 Dependencies

Components that this component calls	Purpose
Anonymisation	To generate keys.
Authentication	To support mutual authentication and credential authentication methods.
Network Security	To access algorithms/keys required for network confidentiality and integrity.
Secure Repository	When preparing sensitive information for storage.

Components that call this component	Purpose
Secure Repository	To store/retrieve keys.

9.7.5.4 Drawing

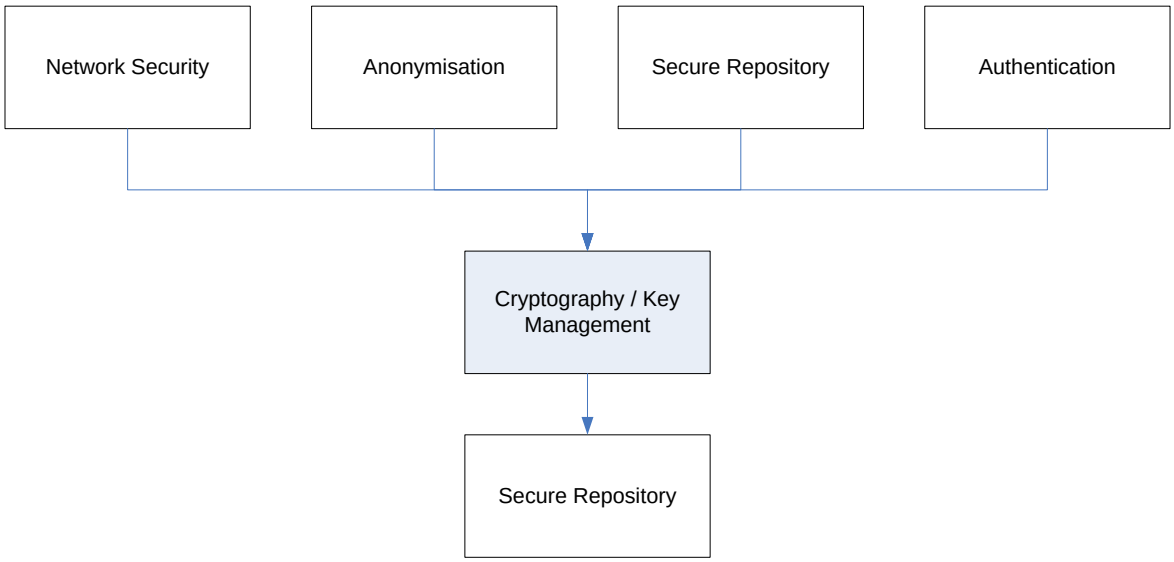


Figure 54 – Cryptography /Key Management

9.7.6 Delegation

History: *Component contributor:* ATOS

PICOS Principle (PP): 2, 23

PICOS Feature (PF): 3, 7, 15

9.7.6.1 Purpose

The *Delegation* component allows members to transfer privileges between partial identities which may (or may not) belong to different members (with different root identities).

9.7.6.2 Description

Using the *Delegation* component, members can delegate privileges that come with their partial identity to other partial identities or to other members' partial identities. The reason they might want to do this is to allow another member to perform an action with an asset that they considered personal. Whenever delegation is invoked, all actions are logged (by the Event logging component) so that it is clear what events have taken place and by whom.

Typically, delegation will expire on completion of an action or after a predetermined time. The member to whom privileges are delegated cannot influence the privileges originally assigned to the delegating member, but it is possible that the reputation of the delegating member (and the delegated member) might be affected by events that occur while delegation is active.

Delegation is only possible with the consent of both parties (*Consent Management* component), and if accepted will affect the profiles of one (and possible both) members (*Profile Management* component, *Privilege Management* component). It is also appropriate that both members are formally notified of the change (*Notification* component).

Delegation can takes two forms:

- Delegation of authentication: Assuming that a credential is used to authenticate a member to a community, that a credential can be delegated to another system or community, pass-through authentication is possible. This is when a member accesses one community, which then automatically signs the member into another community. This is an example of Single Sign-On (SSO), e.g. Open Id.
- Delegation of authorisation: A member can delegate their access rights to another member, so that the other member can act on the delegating member's behalf.

9.7.6.3 Dependencies

Components that this component calls	Purpose
Consent Management	To check the consent of both partied involved in the delegation
Notification	To notify the delegated member that delegation has taken place, e.g. if delegation occurs automatically or because of prior agreement between the two parties.
Privilege Management	To update the privilege of the delegated member.
Profile Management	To update the profile of the delegated and delegating member.

Components that call this component[14]	Purpose
Service Selection	To trigger delegation.

9.7.6.4 Drawing

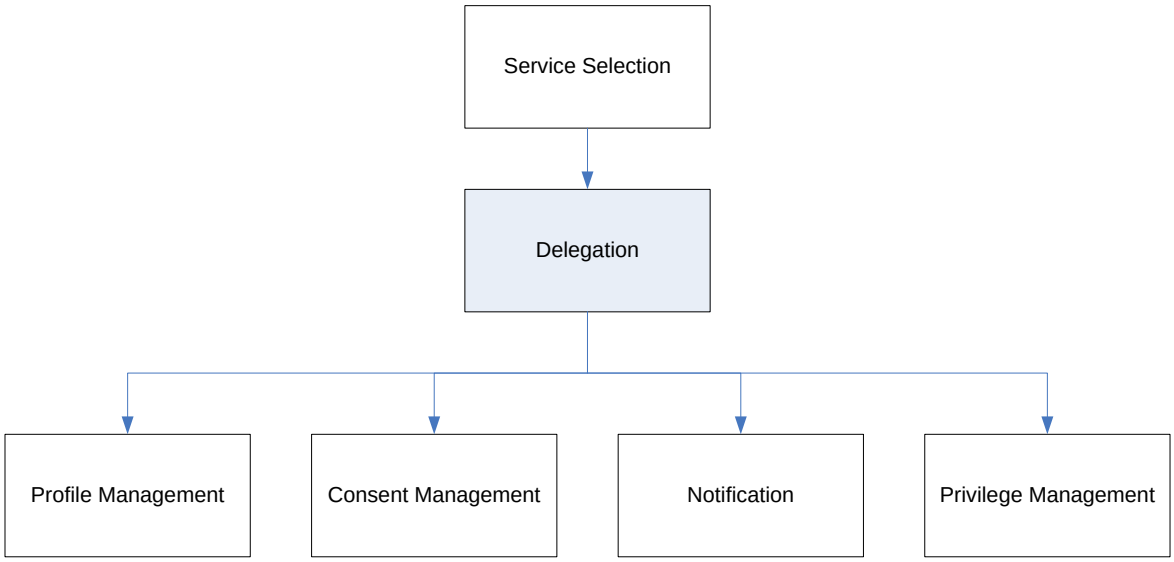


Figure 55 – Delegation

[14] The dependencies with other components of PICOS architecture will be basically the first point to access to the network operator (the SIP Proxy or P-CSCF in case an IMS network is available).

9.7.7 Identity Lifecycle Management

T1 PICOSenhancing

History: *Component contributor:* ATOS

PICOS Principle (PP): 11, 17, 18

PICOS Feature (PF): 1, 3, 4

9.7.7.1 Purpose

The *Identity Lifecycle Management* component manages all aspects of identity in the community.

9.7.7.2 Description[15]

The *Identity Lifecycle Management* component interacts with the main components that support the lifecycle of a member, namely the *Registration*, *Partial Identity Management*, *Privilege Management*, *Delegation* and *Revocation* components. It also accesses the *Policy Management* component to determine community policy.

Identities experience a well-defined lifecycle in PICOS communities (from enrolment of members until their termination). The management of such a lifecycle is a core feature of any identity management solution, whether centralised, user-centric, federated or a combination of several models. Essentially, the Lifecycle Management presents a framework which ensures that identity information is accurately maintained in a context, in accordance with applicable policies, standards and regulations.

Lifecycle Management results in a level of assurance[16] which in turn results in an acceptable level of risk for the individual and the community. In practice, the level of assurance will vary depending on various factors, e.g. personal, business, legal, regulatory, internal policies, etc.

Identity Management consists of:

- Registration
- Creation
- Modification
- Delegation
- Revocation

Identities follow a lifecycle, e.g. established, modified, suspended, terminated, archived and transferred. An identity is typically only valid for a period of time (i.e. has a start/end date) and its

[15] See ISO/IEC JTC 1/SC 27 WG5 N7109 "A framework for Identity Management" pp. 22-26

[16] It may benefit all organizations and individuals in multiple contexts to provide a certain assurance level that an identity is not compromised or will not be repudiated. The assurance level required is determined by the risk of not effectively distinguishing entities with the means of the context.

existence may be dependent on context. The identity of an entity may persist after the entity ceases to exist when entity information still needs to be managed. The possible states of an identity are the following:

- Not Established: the identity of an entity is unrecognised in a given context. In some cases the entity exists, and in others the entity does not exist.
- Established: the identity of an entity is recognized in the context but the entity is not yet able to interact with other entities in the context.
- Activated: the identity of an entity is recognized in the context and the entity is able to interact with other entities in the context according to the purposes of the context.
- Suspended: the identity of an entity is recognized in the context. However, the entity is no longer able to interact with other entities in the context.
- Terminated: an entity is no longer recognized in a context.
- Archived: an entity is no longer recognized in a context but records may be required to remain available to determine whether or not an entity has in the past been recognized in a context with a particular identity.

These states and the sequences of events that can cause transitions between them are depicted below:

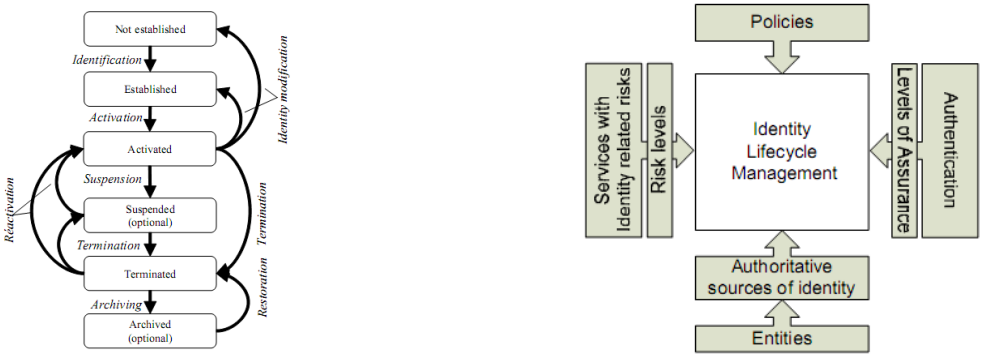


Figure 56 – States in an identity lifecycle

A useful description of how identity lifecycle management is employed in PICOS can be found in Section 13 in:

- PICOS Use Case 1: Registration
- PICOS Use Case 2: Accessing the community
- PICOS Use Case 4: Multiple Partial Identities
- PICOS Use Case 5: Revocation

9.7.7.3 Dependencies

Components that this component calls	Purpose
Delegation	To delegate authority to another partial identity.
Partial Identity Management	To monitor the creation of partial identities.
Policy Management	To manage policies relating to partial identities.
Privilege Management	To set/modify privileges for partial identities.
Registration	To register new root identity (member) and prepare for the creation of a partial identity.
Revocation	To revoke a partial identity or a root identity.

Components that call this component	Purpose
None. (Internal community function)	

9.7.7.4 Drawing

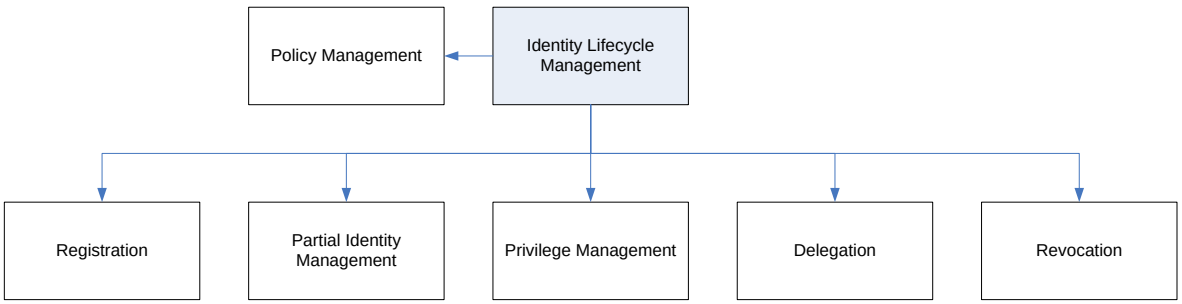


Figure 57 – Identity Lifecycle Management

9.7.8 Privilege Management

History: *Component contributor:* ATOS, UMA, ITO

PICOS Principle (PP): 3

PICOS Feature (PF): 2, 3

9.7.8.1 Purpose

The *Privilege Management* component enables the setting and modification of member privileges, typically by the community operator.

9.7.8.2 Description

Every member is assigned privileges which grant or deny access to community resources. Privileges are established using the *Privilege Management* component at the time the member is enrolled, triggered by the *Registration* component. Members who have multiple identities (partial identities) will have multiple sets of privileges (privilege sets), but only one set of privileges is assigned to each partial identity.

In a typical computer operating system, users are assigned to groups. In a PICOS community, groups are represented by sub-communities. Just as with groups, sub-communities can only be accessed by members who possess the prerequisite privilege.

Partial identities can inherit the privileges of a root identity, or of another partial identity using the *Delegation* component. Delegation is also an area where privileges will change (semi-) automatically.

Another way to consider privilege sets is as definitions of roles. A member may be asked to carry out a specific function on behalf of the community. To perform these functions the member will need an enhanced set of privileges. Thus a privilege set can be assigned to a role, e.g. manager, auditor.

The *Privilege Management* component manages privileges, i.e. it facilitates the creation, modification and assignment of a privilege set to a partial identity.

Privileges are treated just like any other personal information. Following the model of data minimisation that PICOS adopts, only those privileges necessary to perform a function have to be declared. In addition, the rights demonstrated by privileges can be demonstrated in an anonymising (or privacy friendly) way using zero knowledge proofs (ZKP).

The management of privileges makes the assumption that the identification of entities is guaranteed[17]. Managing privileges involves four main activities:

[17] The authentication of 'proper owners' in PICOS is flexible and accounts for different levels of authentication depending on context, policies, etc. (i.e. users could also be pseudonymously or anonymously authenticated to the community).

- Definition of privileges and the privilege set
- Validation of authorisation to assign privileges to entities, ensuring that privileges are securely granted to entities based on conditional attributes specific to each community, roles and tasks and possibly approvals from different actors in some cases
- Provisioning of authorised privileges to entities, so that access to community information and resources in based on a trusted identity/entity
- Control of provisioned privileges when accessing resources, so that the manner in which the controls are operated determines how privileges must be defined and the basis of such controls may be heterogeneous, i.e. based on given mandates, role assignment, contextual constraints and conditions, and possibly automated authorisations

When accessing resources with privilege thresholds, other conditions may also need to be fulfilled, e.g. rules based on context or scenario. Reference to the *Policy Management* component, the *Social Presence* component and the Reputation Management component may be required.

Members are able to view their privileges via their personal profile using the *Profile Management* component.

Privileges may also be adjusted (promoted/demoted) according to reputation, feedback or a change in role.

9.7.8.3 Dependencies

Components that this component calls	Purpose
None defined at present	

Components that call this component	Purpose
Delegation	To retrieve and transfer privileges to a partial identity. Note: Trigged by *Service Selection* component
Profile Management	To set/modify member privileges.
Registration	To assign privileges to a new member.

9.7.8.4 Drawing

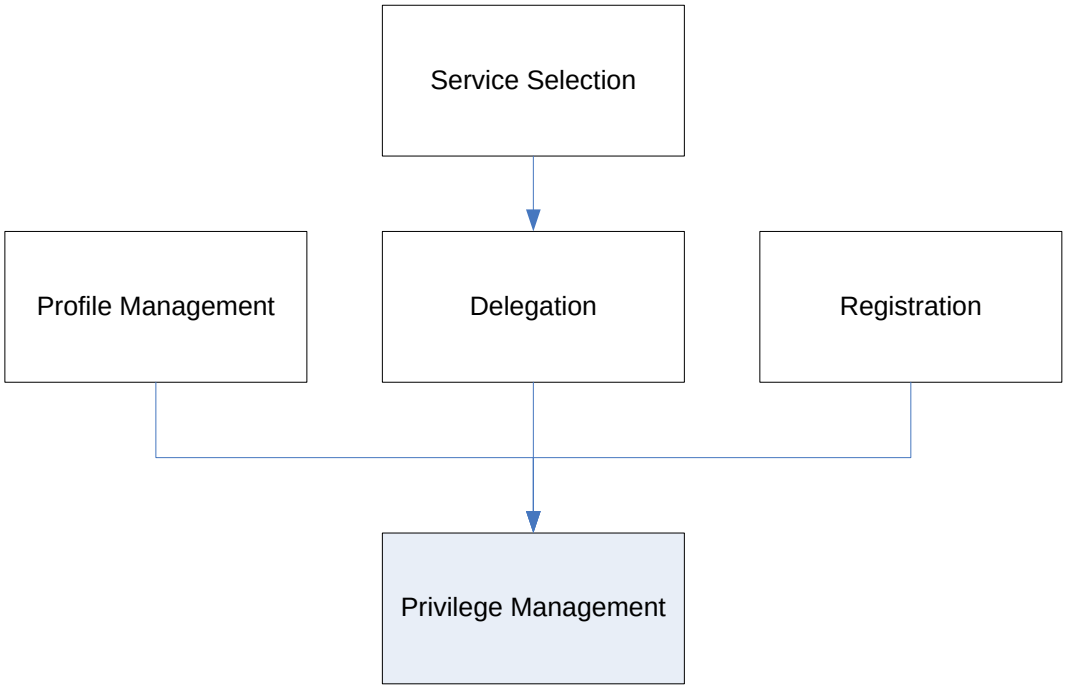


Figure 58 – Delegation

9.7.9 Profile Management

History: *Component contributor:* BRNO

PICOS Principle (PP): 4, 23

PICOS Feature (PF): 4, 8

9.7.9.1 Purpose

The *Profile Management* component provides access to the profile of an entity.

9.7.9.2 Description

Every partial identity possesses a personal profile that contains partly public and partly private information. Note that partial identities apply to a member, but also to other entities within the community that need to be uniquely identified, e.g. external services.

The personal profile describes each member's attributes, interests, general preferences and privacy preferences. The profile also contains a dynamic indication of a member's availability to interact with other members. Members can choose what information maintained by the community is revealed to others, e.g. their diary. Willingness to receive advertising is also recorded in the profile. This is managed by the *Consent Management* component.

Members can modify their personal profiles at any time using the *Profile Management* component, but they cannot alter settings that are established by the community, e.g. role, status, sub-community membership, etc. Information can be added to or removed from a profile, so long as the result is not misleading to other members.

A profile may also carry social presence (status) information, which is conditionally visible to other members and the community operator. For other members, social presence provides a real-time indication of the availability of the member. For the community operator, social presence indicates the status of the member's membership, e.g. pending, expired, account locked, membership 'paid up', roles, sub-communities owned, duration of membership, etc.

Privacy preferences enable members to specify the level of privacy that applies to all or part of the personal information that they share with others. Privacy preference can apply to individual data items or to a set of data items which together profile the member in a particular way. Privacy preferences are interpreted by the community and acted upon, thus fulfilling the privacy wishes of the member. The ability of the system to enforce preferences in a given situation/scenario (or context) is a key factor in establishing member trust and confidence.

9.7.9.3 Dependencies

Components that this component calls	Purpose
None defined at present	

Components that call this component	Purpose
Authorisation	To check the privacy preferences in the profile.
Consent Management	To update the privacy preferences in the profile.
Content Sharing	To check the privacy preferences in the profile.
Privacy Advisor	To check the privacy preferences in the profile.
Registration	To create the profile.
Service Selection	To access the profile management service.

9.7.9.4 Drawing

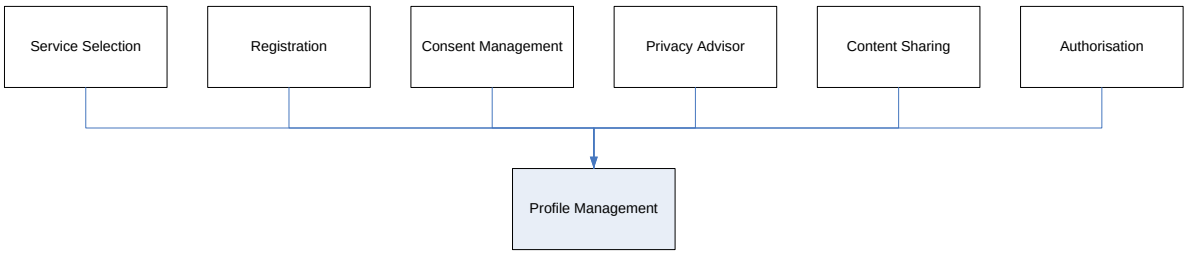


Figure 59 – Profile Management

9.7.10 Registration

T2 PICOSenhancing

History: *Component contributor:* ATOS

PICOS Principle (PP): 17

PICOS Feature (PF): 3

9.7.10.1 Purpose

The *Registration* component handles new member registration.

9.7.10.2 Description

The *Registration* component allows an individual to acquire membership of a community for a period of time, and enjoy the benefits associated with membership. Registration involves creating a unique root identity and one or more associated partial identities, and acquiring the ability to authenticate that identity to the community. It also involves agreeing to the terms and conditions of the community.

For a PICOS community, registration also involves a member stating privacy preferences (these processes are handled by separate components) and being assigned privileges.

Registration involves an individual introducing themselves to the community and establishing the right to gain access. This process usually involves the individual supplying information, some of which can be personal. Registration occurs prior to authentication and authorisation. In a simple model, members register directly with a community, much like an individual registers with a web-based online service.

In a mobile setting, it is possible that the individual has already authenticated themselves to the mobile operator (e.g. T-Mobile) and in theory does not need to authenticate or even register with the community, although in practice (and in the case of a PICOS community) members will be expected to register with the community directly.

Despite the convenience of having just one point of authentication, the community must always satisfy itself that only legitimate members can gain access to its resources. For this reason, the *Access Control* component will always be the first point of contact for new and existing members, and will direct them to the *Registration* component. It is at the time of registration that new members are allocated privileges and resources.

When a member subsequently attempts to access the community, they will be asked for their identity and challenged in order to authenticate. Once access is granted, privileges are retrieved and other members are able to check this member's status.

Example: In the figure below P-CSCF is an authentication point provided by the mobile operator, here using a SIP[18] protocol[1920].

[18] SIP is popular with 3GPP (Third Generation Partnership Project)

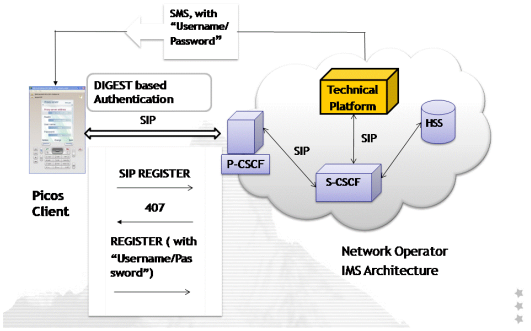


Figure 60 – Example of Registration implementation using SIP

A useful description of how registration is employed in PICOS can be found in Section 13 in:

- PICOS Use Case 1: Registration
- PICOS Use Case 2: Accessing the community
- PICOS Use Case 4: Multiple Partial Identities

[19] All RFC 3261 compliant user agent (SIP client application) support Digest Authentication, which uses a shared secret, as a means for authentication to a SIP Proxy. The registration allows a user agent to express that it is an appropriate entity to which requests should be sent for a particular SIP address (SIP URI).

[20] SIP traffic is initiated by a registration that is challenged by the P-CSCF using Digest based authentication. The Technical Platform relies on the S-CSCF to manage the access control and handle the SIP P-Asserted identity header so that each SIP component can rely on the SIP user identity (public SIP URI) behind the S-CSCF.

9.7.10.3Dependencies

Components that this component calls	Purpose
Partial Identity Management	To create the initial partial identity.
Profile Management	To set root identity profile.

Components that call this component	Purpose
Identity Lifecycle Management	To trigger the registration process.

9.7.10.4Drawing

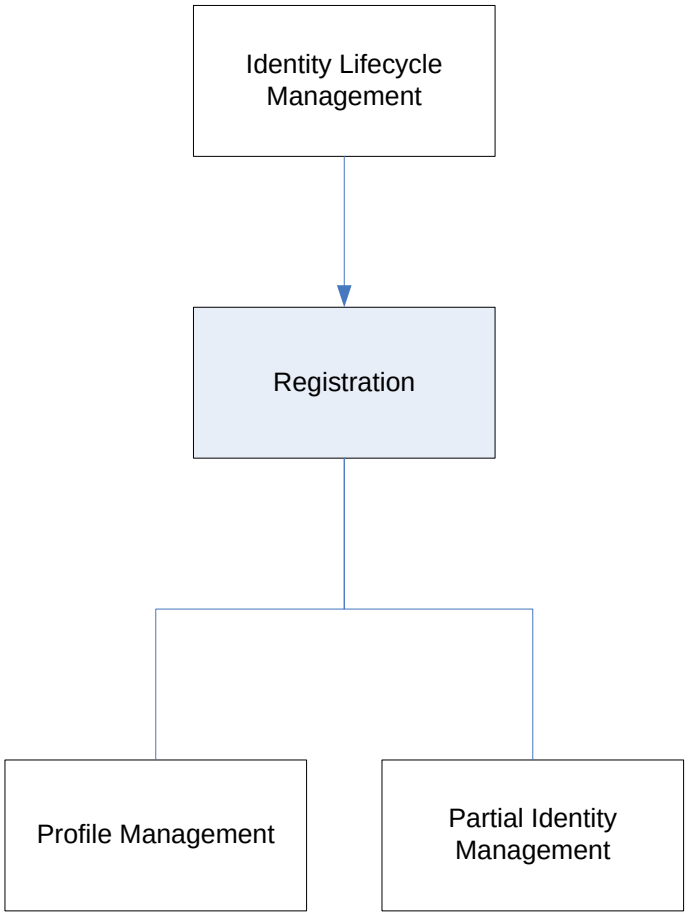


Figure 61 – Registration

9.7.11 Revocation

History: *Component contributor:* ATOS

PICOS Principle (PP): 2, 12, 23

PICOS Feature (PF): 2, 10, 15

9.7.11.1Purpose

The *Revocation* component is responsible for terminating a member's access to the community.

9.7.11.2Description

The *Revocation* component is called whenever a member wishes to leave the community (or is asked to leave the community), or when a member wants to terminate a partial identity.

When a member wishes to leave (resign) from a community they will most likely leave behind information that must be retained and protected. It is unlikely that this information can simply be deleted from the community (or even removed) since it may be shared or required for legal purposes. Regardless, it could contain personal information (personal profile, pictures, personal messages and other personal assets) that continues to need protection. Revocation requests the *Anonymisation* component to pseudonymise (in a reversible way, such as encryption) all references in all databases to the identity of this individual, and then after a second period of time, all these reversible pseudonyms are converted to irreversible pseudonyms (for example a hash of the previous pseudonym). Additionally, after a period of time, all sensitive data belonging to this individual must be erased.

Revocation is not concerned with data that has been transferred (e.g. as an attachment to a personal message) to another member or group of members, as this data is considered to be outside the original member's control.

Revocation can only be initiated by the community operator. However, it is possible for a member to revoke partial identities (which, incidentally are always linked to the creating root identity) but all personal assets are transfer to the creating identity, and the legal requirement to retain information still applies.

Revocation is recorded as a community event, thus it is possible to recreate an identity if necessary.

Revocation also influences membership status, and forces authentication information to be destroyed (to prevent further access). Reputation information may also need to change as a result of the member leaving the community.

A useful description of how revocation is employed in PICOS can be found in Section 13 in:

- PICOS Use Case 3: Revocation

9.7.11.3Dependencies

Components that this component calls	Purpose
Partial Identity Management	To revoke a partial identity.
Profile Management	To revoke a root identity profile.

Components that call this component	Purpose
Identity Lifecycle Management	To activate the revocation process.

9.7.11.4Drawing

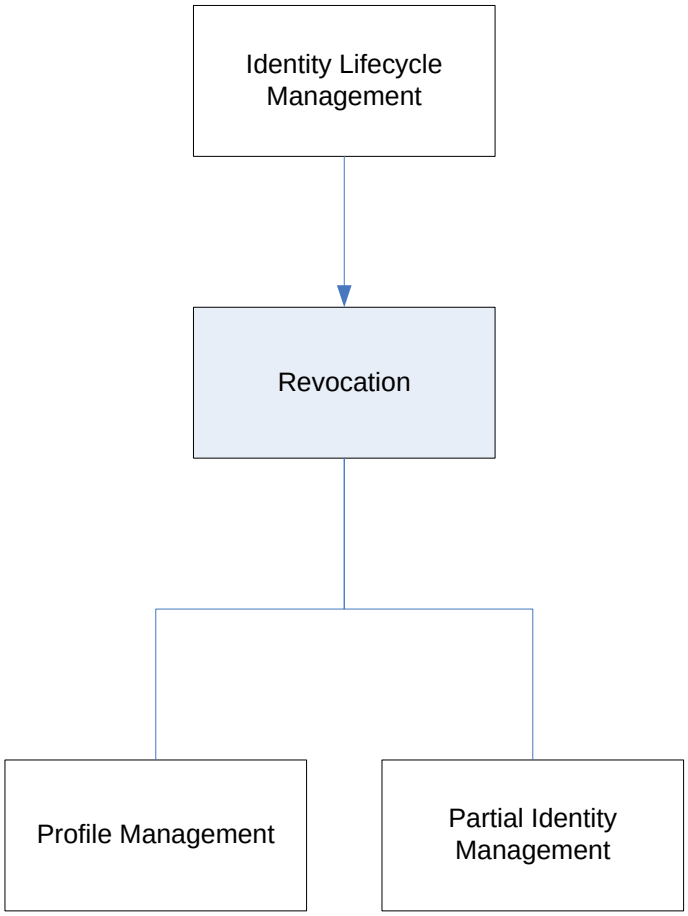


Figure 62 – Revocation

9.7.12 Sub-community Management

History: *Component contributor:* ITO

PICOS Principle (PP): 19

PICOS Feature (PF): 7, 10

9.7.12.1 Purpose

The *Sub-community Management* component is responsible for managing sub-communities created by a partial identity.

9.7.12.2 Description

The *Sub-community Management* component addresses how members interact with external and sub-communities. Every partial identity is a member of the PICOS community or sub-community.

External communities (inter-community) and sub-communities (intra-community) may initially appear very different, but the issues of trust and privacy that each experience are very similar. They concern access rights, which are derived from a member's profile and privileges, assigned when they registered with the community.

The role of the *Sub-community Management* component is to facilitate the integration of an external community or a sub-community into a member's profile. In the case of a sub-community, in response to the request from the member to create a sub-community, the *Sub-community Management* component will build the community and assign ownership to the member. It will set up all monitoring and any other services that the sub community requests. Since an external community already exists, there is no need to create anything, but the monitoring process (which occurs within the local community) will need to be initiated.

Sub-communities maintain a list of members who can access the sub-community (in its profile) using the *Profile Management* component.

A useful description of how sub-community management is employed in PICOS can be found in Section 13 in:

- PICOS Use Case 9: Sub-community

9.7.12.3Dependencies

Components that this component calls	Purpose
Profile Management	To set up and manage the profile of a sub-community.

Components that call this component	Purpose
Service Selection	To administer sub-communities.

9.7.12.4Drawing

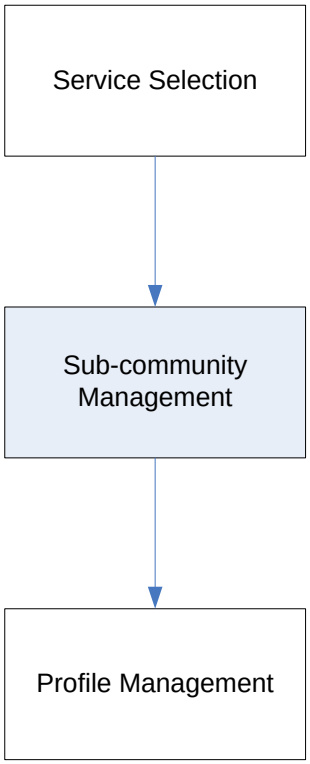


Figure 63 – Sub-community Management

9.8 Content Handling

The Content Handling component group contains the following components

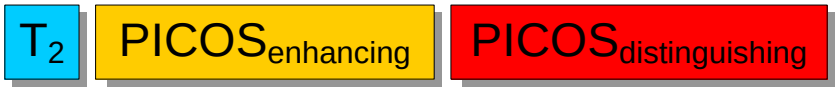
9.8.1 Tier-1 Content Handling components

None

9.8.2 Tier-2 Content Handling components

- Content Sharing
- Data Minimisation
- DRM
- Linkability
- Non-repudiation
- Secure repository

9.8.3 Content Sharing



History: *Component contributor: GUF,* ITO

PICOS Principle (PP): 4

PICOS Feature (PF): 2

9.8.3.1 Purpose

The *Content Sharing* component is responsible for making imported content available to members.

9.8.3.2 Description

The *Content Sharing* component contributes, administers, manipulates and communicates content imported by one member to other members. The contributing member can control who can see the content, using the tagging and privileges, and indirectly by role and context. Members can be notified that new content is available using the *Notification* component.

Content sharing is triggered automatically when a member imports content, using the *Importer/Exporter* component, or on demand using the *Service Selection* component.

Content is shared under a partial identity of the member to help maintain privacy.

The function of the *Content Sharing* component permit:

- Contribution
  - Upload content elements
  - Publishing content on the personal profile (e.g. pictures in a picture album)
- Administration
  - Managing content: Organising personal messages; Organising picture and video albums; Deletion of content elements
  - Setting and managing policies for restriction of the access to content by other users.
- Manipulation
  - Editing content elements
  - Editing tags for content elements
- Communication
  - Publishing content on the personal profile
  - Forwarding content elements to other users

- Attaching content elements to messages, forum posts, etc.

Content is shared with other members in one of three ways:

- Member to member
- Member to sub-community
- Member to community

Other possibilities exist, e.g. community to community, and member to component, i.e. reputation, which are not discussed further.

A useful description of how content sharing operated in PICOS can be found in Section 13 in:

- PICOS Use Case 7: Content Sharing

9.8.3.3 Dependencies

Components that this component calls	Purpose
Notification	To notify other members that new (or changed) content is available.
Privilege Management	To control which members can access content.
Profile Management	To check if a member wishes to be identified when sharing content.

Components that call this component	Purpose
Importer/Exporter	To share imported content.
Service Selection	To share content.

9.8.3.4 Drawing

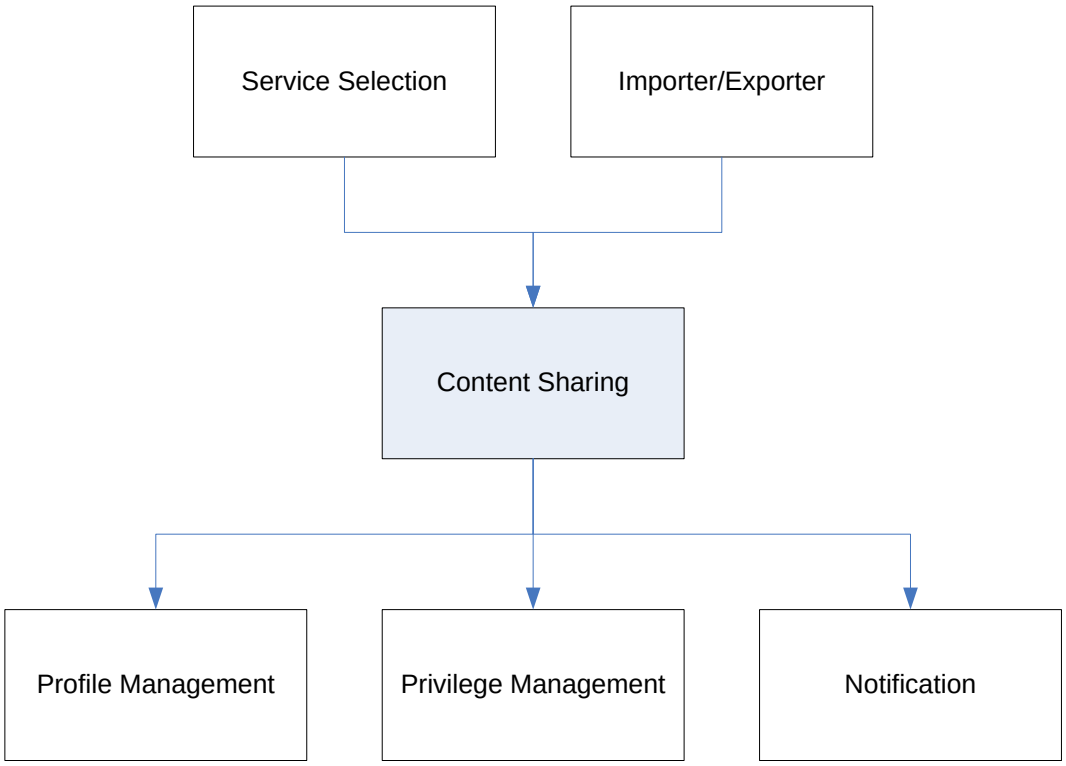


Figure 64 – Content Sharing

Data Minimisation

History: *Component contributor:* ATOS

PICOS Principle (PP): 3, 8

PICOS Feature (PF): 2

9.8.3.5 Purpose

The *Data Minimisation* component minimises the unnecessary exposure of member personal information.

9.8.3.6 Description

Data minimisation is one of the main building blocks on a PICOS community. The objective is for a member to never be required to provide more personal information than is absolutely necessary to gain access to a service. Data minimisation is the primary method for providing privacy and protecting identities[21].

Strictly speaking, data minimisation minimises the collection of data regardless of its purpose (going beyond the "Collection Limitation" Privacy Principle[22]). Additionally data may be requested if it is clear to the individual that it is optional and there is clear justification for collecting it, or when collection is an obvious and agreed benefit to the member.

PICOS uses the *Data Minimisation* component to support members in minimising the information they provide, following the principle of 'minimal disclosure of information'[23]. Thus, data management is a core feature of identity management within PICOS, since data minimisation ensures that data is accurately managed in a context, in accordance with applicable policies, standards and regulations.

In the scope of community operations, only data relevant and necessary will be requested and transmitted, except for data that members consciously and willingly choose to share with each other and/or the community.

The design of data processing systems considers non-identifiable interactions and transactions by default and, wherever possible, identification, observation and linking of personally identifiable information is minimised.

Often members are not aware which kind of information has to be provided in order to use a certain service. Consequently they often either give too much information or refuse to use the services. By

---

[21] See PICOS D2.3 Contextual Framework, p.29

[22] Privacy principles represent a basic set of overall commonalities in the fundamental privacy requirements to prevent the misuse of personally identifiable information when processing it in information and communication technology. See ISO/IEC JTC 1/SC 27 N56734 "A Privacy Framework" p.16.

[23] See PICOS D2.4 Requirements, section 3.1.2, p.56.

providing guidance to member, PICOS helps member understand their options in the context of their own actions. Furthermore, the system supports members' right to be informed before the processing of data starts and allows rectifying, erasing, or blocking their data. On the other hand, community providers tend to collect data ahead e.g., for statistical or advertising reasons. Therefore, it is necessary to constrain them to collect only data that is needed to provide respective information or services.

In addition, members have very different values, and some member may want to publish/share more personal data than others, whether sharing information with other members, with the community provider (e.g. for using specific community services) or with third parties (e.g. for marketing purposes). For this reason PICOS provides members with the option to manage their own level of data minimisation, e.g. giving them the option to choose what information they share and with whom[24], while showing in an easily understood manner how their information will be held, taking into account the context in which information will be handled.

Data minimisation can be implemented in the client side and/or the server.

[24] Users may also choose to make use of anonymisation / pseudonymisation services and other mechanisms to further protect their personal information, even after deciding to share more personal data (see section 7 of ISO/IEC JTC 1/SC 27 N6736 "A Privacy Reference Architecture"

9.8.3.7 Dependencies

Components that this component calls	Purpose
None define at present	

Components that call this component	Purpose
External Service Delivery	To minimise information exposed.
Identity Translator	To minimise information exposed.
Privacy Advisor	To determine options to minimise information exposed.

9.8.3.8 Drawing

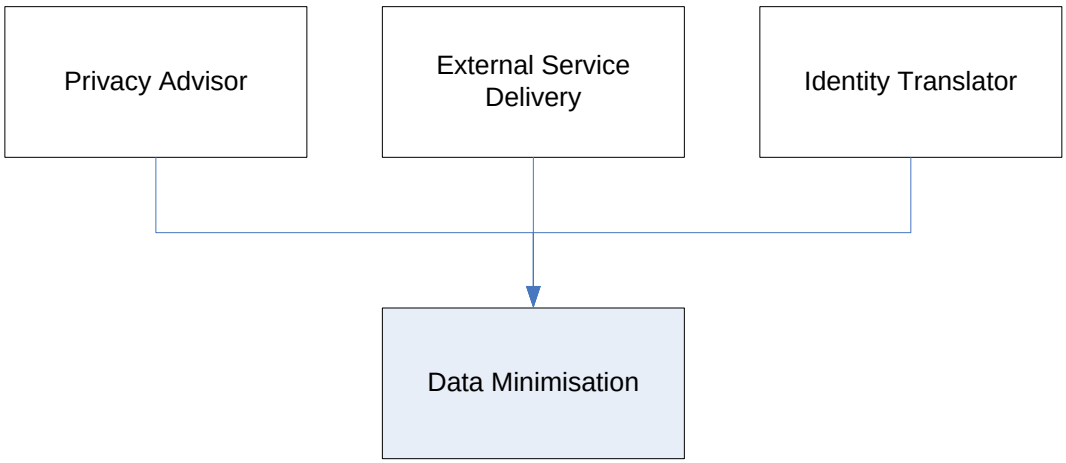


Figure 65 – Data Minimisation

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

9.8.4 DRM

History: *Component contributor:* HPF

PICOS Principle (PP): 2, 10, 23

PICOS Feature (PF): 2

9.8.4.1 Purpose

The *DRM* component ensures that content is used in accordance with the terms and conditions of the owner.

9.8.4.2 Description

The *DRM* component is associated with the *Access Control* component. It ensures that content on the community portal – whether professional content or user-generated content - is accessed (viewed, downloaded, shared) in accordance with the terms & conditions defined by the content owner or community operator.

If the content owner or community operator specifies certain usage terms (e.g. which devices can be used to view content, who can the content be shared with, how many times can the content be accessed, etc), this function ensures that those policies are respected.

Although it is possible that PICOS will manage DRM as an external third party provided service, some level of DRM functionality may need to be provided within the community, to the extent that enforcement is possible.

The DRM component acts as an interface that allows content owners, community members and community operators to set additional policies using the *Policy Management* component that are community-specific (e.g. defining community-wide licenses, community-wide usage conditions, etc).

For example, if a content owner provides a usage license for a community, some interaction is required between the DRM system (which enforces the rights) and the community platform to share information on who is currently a member of that community.

DRM may also be provided at the client.

9.8.4.3 Dependencies

Components that this component calls	Purpose
Access Control	To control access to content.

Components that call this component	Purpose
Consent Management	To enforce member-specific DRM.
Policy Management	To enforce community-wide DRM.

9.8.4.4 Drawing

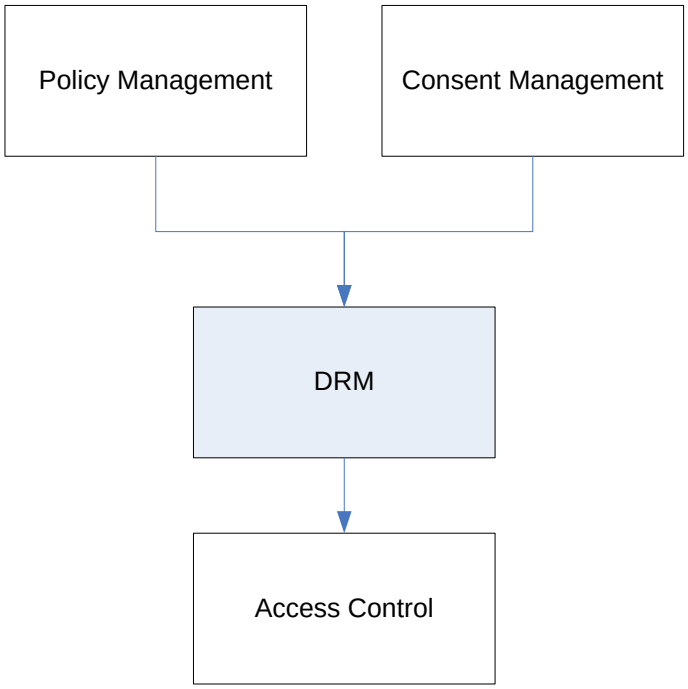


Figure 66 – DRM

9.8.5 Linkability

History: *Component contributor:* BRNO

PICOS Principle (PP): 2, 3, 8, 11, 18

PICOS Feature (PF): 1, 2, 10, 15

9.8.5.1 Purpose

The *Linkability* component determines if a link can be established between independent events.

9.8.5.2 Description

Linkability is the ability to link together pseudonyms, posts or other events performed by a single entity (member). The main contribution of the *Linkability* component is a profile of activities performed by a single entity. The more sources of information available, the more information can be linked together and this has a direct impact on the quality of the profile. Well created member profile may contain very sensitive information.

Providing anonymity and unlinkability increases member privacy. However, identification and linkability provides easier accountability/non-repudiation and tracking. Between the two extremes of complete unlinkability and easily visible linkability there is a range of options where events can be only linked with additional knowledge or with the parties collaboration of one (though typically several) trusted party.

The Linkability component provides two roles:

- Resolving linkability: When required, this component will assemble the necessary information to prove a link between a member and contributed content, or identify a member from contributed content
- Providing unlinkability: Several techniques exist that provide unlinkability, but they are generally designed for specific situations, e.g. message transfer, group signing (where the actual signer does not want to be identified other than a member of the group. For each situation, a range of actions will need to take place, and the role or the Linkability component is to co-ordinate the various components and service that will provide the solution, e.g. key management, cryptography, authorisation.

At present the *Linkability* component will access the *Privacy Adviser* component in determining where excessive linking is taking place. It is possible that the *Linkability* component can assist legislation, non-repudiation, anonymity and accountability.

9.8.5.3 Dependencies

Components that this component calls	Purpose
None defined at present	To control access to content.

Components that call this component	Purpose
Privacy Advisor	To determine the extent of linkability.

9.8.5.4 Drawing

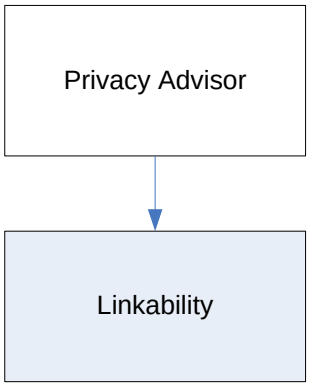


Figure 67 – Linkability

9.8.6 Non-repudiation

History: *Component contributor:* UMA

PICOS Principle (PP): 1, 12, 14, 23

PICOS Feature (PF): 15

9.8.6.1 Purpose

The *Non-repudiation* component provides a non-reputable binding.

9.8.6.2 Description

In order to provide a suitable support for accountability within PICOS, all members' contributions should provide a (direct or indirect) non-reputable binding with the originating member.

This *Non-repudiation* component adds a non-reputable binding to all content that is contributed to the community. Alternatively, it can provide optional 'on demand' binding or component and originator ID to yield the corresponding non-reputable binding.

A further option is for the contribution to be digitally signed at the client using the private key of the contributing member, possibly the same key that the member uses when authenticating themselves to the community. In this way the community can verify the authenticity of the contributing member and the content before acceptance.

9.8.6.3 Dependencies

Components that this component calls	Purpose
Cryptography Key Management	To provide cryptographic primitives, e.g. digital signature.

Components that call this component	Purpose
Content Sharing	To provide provenance of contributed content.
Event Logging	To maintain integrity of event information.

9.8.6.4 Drawing

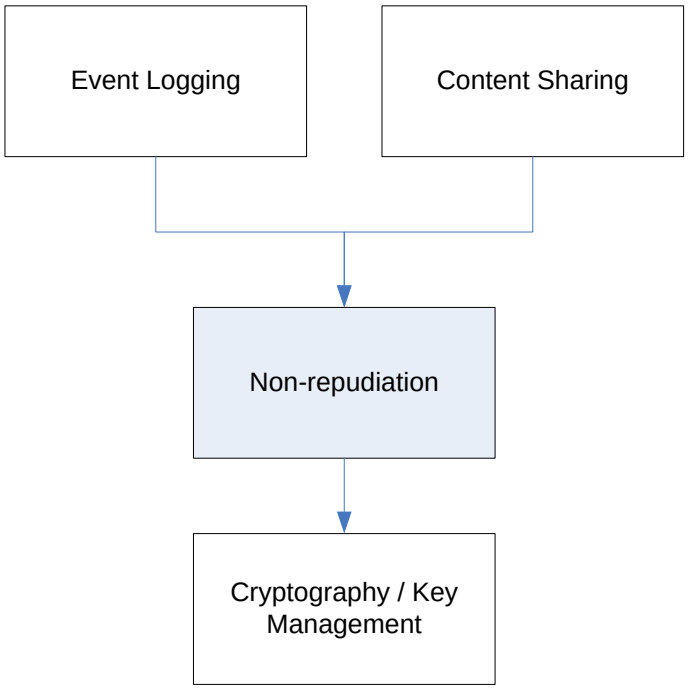


Figure 68 – Non-repudiation

9.8.7 Secure repository

History: *Component contributor:* TMO

PICOS Principle (PP): 4, 10

PICOS Feature (PF): 2

9.8.7.1 Purpose

The *Secure Repository* component provides a safe location to store personal sensitive information.

9.8.7.2 Description

The *Secure Repository* component provides a safe place to store content and security-related data (e.g. keys) on the client or in the community. Content can be text, audio or video contribution by a member.

All member data needs to be stored in a secure location, i.e. encrypted on the client or community. The reason for this is to protect member data in case the client device is lost or stolen, and to prevent others gaining unauthorised access to private sensitive data.

The *Secure Repository* component interacts with the storage medium to store and retrieve content. In one mode, the Encryption/decryption takes place within the component so that content transferred to the medium is protected. Therefore the component will need to be provided with appropriate keys.

Alternatively, content can be provided to the component in encrypted form.

Where the component provides the encryption, it will only do so after receiving a valid authentication from the content owner. Authentication is achieved with support from the *Access Control* component and *Authorisation* component.

9.8.7.3 Dependencies

Components that this component calls	Purpose
Cryptography Key Management	To provide cryptographic primitives, e.g. digital signature.

Components that call this component	Purpose
Access Control	To retrieve identity information.
Authorisation	To retrieve authentication information.
Cryptography Key Management	To provide cryptographic primitives, e.g. digital signature.

9.8.7.4 Drawing

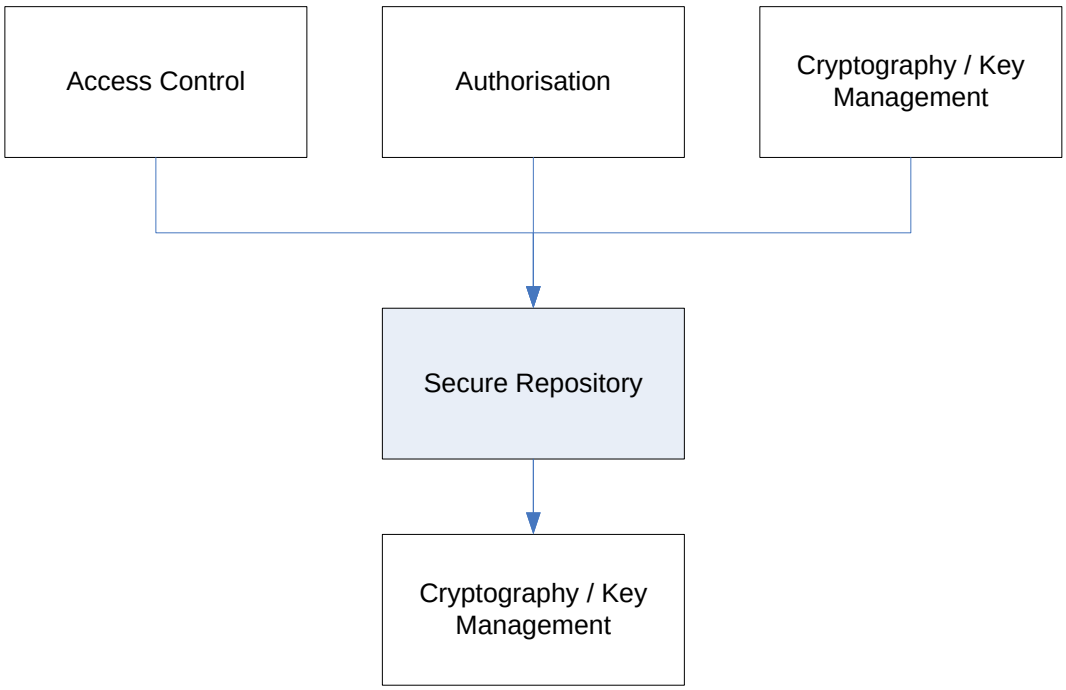


Figure 69 – Secure Repository

10 PICOS Toolbox

T0 T1 T2 PICOS distinguishing PICOS research

History: *Component contributor:* All

PICOS Principle (PP): All!

PICOS Feature (PF): All!

10.1 Purpose

10.2 Description

The principal objective of the PICOS project is to develop an open, privacy-respecting, trust-enabling identity management platform that supports the provision of community services by mobile communication service providers. The project addresses the tension that arises when a community carries marketing activities initiated by third party sponsors/advertisers.

Within the PICOS project, mobile communication services are defined as services that can be used from mobile terminals as well as from fixed line terminals. The PICOS platform includes:

- Data model that contains the identity/credential information
- A Toolbox of components that provide identity management functions
- Data flows between components and the protocols they use.

10.3 Service Toolbox

The toolbox can therefore be seen as a set of API's within a SDK, which provides community developers and community services developers a rich set of functionality derived through tailored components. Members and community providers are free to choice which services best suit their community model, though the interfaces provided. The Toolbox provides interfaces to services and functionalities according to the community's needs, with respect to privacy, trust and identity management in order to allow application developers to build community-specific solutions. It also offers value-add customisable services that combine information, context and transaction information to create privacy-respecting mash-ups (workflows), subscription and planning management. The aggregation of different services should be transparent to members.

The toolbox, which is implemented as a WP5 deliverable, presents new interfaces to network/services-based routines, as well as member/operator management portals. In combination, the Toolkit provides a business support system. It represents a technical architecture, and the basis for designing and building a PICOS identity management platform, able to addresses different classes (trust models) of communities.

A primary focus for the Toolbox is community designers. The SDK and API provide the building blocks from which a PICOS community is created. Community designers will be able to create and customise their community, and associate appropriate privacy management and trust management features. The Toolbox will allow them to bind information services and communication services to their community, while respecting the overall objectives on privacy and trust.

Design of the Toolbox begins in D4.1, and continues through to D4.2.

The functionality of the PICOS platform is exposed via a set of open APIs and value-add building blocks that facilitate the development of community-specific applications. Each application is adapted to the needs of a particular community. The Toolbox allows developers to quickly bundle and customise the features of the PICOS platform for the target community.

The APIs that the PICOS platform exposes should ideally be based on open web standards, and include community administration functions, privacy, IdM, data administration and trust. They may also be exposed via a web interface, allowing easy administration from any web browser.

10.4 Service Composition

The system should support merging of public, personal and sensitive data to enable complex, trusted services to be delivered. Such merging must be with the knowledge and agreement of data owners, who retain the right to withdraw their consent at any time.

Service composition allows for building composite services by combining existing elementary or complex services, possibly offered by different third parties. For example, where a Taxi Driver has been asked to pickup a child from school, parents would be happier if they could track the taxi in real-time and see an audit trail that confirms that the driver and child were in close contact at the prescribed time. Knowing that information about regular school pick-ups is protected, and being able to obtain reputation information about the driver, would engender trust.

10.5 Application Orchestrator

The application orchestrator helps non-PICOS developers organise (orchestrate) the PICOS services offered by the PICOS Toolbox. It assists by allowing them to build community-specific solutions in a flexible and customised manner, according to their needs with respect to privacy, trust and identity management.

Service composition allows for building composite services by combining existing elementary or complex services, possibly offered by different third parties. The Application Orchestrator also helps designers and operators combine local services to achieve complex PICOS functionalities without necessitating the use of external community applications which may be considered less trustworthy.

11 PICOS Client

History: *Component contributor:* ATOS

PICOS Principle (PP): 4, 9, 17 (as a minimum)

PICOS Feature (PF): 15 (as a minimum)

11.1 Purpose

11.2 Description

The PICOS architecture consists of a combination of client-based and network-based components, which offers generic functionality that can be combined to build community-specific solutions.

The PICOS client may be fixed or mobile. In the case of a mobile community, it is anticipated that the mobile device will be a smart phone. Thus, services may run in the PICOS client.

Note: for the first prototype that WP5 and WP6 will create, and to which D4.1 is closely aligned, the extent to which the client will run local functionality is limited to simple web-based access and possible client device authentication. In D4.2 further opportunities will be explored, including those required to support a stronger trust model.

11.2.1 Client connectivity

The client must be able to:

- Connect to the community/PICOS platform and establish (access) a range of services – possibly web-based – that PICOS offers, e.g. reputation, identity management, profile management, etc.
- Support instant communication between members, e.g. chat, IM.

For example, using the SIP protocol that is widely deployed by mobile network providers, a possible configuration may be:

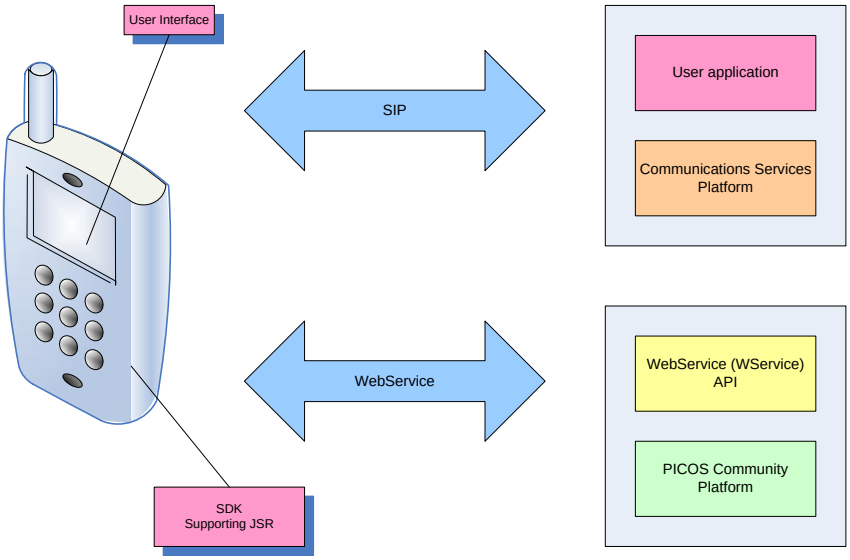


Figure 70 – PICOS Client

11.2.2 Client functionality

A typical mobile client is shown below. The lower green component is the Software Development Kit (SDK), which presents several client platform APIs, e.g. Security, SIP, Web Services.

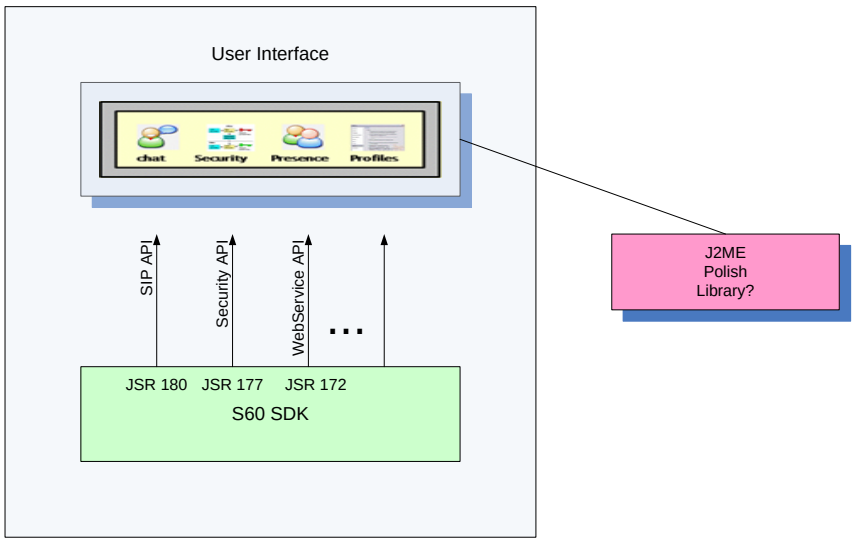


Figure 71 – PICOS Client functionality

12 Overall PICOS architecture

The PICOS architecture is complex and dependent on a large number of inter-related components. The easiest way to understand its operation is to study the use case. However, we have attempted to show in a single diagram how the architecture looks.

At is simplest level it consists of components that support access to the community and 2) components that deliver a range of services. The architecture is essentially services-based.:

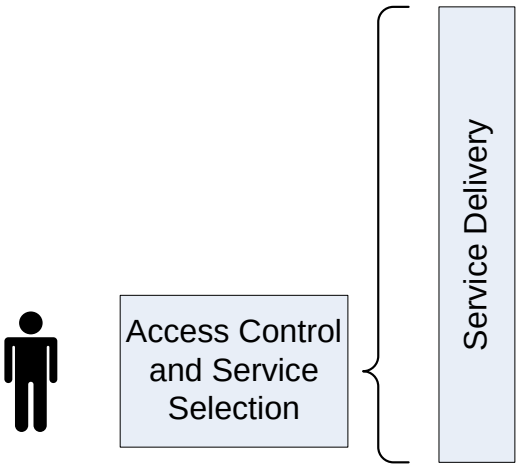


Figure 72 – High level architecture visualisation

The layout is reflected in the overall diagram, which is shown below. To improve legibility, some of the comments appear more than once in this representation.

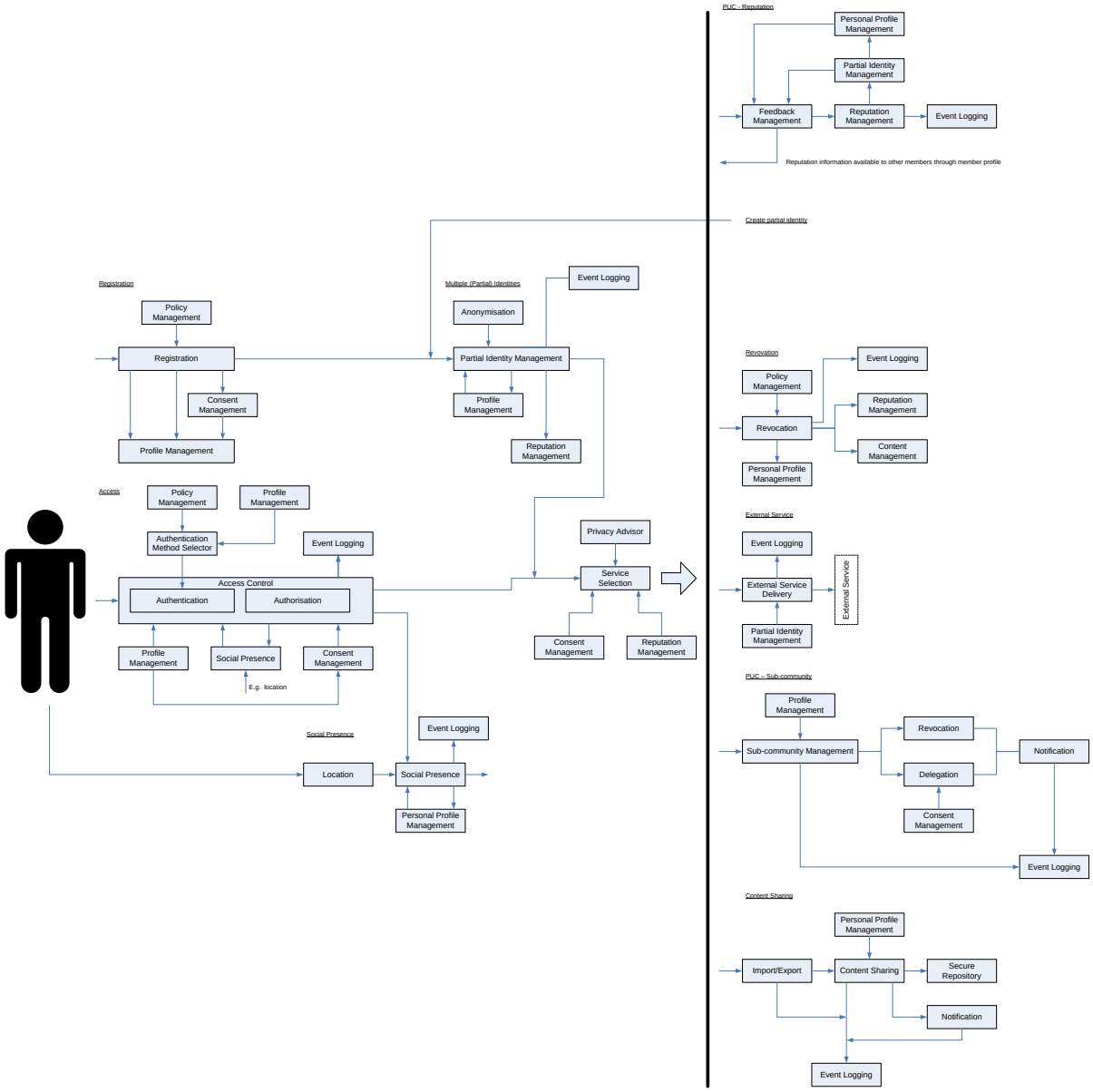


Figure 73 – Overall architecture diagram

13 PICOS Use Cases

To illustrate how the architecture solves practical day-to-day situations which our target community may encounter, and to indicate the interaction between components, we have identified 15 PICOS Use Cases (PUC)[25]. (Note that the order of this list does not imply any priority.)

Each use case has been selected because it satisfies an element of the Scenario presented in Section 3, and because from experience we know that the trust and privacy issues that PICOS will raise requires a detailed level of understanding of interactions.

Of these 15 PUCs, we now examine the first nine so as to describe the interaction between components in detail. These are considered higher priority PUCs because they cover the application of the core privacy and trust management components. The remaining PUCs, and any additional PUCs that arise, will be examined beyond D4.1 when a better understanding of the interaction of these first nine components is gained.

- **PUC 1: Registration**: Registration and creation of a new member profile. Creating an initial identity, importing reputation, setting policies and respecting different roles.
- **PUC 2: Accessing the community**: Identifying, authenticating and granting authorisation to a member. Selecting a service.
- **PUC 3: Revocation**: Leaving a community, giving due consideration to content contributed while a member.
- **PUC 4: Multiple partial identities**: Creating, selecting and managing multiple member identities (pseudonymous/partial identities).
- **PUC 5: Reputation**: Establishing the reputation of members within and across communities. Providing recommendation and feedback. Registering to receive notifications.
- **PUC 6: External services**: Exposing partial identity / profile to external services
- **PUC 7: Content sharing**: Importing/exporting and controlling the sharing of content contributed to the community by members, including automatic/manual tagging and notification.
- **PUC 8: Presence**: Setting and controlling the sharing of online status information (location, presence, etc.) about members.
- **PUC 9: Sub-community**: Creating and managing a sub-community (sub-group) within the overall community.

(Note: The following six Use Cases are not examined further in this deliverable.)

- **PUC 10: Community reputation**: Check and validate the reputation of a community (prior to becoming a member). Establishing, and making publicly available, the reputation of a community, for the use by new members considering joining the community.

[25] The remaining Use Cases will be examined, and possible further use cases created, in deliverable D4.2.

- **PUC 11: Searching**: Searching for and establishing contact with other members within the community, who share similar interests.
- **PUC 12: Offline working**: Enabling member to benefit from community services when offline (typically mobile)
- **PUC 13: Feedback**: Providing recommendations / feedback
- **PUC 14: Real-time communication**: Allowing member to interact one-to-one, thus sharing content in real-time.
- **PUC 15: Audit**: Audit, correcting errors, removing privacy-violating content

The purpose of each use case is to illustrate the role and functionality of components in delivering the stated service to the members or the operator. The choice of the nine use cases is motivated by a desire to demonstrate the breadth of functionality that the PICOS community can provide. However, this choice should not be seen as an indication of the priority that the target community might place on the functionality delivered.

For each use case we describe the situation, and then 'walk through' the sequence of interactions, between member and component, or between component and component, to illustrate the process. We also include a simple reference diagram, which shows the key component required to address the situation. (Each component is shown as a box which contains all the functionality described in the component description). The described use cases cover 30 of the 49 defined components. For those components not described in the use case, their relationship with other components can be seen in the diagram included with each component description.

Some components occur in almost every use case, e.g. Event Logging, Audit and the Secure Repository. Because they provide ancillary functionality, it is easy to overlook the importance that they play in the overall architecture. It is also easy to miss that the central role that these components play can introduce a point of weakness for privacy and trust management. For example, the Event Logging components receive information that is potentially very sensitive. It is also well placed to compromise anonymity because of the ease by which links can be made between entries, and therefore partial identities and transactions. We recognise these and other risks, and will seek appropriate solutions in the implementation.

13.1 PUC 1: Registration

13.1.1 Situation

All members of the community must be registered if they wish to have full access to the facilities on offer. Guests, who may be considered anonymous members, are an exception to the registration rule, since registration is not necessary, but the range of services available to a Guest is severely restricted. In order to gain access to the full range of services, registration is necessary.

Registration provides the community operator with assurance that a member is suitable to join the community. This assurance may be through prior knowledge of the member, or through evidence that minimises risks to the community (e.g. the real name and address of the prospective member, which provides a route to compensation if required).

Obviously, new 'unknown' members must be able to join the community, and for this PICOS offers an 'application process' (e.g. application form) where applicants provide information requested by the community operator and are granted membership if they meet the requirements set by the community operator or elected member representatives.

Every member is allocated a root identity. The root identity is only used for registration and to link subsequently created partial identities. The root identity is only known to the community operator, and is linked to the registration information (which in most cases will ultimately be a real name and address).

In order to interact with the community, each member must create a partial identity (a pseudonym). They can create any many partial identities as they feel they need, but they must create at least one. Creation of the first partial identity would normally take place at the time of registration, but can occur later.

Every partial identity is allocated a feedback pseudonym. The feedback pseudonym enables members to provide feedback to the community without revealing their partial identity.

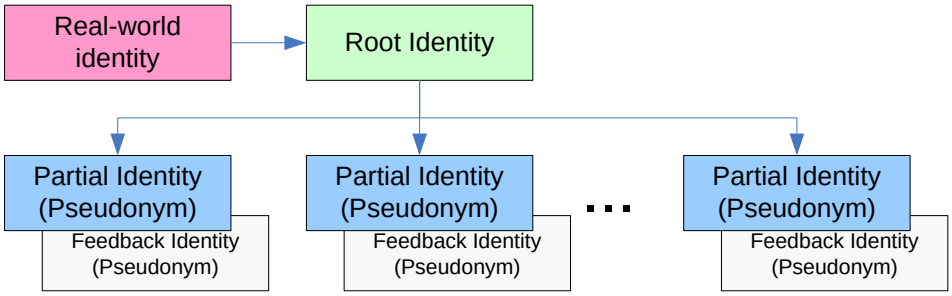


Figure 74 – Root and Partial Identity overview

To summarise, every member has one root identity which is assigned when they register with the community. They are immediately allocated a partial identity when they first interact with the community. All identities, including root, have a profile, but only partial identities have feedback

identities. A member can only have one root identity. Roles other than member, e.g. community administrator, are treated as special cases.

Optionally, a member can create further pseudonyms at the time of registering. Just like the first partial identity, each subsequent partial identity is assigned a feedback pseudonym[26].

Pre-registered identities are also accepted by the community. For example, a member who is already registered with a mobile phone provider may be permitted to use the allocated identity as their community root identity, so long as the member consents to this. This situation is particularly useful where a community is hosted by the telecoms provider.

13.1.2 Reference diagram

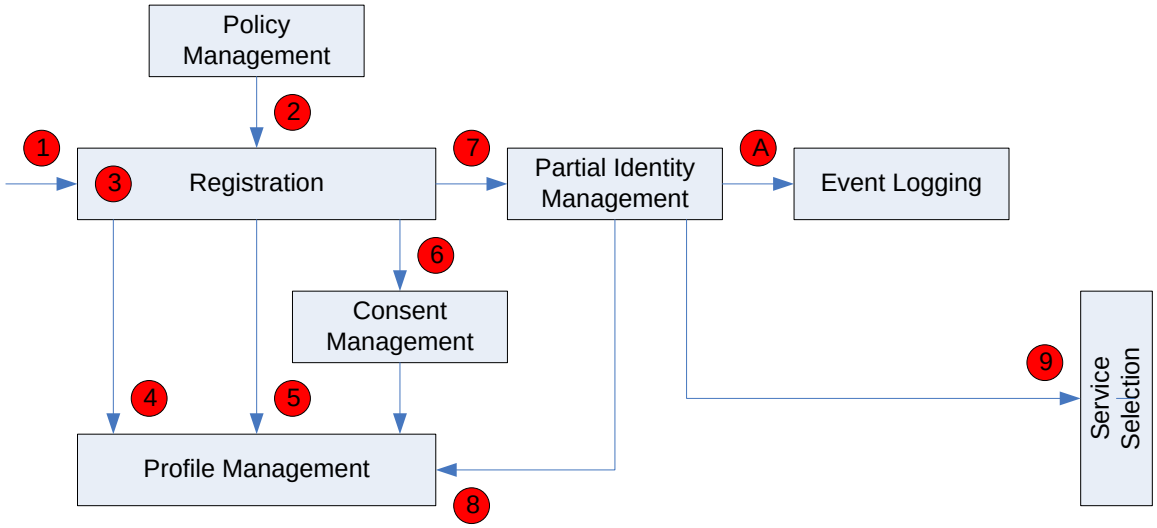


Figure 75 – PUC 1: Registration

13.1.3 Walk-through

A prospective member approaches the community and presents to the *Registration* component {1} evidence of one or more of the following[27]:

- prior registration with a real-world community
- prior registration with a mobile phone network provider
- prior registration with a 'partner' community
- another acceptable credential as defined by the community policy

[26] The diagram shows two levels of identity (Root and Partial) where all partial identities are directly linked to the root identity. We recognise that partial identities may also spawn partial identities, creating a tree-like structure. To date we have not seen a need for this approach, but we keep it in mind should a need arise.
[27] We recognise that other forms of registration may be possible and conclude that this list is not exhaustive.

- successful completion of the application process (e.g. application form)

The choice of what is considered an acceptable form of registration is notified by the *Policy Management* component {2}.

The *Registration* component assigns a root identity {3}, and records this alongside the evidence that the member provided in the member's profile using the *Profile Management* component {4}. Registration also assigns a default role and privileges to the new member {5}.

Once registered, the member can set certain elements of their personal profile to help manage privacy using the Consent Management component {6}.

Following registration, the member creates their first (and possibly only) partial identity, which they subsequently use to interact with the community {7}. Every partial identity has a unique profile which holds information about the new identity, e.g. reputation {8}. (The member can also set consent preferences for the new identity, but for clarity this is not shown in the diagram. See PUC covering Multiple (partial) identities.)

Once a partial identity has been assigned, the member can access the service {9}.

All actions performed are logged by the *Event Logging* component {A}.

13.1.4 Reference to the User Scenario in Section 3

Before John can access the services he must first register. This involves John providing supporting evidence to authenticate his identity, some of which is personal information. The evidence that John sends is actually a credential issued by a governmental fishing authority.

13.2 PUC 2: Accessing the community

13.2.1 Situation

Having first registered with a community, a member can then access the community to call on the services offered by the community.

Accessing the community involves presenting a partial identity and supporting authentication information. Once authorised, a member is able to select the required service from those provided by the PICOS community application. The process of accessing the community assigns pre-set privileges to the member and can reveal their status (presence) to other members subject to the member giving consent.

13.2.2 Reference diagram

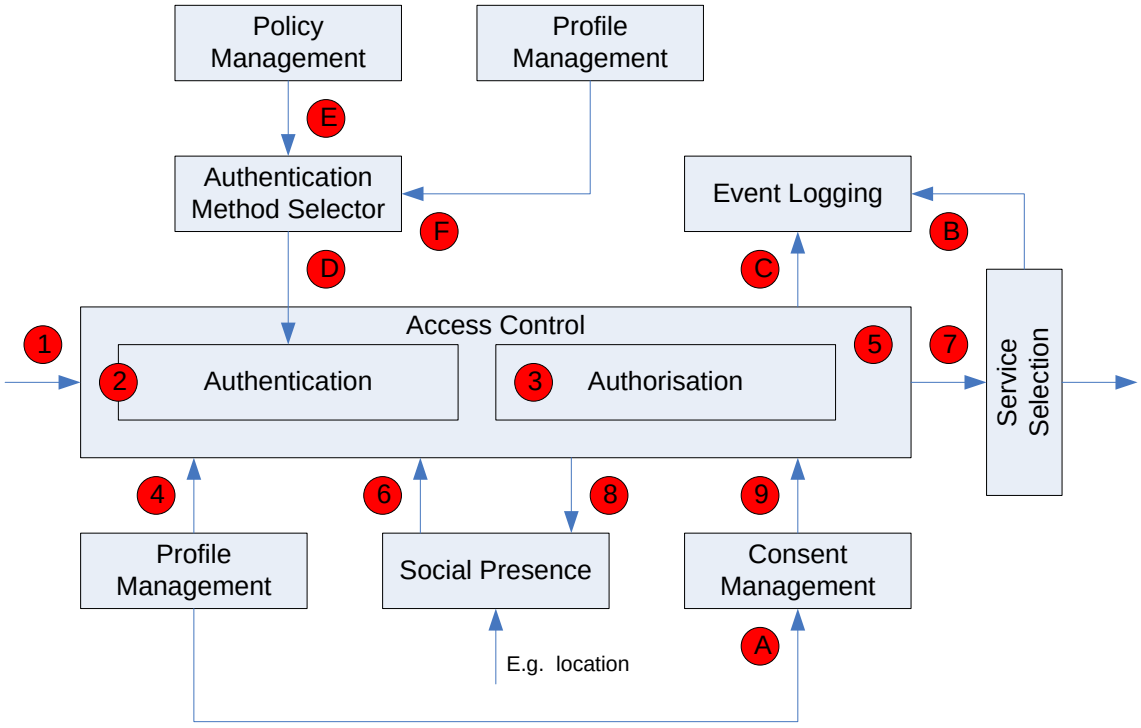


Figure 76 – PUC 2: Access control

13.2.3 Walk-through

The member presents their partial identity via the Access Control component {1} and authentication information to the *Authentication* component {2}. The method of authentication is indicated by the Authentication Method Selection component {D}, which takes input from the community policy and the member profile, via the Policy {E} and Profile Management {F} components respectively. This

means that either the community can dictate the method of authentication for all members, or each member can select a preferred method. The community policy would typically indicate a range of acceptable authentication methods, leaving the member to choose the preferred method, perhaps taking into account data minimisation concerns. If accepted, they are authorised {3} and allocated privileges by the *Personal Profile Management* component according to their role {4}.

Member privileges dictate the access that a member has within a community. The *Access Control* component determines the set of services that the member can access {5} by checking the member profile {4}, taking into account any context information like social presence {6}. Available services are then presented to the member {7}. At the same time, a member's social presence is updated to show to other members that they are active within the community {8}, assuming that they have given consent as defined by the Consent Management component {9} and available through the Profile management component {A}.

All actions performed are logged by the *Event Logging* component {B}{C}.

13.2.4 Reference to the User Scenario in Section 3

After registering, John accesses the community. He previously created a profile that defines personal attributes (e.g. buddy list) and personal interests. The profile also let John set preferences that define who can see his personal status (presence).

13.3 PUC 3: Revocation

13.3.1 Situation

Revocation occurs when members leave a community for the final time. This may occur through choice or because the community has terminated the member's membership. In addition to denying the member further access to the community, the community operator must take action with respect to content that was contributed to the community during the lifetime of the membership.

Content is handled in several ways, depending on the policy set for the community:

- Removed from the community (Probably default position)
- Retained in the community for other members to (continue to) access

There are multiple reasons for retaining content: 1) legislation dictates so, 2) content is still useful to the community (e.g. reputation), 3) content can be transferred to another member or the community operator.

Where content is retained by the community, the most likely action will be to pseudonymise the identity of the contributing member. This will satisfy both legal requirements and member revocation preferences. The decision to 'anonymise' or destroy may depend on the nature of the content. For example, personal data may need to be destroyed in order for the community to comply with EU Data Protection legislation. It may be acceptable for other content to be destroyed.

There is a possibility that a member may choose to return to the community. If this is a concern, then the community policy will state whether reputation information (and possible other content subject to Data Protection Law and the need to only retain information that is relevant and not excessive), and presumably the real identity of the member to who it related, can be retained. Options are that reputation is deleted or frozen/suspended.

We have already mentioned that event and audit services require special attention if privacy it to be maintained. If retained, or removed and archived, content may need to be anonymised.

The process for deciding which of the above options is appropriate will depend on the needs of the individual, the community and legislation, as stated in the community policy.

13.3.2 Reference diagram

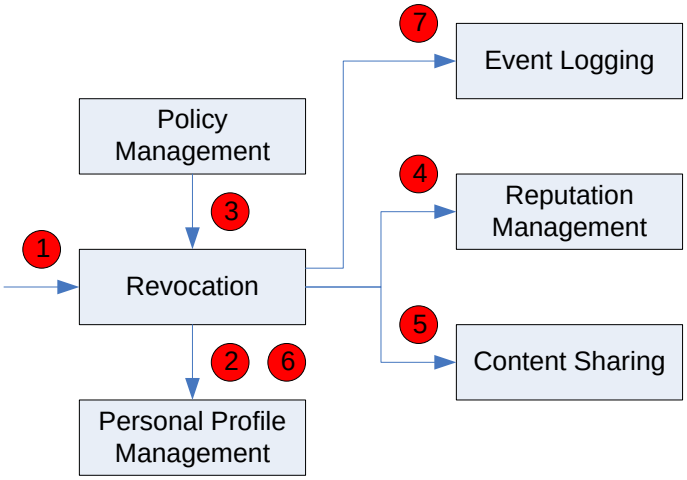


Figure 77 – PUC 3: Revocation

13.3.3 Walk-through

The *Revocation* component is responsible for terminating membership. Revocation would normally be initiated by the community operator {1}.

The first step in the revocation process is for the member profile to be set to deny further access to the community by the member {2}. This may affect the root identity and all partial identities, or just the partial identity in question. Thus authorisation will fail if the terminated identity attempts to regain access.

According to community policy, as communicated by the *Policy Management* component, content for each and every identity affected by the termination will be either destroyed, archived or anonymised {3}. Responsibility for this decision lies with the *Revocation* component, which will trigger appropriate action {4} {5}.

The feedback pseudonym, which would have been created if the member provided feedback from their root and/or partial identities, can remain and will ensure that feedback is present for future inspection by members. The profile of each feedback pseudonym will show 'membership terminated' {6}. The profile of the member may also show the reason for leaving community, e.g. 'resigned', 'terminated', 'excluded'. A member who leaves the community voluntarily would have a different reason recorded than a member whose membership was terminated by the community manager due to (say) bad behaviour. The reputation might also be downgraded if the member was asked to leave the community, possibly to a special reputation value, e.g. a dash/hyphen.

All actions performed are logged by the *Event Logging* component {7}.

13.3.4 Reference to the User Scenario in Section 3

When John decides to leave the community he cancels his membership, but history of his membership and messages he has posted remain.

13.4 PUC 4: Multiple partial identities

13.4.1 Situation

In addition to partial identity created when a member registers with the community, a member may create one or more additional partial identities (or pseudonyms). The reason that this facility is provided is to allow a member to represent themselves in different ways within a sub-community, or across multiple sub-communities. It means that each partial identity can potentially gain a different level of respect (reputation) depending on their interaction with the sub-community(ies). For example, a member may be an expert on fly fishing, and thus willing to provide advice and help to others using one partial identity. However, they may also be just learning about sea fishing and do not want to present the same partial identity for fear that it may lessen the respect that they have in the fly fishing community. In such a case, the member chooses the most suitable partial identity depending on whether they are interacting with the fly fishing sub-community or sea fishing sub-community.

Each partial identity can be used to access one or more sub-communities. For example, a member may have three partial identities that each access the same sub-community, or they may have five partial identities that access five sub-communities (one partial identity per sub-community). Other permutations are equally possible. Essentially, any partial identity can access any sub-community(ies) assuming that the owner (creator) of the sub-community grants permission.

Each member partial identity appears to all other members as a unique, individual member. Only the community operator knows the true link between root and partial identity(ies).

Every partial identity has an associated feedback identity, which is used to provide feedback (including reputation), and a personal profile (which records a member's reputation). When a member provides feedback and/or reputation, they do so anonymously. The member may want to be anonymous because that are worried that if they are truthful and provide negative feedback they may feel threatened. (Extreme examples are e-voting and 'whistle blowing', e.g. a police informer.)

However, the member receiving the feedback wants to know who provided the feedback, so that they have some confidence that the feedback is honest and fair. Further, an 'observing' member wants to have faith in the reputation system as a whole. For example, on eBay a buyer probably has no idea who the seller and previous purchasers are, and has only the reputation system to help decide.

At a technical level, partial and feedback identities are identical. They both are unique values within the community. There should be no linkability between partial identifies, or between partial and feedback identities (accepting that in the initial trust model this will be possible at the community operator level). Feedback identities do not need to be visible to any other members. They could be randomly generated values, produced by the system when the partial identity is created, and remain an internal 'system value'.

Feedback identities allow members to provide 'anonymous' feedback and reputation. The fact that feedback identities are actually pseudonymous (i.e. know to the community operator) means that there is some control over how feedback is provided, and consequently higher confidence in the reputation system as a whole.

While members who provide feedback cannot be personally identified, certain attributes about them can be determined. For example, a member can only provide feedback once. If they have ten partial

identities, they can only use one to provide feedback (specifically reputation) on another member. The linkability achieved through the member's root identity enables the community operator to police this community policy restriction. The reputation of the member providing the feedback can also be revealed, as too can an aggregated reputation (root identity reputation), which is based on the reputation of all partial identities under a single root identity. Thus, without revealing the partial identity, it is possible to gain a strong feeling about the member who is providing feedback. Other factors help build confidence in the reputation system, for example members join the community through a strong registration process.

Members who are comfortable revealing their partial identity when providing feedback are also catered for. Members can add their true/real identity in the profile of a partial identity, if they chose, but it is also possible for a member to select the 'non-default' option to publicly link their partial and feedback identities in the feedback provided. This may be an attractive option for a sub-community administrator, or for someone which a high real-world reputation.

A community member who has multiple partial identities may want to be recognized for his achievements in united way. Therefore it may be reasonable in certain situation for the member to merge previously created partial identities. For example, as a member obtains a higher level of expertise/reputation in sea fishing, he may want to be recognized under single partial entity in both fly and sea fishing communities.

PUC5 discusses reputation in more detail.

13.4.2 Reference diagram

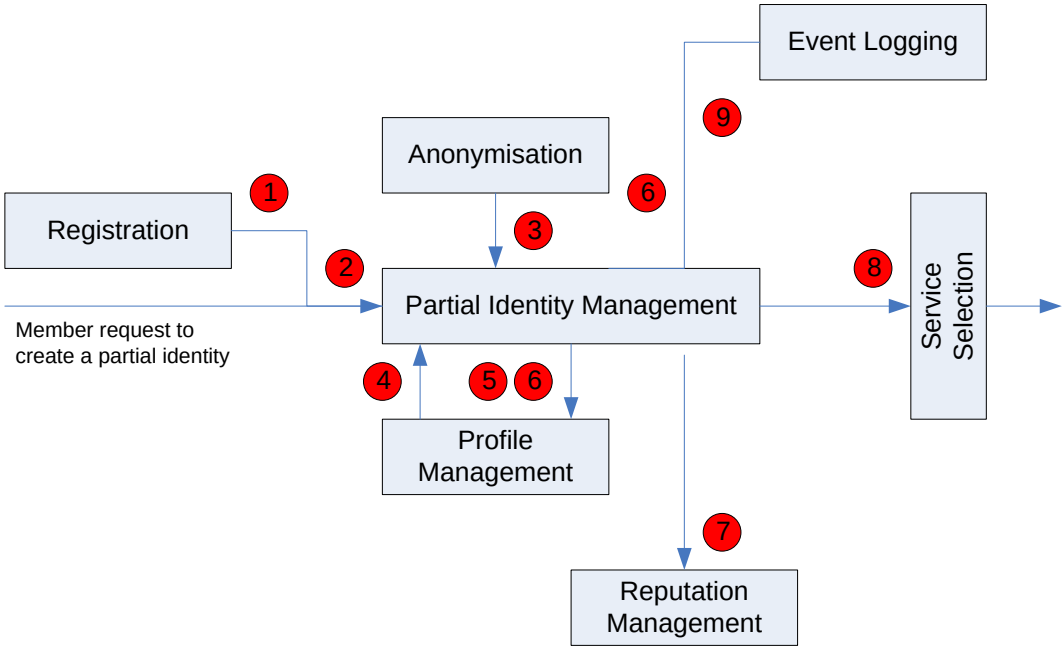


Figure 78 – PUC 4: Multiple partial identities

13.4.3 Walk-through

Once registered with a community, a member can create one or more partial identities. Each partial identity is registered with the community. (Note: When a member first registers, the *Registration* component calls the Partial Identifier component to create the first partial identity {1}.)

Partial identity creation is a service available to members through the *Partial Identity Management* component {2}. Every partial identity must be unique. The identity can be chosen by the member (so long as it is unique) but will more likely be automatically created by the *Anonymisation* component {3}. (Member-chosen identities may reveal the real identity of the member and should be discouraged. If members wish to identify themselves through their partial identity then they can include a photo or a name in the partial identity profile, and then selectively reveal this information to others members. {4})

The new partial identity adopts the privileges of the root identity {5}, but these can subsequently be modified (reduced) by the member through the Profile Management component {6}. The reputation of the partial identity would most probably set to a default 'neutral' value {7}, or preferably might inherit the reputation of the root identity (which is the 'average' of all partial identities linked to the root identity).

As currently envisaged, all partial identities are directly linked to a single root identity. A partial identity cannot be linked to another partial identity (but see Footnote to PUC 1).

Once a partial identity has been created, the member can use the new identity to interact with the community and select services available {8}.

It is also possible for a member to import a partial identity from another community. There are some limitations, e.g. both communities must operate compatible identity and reputation management systems, but in principle importing should be possible.

All actions performed are logged by the *Event Logging* component {9}.

13.4.4 Reference to the User Scenario in Section 3

John creates two partial identities, one for his fishing holiday and another for fly-fishing.

13.5 PUC 5: Reputation

13.5.1 Situation

Reputation is a reflection on the ability of an entity (normally a member, but also content, an asset, an external service provider or sub-community) to satisfy the values that a community maintains. It is directly related to trust. Reputation involves two processes: 1) providing information that builds a reputation, and 2) retrieving the reputation of an entity. In addition, members wish to contribute reputation anonymously while still being able to rely on (trust) reputation received. In practice, this means receiving assurance that reputation is provided by an authorised member of a sub-community, and that a level of accountability exists.

Members should also be able to 'link' items of feedback – a member may consider that feedback pseudonym 'xyz' provides feedback that they find particularly useful. This is one reason why the feedback pseudonym for a given partial identity never changes. Every feedback pseudonym has a profile.

Every partial identity has reputation, but so too does the root identity. Root identity reputation helps members understand the member behind the partial identity. Take a member who has two partial identities, one with a good reputation and the other with a bad reputation. It is arguably useful for members to know that these two reputations relate to the same members, but unlinkability prevents this. Although root identities are not visible to the membership, its reputation can be provided in such way that it 'averages' all partial identities under that root identity. The exact process is the subject of further research.

Members can provide feedback using each of their partial identities (pseudonyms). In each case, feedback is recorded under the corresponding partial identity feedback pseudonym. An exception is where a member can only provide feedback once, e.g. when voting. A member can provide feedback (textural comments) more than once, but can only comment on reputation once. However, reputation values can be edited if the contributing member wishes to revise their opinion.

Reputation can also relate to external services providers. Here, every service provider is assigned a partial identity which attracts a reputation value when members provide feedback. The service provider cannot influence the reputation value themselves. Third parties can also provide feedback on service providers. This might result from a trusted independent review (e.g. audit, consumer body) of the service providers operation, and their ability and willingness to comply with legislation.

13.5.2 Reference diagram

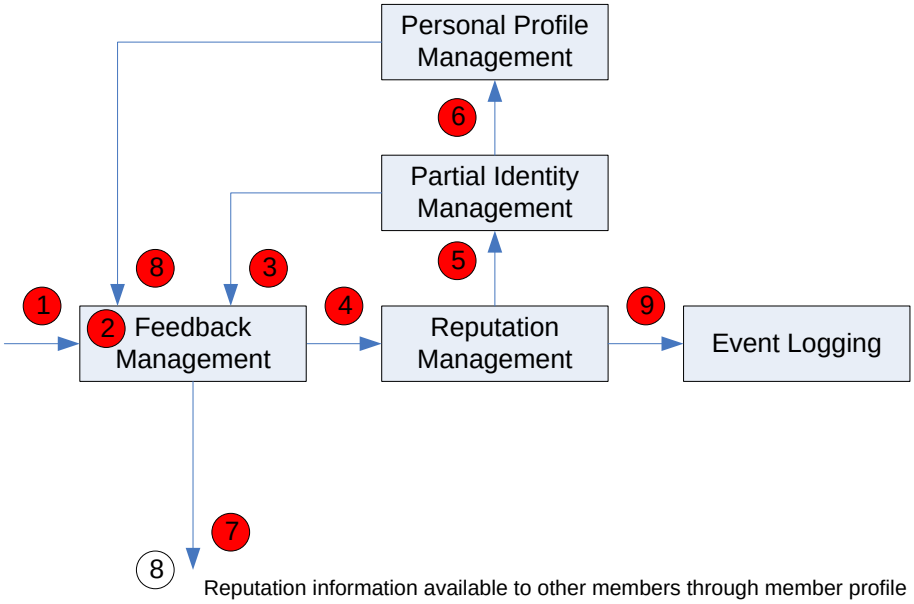


Figure 79 – PUC 5: Reputation

13.5.3 Walk-through

Reputation is received from members through the *Feedback Management* component {1}. This component allows members to select the entity {2} that they wish to provide feedback on, and prompts for a reputation indicator (e.g. on a scale of 1-5). Entity identities are obtained from the *Partial Identity Management* component {3}and passed to the *Reputation Management* component {4}, where the feedback pseudonym associated with the partial identity (of the contributing member) is located and {5}, and returned to the *Profile Management* component {6} where they are recorded in the profile of the entity to which it relates (normally a member).

The feedback pseudonym is a 'special' pseudonym that ensures contributing members are not identified to other members. However, it is important that the reputation service is not open to abuse. Therefore, feedback is provided under a pseudonym which allows the community operator to inspect the member root identity if necessary. The feedback pseudonym is created when the member first uses the feedback service.

Reputation is a read-only value that cannot be modified except by the reputation service or the community operator. When a member inspects the profile of a member they will see their reputation {7}. Other services that rely on reputation can also inspect the profile value. For example, in some situations, reputation can only be provided by members who have a positive or neutral value, in which case the reputation service will block feedback from members who have a negative reputation {8}.

All actions performed are logged by the *Event Logging* component {9}.

13.5.4 Reference to the User Scenario in Section 3

John uses the reputation service to check the reliability of information he uses to plan his trip. He also establishes a personal reputation, which he leaves with the community when he cancels his membership.

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

13.6 PUC 6: External services

13.6.1 Situation

The PICOS community supports a wide range of service. However, there will be times when a community requires a specific service that is provided by another 'external' community. For example, the angling community may require road or rail information in order to plan a fishing trip.

External services raise two sets of concerns, namely:

- Privacy
- Accuracy and reliability

External communities are represented within the community just like any other entity (i.e. member, resource). Every service has a locally maintained profile which contains the reputation information. Consequently, every service provided has a partial identifier against which reputation is recorded. In the case of a service provider, it may be desirable for the partial identity to actually name the provider (which is possible, though not advisable for ordinary members, because entities can choose a partial identity so long as it is unique within the community).

The selection of an external service triggers a request to the External Service Delivery component, which acts as a proxy for the external service.

Privacy

When a member accesses an external service, they do so under a partial identity (pseudonym). This special 'external services' partial identity is created automatically by the community 'on demand', in a similar way to which a standard partial identity is created under the member's root identity.

Accuracy and reliability

The accuracy and reliability of an external service is described by the service's reputation. Just like entities within the community, external services carry a reputation indicator that has been compiled by all communities that use the service, according to a standardised format for expressing and sharing reputation information. In addition, feedback information is available, although the quality of this information depends on the nature of the contributing community and its alignment (in terms of members' interests) with the local community.

Members can check the reputation of an external service using the Personal Profile Management component. This component returns the reputation to the member.

13.6.2 Reference diagram

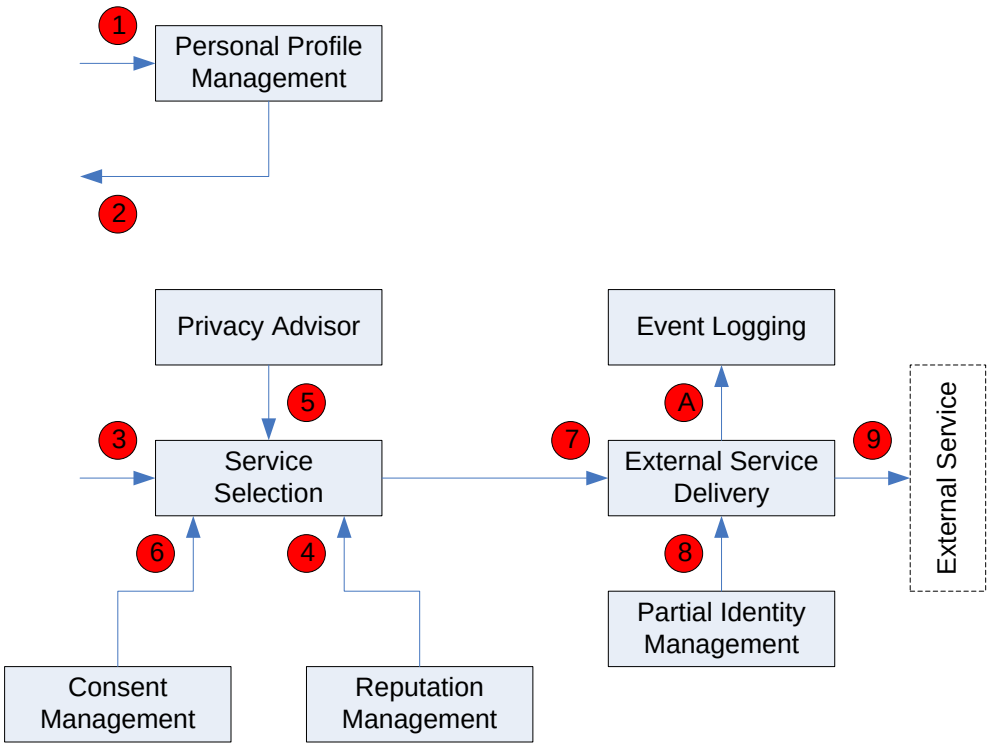


Figure 80 – PUC 6: External services

13.6.3 Walk-through

A member requests the reputation of an external service by selecting the entity identity from the *Personal Profile Management* component {1}. This component returns the reputation {2}.

An external service is selected in the same way as an internal PICOS service, except that it is accessed via a proxy {3}.

Prior to selecting a service, a member can inspect the reputation of the service provider {4} and obtain privacy advice from the *Privacy Advisor* component {5}.

The choice of whether a member shares their partial identity and other personal information with the service provider is stipulated on the consent that the member has given, and is available through the Consent Management component {6}.

The external service is accessed through the *External Service Deliver* component {7}, which calls the *Anonymiser* component if members have stated that they do not want to share their partial identity {8}. The *Anonymisation* component generates a pseudonym 'on-the-fly'. This pseudonym is shared with the external service {9}.

All actions performed are logged by the *Event Logging* component {A}.

13.6.4 Reference to the User Scenario in Section 3

John uses external services to check weather and biological information from FishBase. Access to the external services is described in the scenario as using Federated Identities. At present we do not have a use case covering federated access.

13.7 PUC 7: Content sharing

13.7.1 Situation

Members can contribute content, which includes all types of member generated data, e.g. text, video, (recorded) voice and images, to the community. This is called importing. Members can also remove (copy) content from the community, which is called exporting. The import/export component co-ordinates both activities. For example, the *Import/Export* component identifies the source (or name) of a photography that is to be imported.

Content is always associated with the identity of the member who performed the import, which can be a partial identity. When imported, content is tagged (i.e. attributes, or meta-data, are attached) to help identify the content to other member of the community. Certain tags are assigned by the contributing member, while others are derived by the community. For example, member tags may include description, sharing options and target community, while community tags include contributing member reputation (but not necessarily identity). The *Content Sharing* component is responsible for associating tags with content.

Once content has been successfully imported, the content is available for others members to view (subject to matching the profile stated by the importing member). Optionally, the importing member may actively notify other members (push) using the *Notification* component. The decision to notify or not is taken at the time of import, or is set in the importing member's profile.

Members must be connected to the community and authorised to access the import/export service in order to perform this action.

Tagging is a complex subject that requires further research. For example, members are likely to want to tag content, especially photographs, with the identity of the subject. This can be useful since it provides an opportunity to notify the subject (assuming that they are a member of the community) that a photograph of them is visible to the community. This only works if a valid identity is selected; if free format tags are permitted, then analysing tags is more complicated. The reality is that members would probably prefer free format tags. A partial solution is to notify all members of a sub-community every time any photograph is imported, so that it can be inspected. This is not a convenient solution for members.

13.7.2 Reference diagram

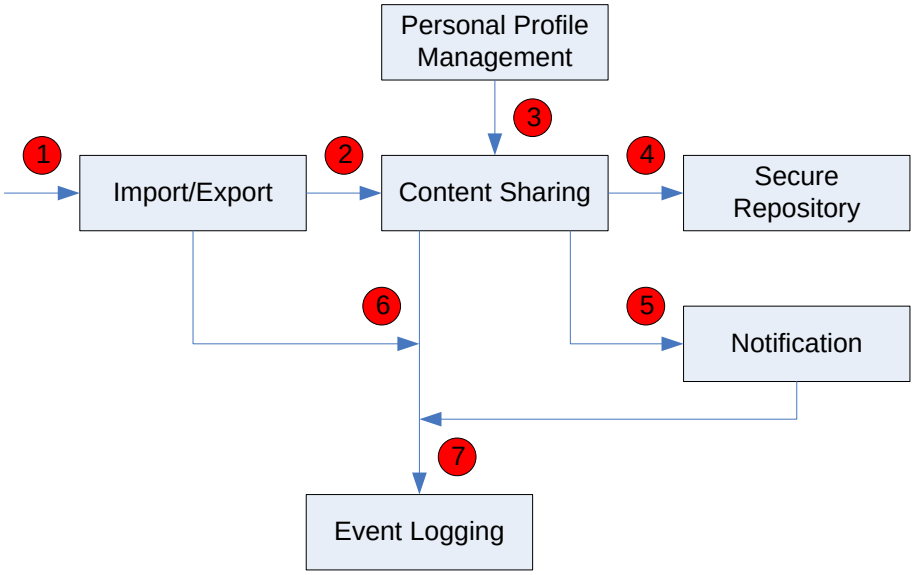


Figure 81 – PUC 7: Content sharing

13.7.3 Walk-through

A member, authenticated and authorised by the community, selects the Import/Export service and specifies the source of the content to be imported {1}. Next, the *Content Sharing* component tags the content {2}, taking tag information from the member and the member's profile {3}, and places the tagged content in the content store {4}.

Finally, the *Content Sharing* informs the *Notification* component to issue a notification that new content is available {5}. (Notification can be to the whole community, one or more sub-groups and/or specified members.)

All actions performed are logged by the *Event Logging* component {6} {7}.

13.7.4 Reference to the User Scenario in Section 3

John is keen to share content. For example, he creates a 'holiday in the Alps' sub-community. When cancelling his membership he is 'rated' (reputation) for his contribution.

13.8 PUC 8: Presence

13.8.1 Situation

The status of a member (called social presence) is freely available for other members to see unless the member concerned chooses to deny access to all or part of their presence information.

Presence describes a member's current situation, e.g. 'online', their location, role, sub-group membership.

Presence information is held as part of a member's profile. Every partial identity has a profile. Presence information is partially under the control of the member. For example, a member can choose whether to reveal their identity to other members, possibly using a simply on/off option that links to the *Preference Management* component. However, a member cannot falsify location information. For convenience the choice of whether to display presence information can be set as a preference for each partial identity.

13.8.2 Reference diagram

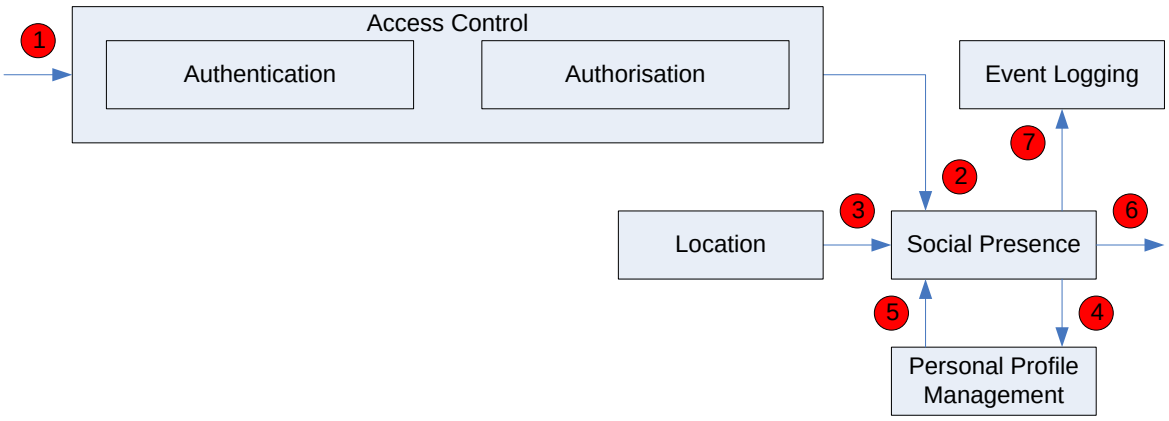


Figure 82 – PUC 8: Presence

13.8.3 Walk-through

When a member accesses the community {1} their profile is updated by the *Personal Profile Management* component to reflect their online status and location (and other context information that is considered relevant and is available). The *Access Control* component informs the *Social Presence* component of the change in member status {2}, which in turn triggers the *Social Presence* component to acquire the member location {3} (and other context information).

A member can influence how much information about themselves is revealed to other members, through the setting on their personal profile. Thus, the *Social Presence* component requests {4} this information from the *Personal Profile Management* component, and uses it {5} to set the member's social presence 'filter' before revealing presence information {6}.

All actions performed are logged by the *Event Logging* component {7}.

13.8.4 Reference to the User Scenario in Section 3

John creates several sub-communities, e.g. he creates a 'holiday in the Alps' sub-community which will only be accessibly by members when they are in the Alps on holiday.

13.9 PUC 9: Sub-community

13.9.1 Situation

All members belong to the single PICOS community, but each member can create one or more sub-communities. A member selects[28] which sub-community they want to interact with when they connect to the community. As part of the service selection process they can chose one of the sub-communities list in their profile.

These sub-communities serve a specific purpose identified by the creating member, and can be joined by any member with the permission of the creating member. The creating member can specify individual members, sub-community membership or filter on a set of member personal profile characteristics. For example, a member of the angling community might create a sub-community for anglers interested in using sonar[29] to locate fish.

Sub-communities created in this way take on some of the characteristics of the creating member. For example, the sub-community initially has the same reputation of the creating member. Sub-communities have profiles, just like any other entity, which can record reputation and maintain a list of all sub-community members.

When a member leaves a community, their sub-community(ies) can be deleted, transferred to community operator or delegated to another member. There may also be legal reasons for keeping the sub-community content, even though sub-community is no longer active. Transfer to another owner would require the consent of all sub-group members. Where a sub-group member does not accept the transfer, it must be possible to remove or anonymise their content. Any changes to the sub-community must be notified to all sub-group members.

[28] The number of members with the ability to confirm new members into a sub-community should not be limited to exactly one.
[29] Acronym for SOund NAvigation and Ranging.

13.9.2 Reference diagram

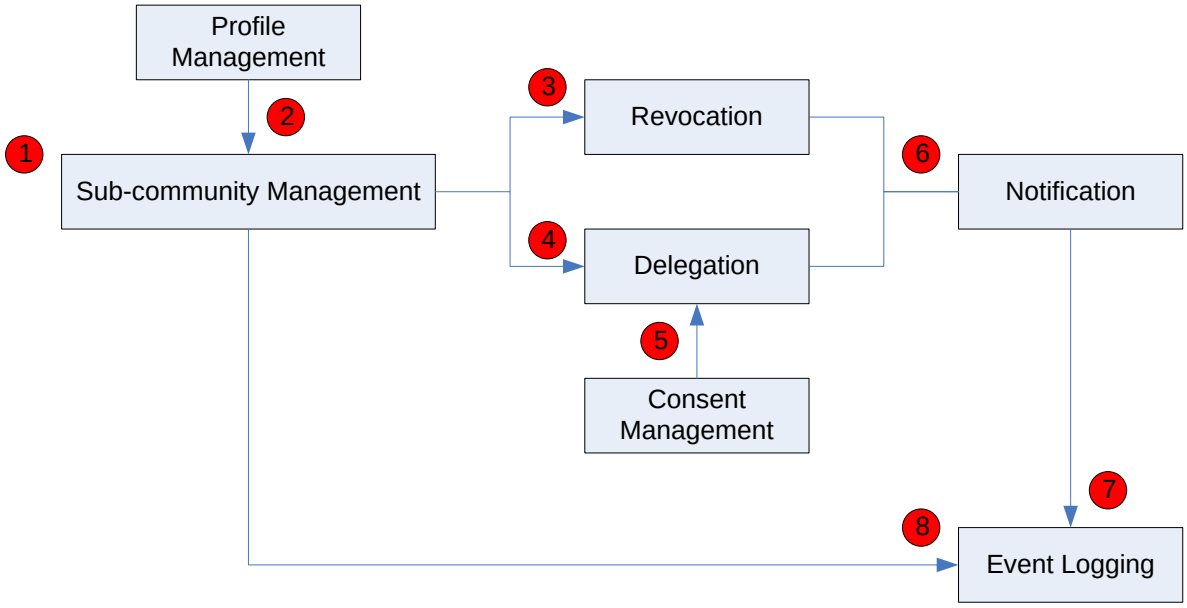


Figure 83 – PUC 9: Sub-communities

13.9.3 Walk-through

A member creates a sub-community by issuing a request {1} to the *Sub-community Management* component. As the sub-community is created, it adopts some of the profile properties of the creating member using the *Profile Management* component {2}.

The sub-community is either revoked using the *Revocation* component {3} or delegated using the *Delegation* component {4}. Delegation requires the consent of all sub-group members, obtained through the *Consent Management* component {5}. Whichever action is taken, all sub-group members are told using the *Notification* component {6}

All actions performed are logged by the *Event Logging* component {7} {8}.

13.9.4 Reference to the User Scenario in Section 3

John creates several sub-communities, e.g. he creates a ‘holiday in the Alps’ sub-community.

14 Example implementation

Although D4.1 is not primarily concerned with implementation, issues of building a practical architecture have arisen in discussions between partners. The focus of the discussion has been on platforms, architectures and applications. Two interpretations have emerged that provide a useful insight into how PICOS might be built. However, the final decision on how best to construct the prototype remains with WP5 and WP6.

In an earlier section we discussed topologies, mainly client-server and P2P. For the first prototype, Client-server is likely to be the preferred topology, not least because it fits with the models already adopted by communities and seems to be good fit for our target community in the light of the chosen trust model. Even so, it is not at first obvious where the split in functionality between client and service lies. The emphasis has been on a thin client and a think server, but beyond the first prototype this position must be reviewed. In addition, some functionality may not sit nicely on a single platform, so options to distribute functionality across platforms on the client, server or both sides will need to be considered.

It should also be noted that beyond the prototypes and the project, PICOS will almost certainly have to integrate with existing community platform technology, whether community management platforms, identity management systems or platforms that offer mobility support. This introduces an extra level of complexity, since PICOS cannot simply assume that the community application runs on top of the PICOS platform. Rather, PICOS needs to consider the case where the community application uses other (existing) middleware and services, e.g. services for community management, content sharing, for identity management, etc. This added complexity means that PICOS must offer extensions to existing middleware/services, rather than being a set of independent services. This third possibility is illustrated in the final diagram in this section.

All approaches are based on a client-server configuration, with a ‘thin’ client and a ‘thick’ server. Both split communication, community and PICOS functionality across different platforms. This is partly in recognition that in practice some of the technology may already exist. For example, an operational community will have communication and content sharing facilities already, and will look to PICOS to provide the additional privacy enhancing features. This is one reason why a PICOS Toolbox approach makes sense when dealing with legacy systems.

14.1 From Client to Toolbox

Since this initial architecture is essentially services-based, it can be visualised thus:

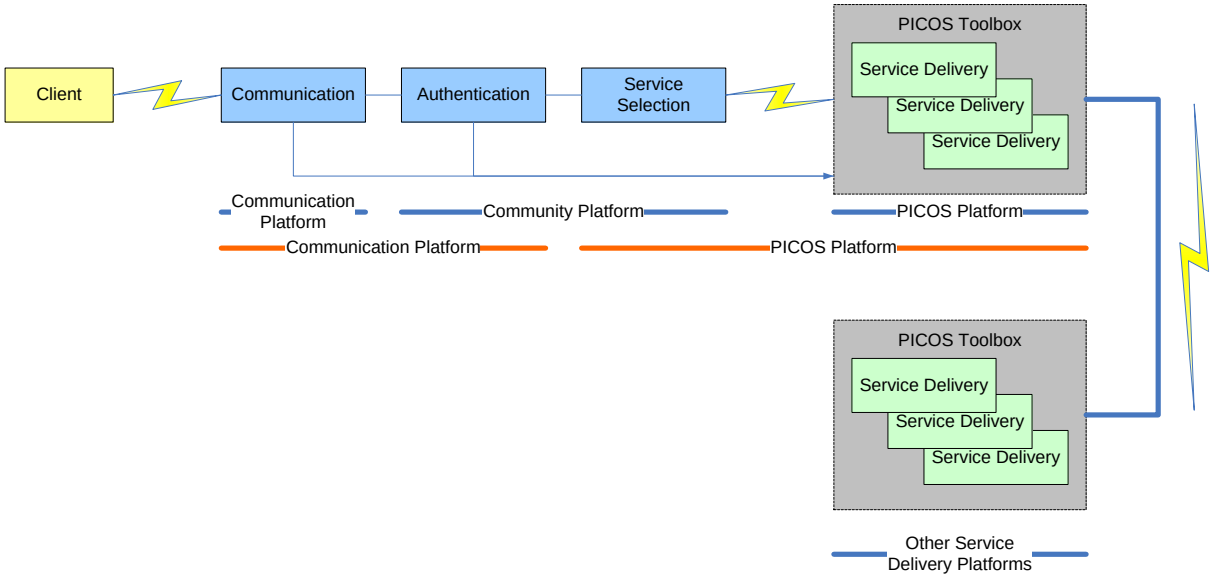


Figure 84 – Simplified services-based architecture

14.2 Platform-centric approach

In the first approach, the client is able to talk directly to three platforms (each providing communication, community and PICOS functionality respectively).

The community platform is responsible for managing content, and higher level communication and context (location) services.

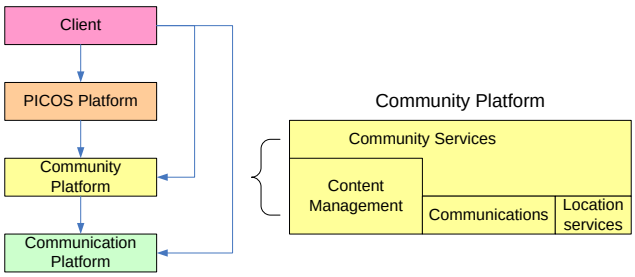


Figure 85 – Platform-centric implementation

The communication platform handles all communications between client and the community platform. For example, this service might be provided by a (mobile) network operator.

The PICOS platform hosts the privacy, trust and identity management functionality that PICOS offers.

14.3 Services-centric approach

An alternative approach is to look at the implementation from the perspective of services. A (mobile) client interacts first with an application, which in turn draws on services provided by a community platform or the PICOS enhancements (PICOS Toolbox). Additional services are supplied by external service providers.

Visualising the implementation in this way shows nicely the relationship between WP4, WP5 and WP6.

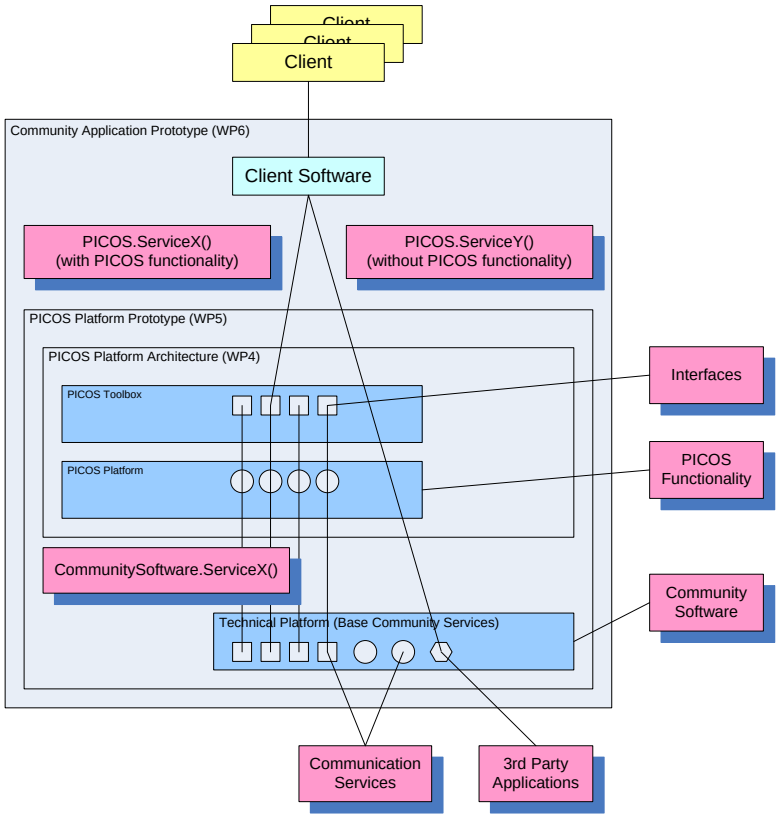


Figure 86 – Services-centric implementation

Other configurations are possible. In practice, the hardware may support more than one platform type, or the services may be distributed across several platforms.

In the earlier section on topologies we suggest that client-service is not the only option. A peer-to-peer configuration, which offers some attraction to members who are at the less trusting (low trust) ending of the trust spectrum, would position more functionality at the client. Another possibility is a hybrid, or pseudo-P2P configuration, in which P2P services are routed via a central hub (e.g. the community operator).

D4.1 focuses on the immediate needs of WP5 and WP6 and the first prototype, which in reality is likely to be client-server based. Thus any discussion about P2P implementations is left for D4.2

14.4 Working with existing communities and technology

As mentioned earlier, PICOS will need to inter-operate with established communities and existing technologies. One approach to achieve this is as follows:

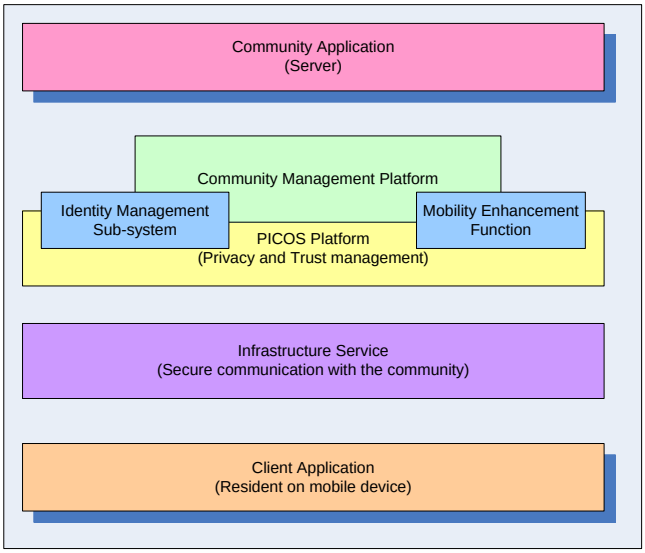


Figure 87 – Implementation w.r.t. existing communities

The PICOS implementation will need to integrate functions that enable privacy and trust into existing community management and identity management systems. Most community applications are implemented on top of community management middleware and they use existing services such as identity management services. In such a situation, the PICOS platform will not be implemented as a standalone set of services, but as extensions to existing management platforms/systems.

For mobile communities, or communities that want to offer services to mobile users, the same set of extensions can be applied to the community mobility platform (shown as the 'mobility enhancement function in the above diagram). Some functions of the PICOS platform may actually be implemented as extensions to this mobility enhancement function.

With a main focus on privacy and trust management, the PICOS platform will rely on the underlying infrastructure to provide generic security features, for example for secure connectivity between mobile devices and community applications and for encryption of data.

15 Link to WP5 / WP6

The first prototype is a learning opportunity for all partners, and likely to raise many questions surrounding the provision of PICOS functionality: What are the APIs that give access to the SDK? How is the Toolbox presented? What is the right level of abstraction? What practical problems arise when trying to extend existing platforms?

There is a strong connection between the three technical work packages. WP5 and WP6 use the WP4 architecture as the basis for their deliverables and, specifically, the prototype development. They extend the approach begun in D4.1 by gaining clearer understanding of the requirements and dependences of components and services. D4.1's features, components and the overall architecture provide a reference for future development.

The architecture feeds WP5 and WP6, by establishing the role and importance of the Use Cases, which give rise to a better understanding of the interaction of components and implementation needs. D4.1 helps with the selection of components needed to narrow the scope for the first prototype. It also helps the project decide where new technology must be created and where existing technology can be re-used.

But this process is not only one-way. In due course D4.2 will benefit from the work of WP5 and WP6, as components and concepts are tested, and opportunities arise to take the architecture further forward. In particular, the Use Cases that are started in D4.1, and are so critical to a shared understanding of the architecture, will be extended and improved as the prototype unfolds.

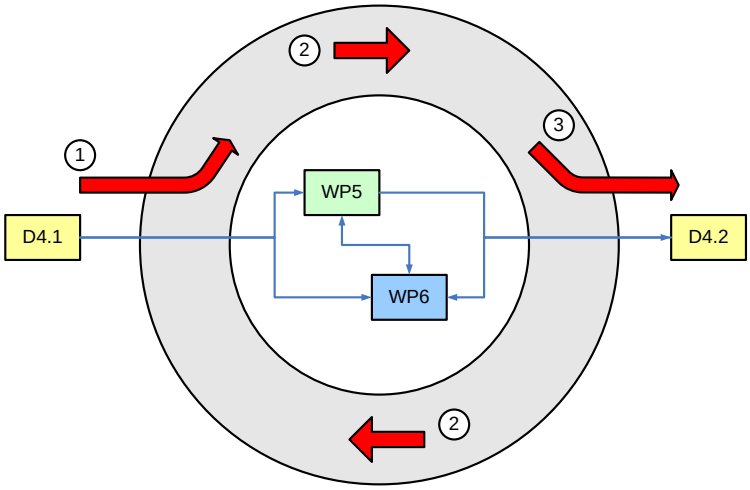


Figure 88 – Link to WP5/WP6

The above figure shows how D4.1 feeds into this interaction the interaction between WP5 and WP6 that will occur post-D4.1 {1}, how the interaction creates better understanding {2}, and how D4.2 is to be built on the experiences gained from all three activities {3}.

16 Research outlook

Research is an important element of the PICOS project, whether technical social or a combination of the two disciplines.

As the architecture has been designed it has become clear that some features that appear in components are either not well understood, or have the potential to make a significant contribution to PICOS if developed beyond their current implementation.

Each component that forms the PICOS architecture is marked if it is believed it requires further research:

Currently there are eight components that require additional research:

- Accountability
- Data Minimisation
- Linkability
- Partial Identity Management
- Policy Management
- Privacy Advisor
- Reputation Management
- Trust Negotiation

D4.2, which begins in November 2009 (and delivers in February 2010), will require greater understanding in these areas.

As we move beyond the architecture for the first prototype, new challenges arise:

- A stronger trust model
- Independent identity endorsement
- Independent law enforcement
- Move sensitive functionality to the client
- Privacy from Community Operator

The trust model described in Section 5 is highly trusting of the community operator. This offers convenience, but is unlikely to satisfy the needs of the more cautious member. It is possible to addresses this situation with stronger mechanisms to protect information. This was hinted at in the conclusion of the trust discussion, where cryptographic primitives called Group Signatures schemes

and Traceable (fair blinded) Signature schemes were mentioned. This is an area that D4.2 hopes to be able to build on.

Extending the current architecture to make sensible use of these techniques would in principle not be difficult. However, the techniques are complicated and more difficult to implement, and are thus an important area for further research within the scope of PICOS.

As we consider the stronger trust model, issues of scalability arise. Typically, the stronger trust model involves more external 'players', who need to coordinate their activities and share information. Trust Authorities and Public Key Infrastructures (PKI) become important technologies that can assist, but while not technically challenges, they require careful thought if they are to be implemented efficiently and effectively.

Placing greater trust in the client device also raises new issues. The client device may be required to store highly sensitive information, e.g. cryptographic keys, authentication information, or the most sensitive personal information. It is not clear at this stage if the current generation of smart phones will offer sufficient overall protection, despite the fact that many now support comprehensive cryptographic algorithms.

Appendix A Summary of PICOS components

Communications	
Tier-1	Tier-2
Communication Management	Network Security P2P Communication

Services and Applications	
Tier-1	Tier-2
Access Control Application Orchestrator Importer/Exporter Preparation Area Service Selection	Anonymisation Authorisation Authorisation Date/Time Stamper External Recommendation Federated Access Feedback Management Identity Translator Location Sensor Notification Partial Identity Management Payment Services Privacy Advisor Recruitment Reputation Management Scenario Management Service Delivery Social Presence Trust Negotiation TTP Management

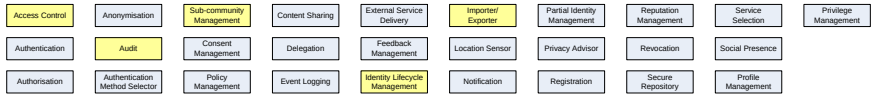
Audit, Control and Reporting	
Tier-1	Tier-2
Audit Intrusion Detection	Accountability Event Logging Event Reconstruction

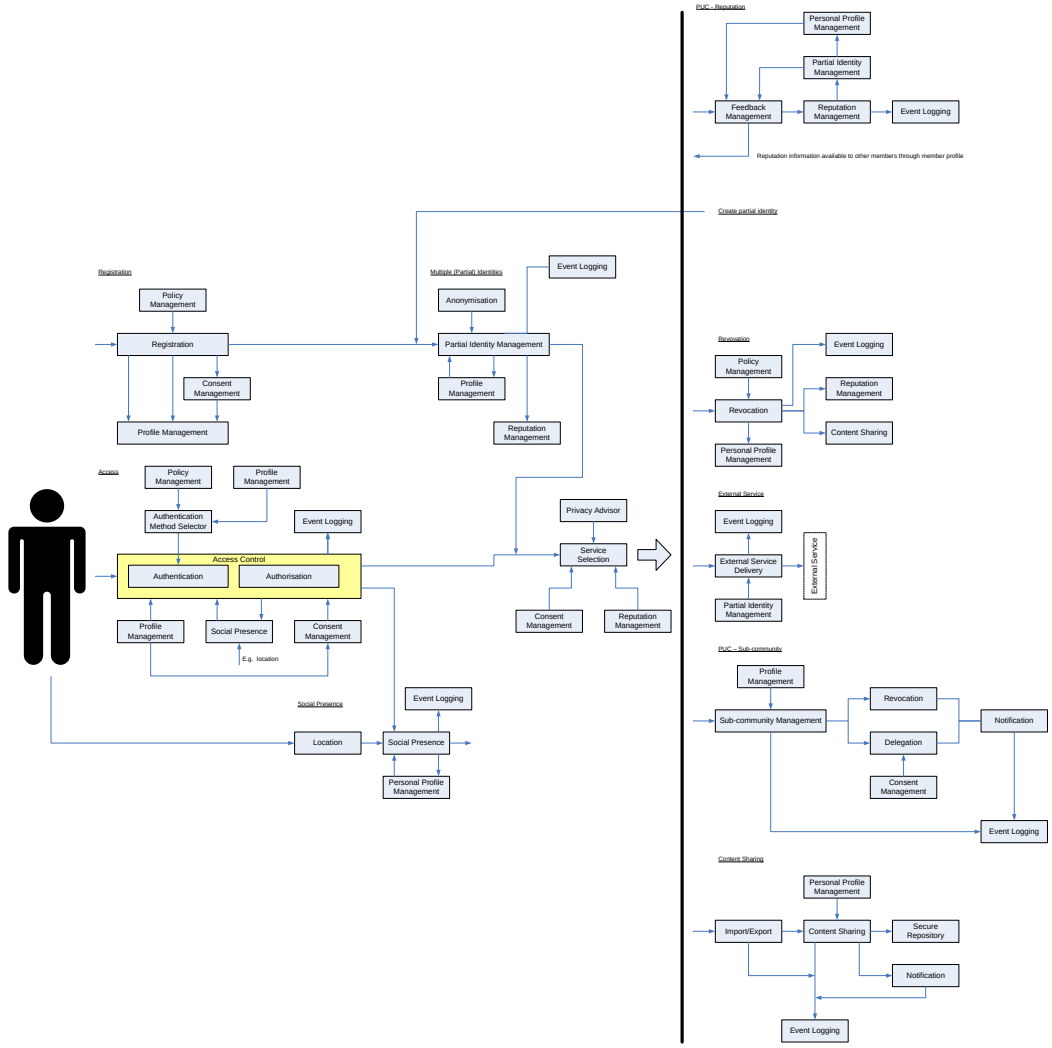
Member Administration	
Tier-1	Tier-2
Identity Lifecycle Management Sub-community management	Authentication Method Selection Consent Management Cryptography/Key Management Delegation Personal Profile Management Privilege Management Registration Revocation

Content Handling	
Tier-1	Tier-2
DRM	Content Sharing Data Minimisation Linkability Non-repudiation Secure Repository

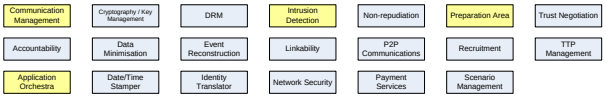
Appendix B Overall PICOS architecture / All components

Components featured in this version of the overall PICOS Architecture





Components NOT featured in this version of the overall PICOS Architecture.



Please see individual component description for an indication of how these components interact with one another.