First trustworthy wiki/blog prototype: Functional description

Editor:	Christian Hörtnagl (IBM)
Reviewers:	Benjamin Kellermann (TUD)
	Gregory Neven (IBM)
	Sandra Steinbrecher (TUD)

Identifier:	D1.1.2
Type:	Deliverable
Class:	Public
Date:	November 24, 2008

Abstract

This report forms part of the deliverable of a focal demonstrator for milestone M2 of the PrimeLife project. It discusses the selected use case scenario in terms of desired benefits and implemented functions, it outlines the high-level system architecture and public interfaces for integration, and it gives a brief overview of related work and a future outlook.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 216483 for the project PrimeLife.

Members of the PrimeLife Consortium

1.	IBM Research GmbH	IBM	Switzerland
2.	Unabhängiges Landeszentrum für Datenschutz	ULD	Germany
3.	Technische Universität Dresden	TUD	Germany
4.	Karlstads Universitet	KAU	Sweden
5.	Università degli Studi di Milano	UNIMI	Italy
6.	Johann Wolfgang Goethe – Universität Frankfurt am Main	GUF	Germany
7.	Stichting Katholieke Universiteit Brabant	TILT	Netherlands
8.	GEIE ERCIM	W3C	France
9.	Katholieke Universiteit Leuven	K.U.Leuven	Belgium
10.	Università degli Studi di Bergamo	UNIBG	Italy
11.	Giesecke & Devrient GmbH	GD	Germany
12.	Center for Usability Research & Engineering	CURE	Austria
13.	Europäisches Microsoft Innovations Center GmbH	EMIC	Germany
14.	SAP AG	SAP	Germany
15.	Brown University	UBR	USA

Disclaimer: *The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The below referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law. Copyright 2008 by IBM Research GmbH.*

Contents

List of Figures

Chapter 1

Introduction

This report contains a functional description of the focal demonstrator (internal code names: Bellini, Veracite) that was built in fulfillment of PrimeLife [PL07] milestone M2 (activity 1, work package WP 1.1). IBM Research GmbH leads work package 1.1. According to the PrimeLife schedule the focal demonstrator is due at the end of project month 9 (November 2008). It will be referred to simply as focal demonstrator in the remainder of this document. The final version of this document also incorporates feedback received during a scheduled project-internal review period (beginning in early November 2008).

The high-level objective of the focal demonstrator is to enable users to assess the trustworthiness of digital information found on the Internet, in particular in cases where the information is provided by many individuals and comes from diverse sources. In such situations, including similar situations preceding information technology, humans typically assess what other secondary information they can obtain about the (primary) information and who has provided the primary and secondary information.

Because of existing computer science terminology that is of relevance to the current implementation of the focal demonstrator, we will often refer to the primary information simply as information, and to the secondary information as meta-data. In this report, we are not concerned with a possible exact distinction between data, information, and knowledge on philosophical grounds, and will in particular use the terms data and information (as well as content) interchangeably, unless it is mentioned otherwise.

1.1 Trusted information

In approaching the focal demonstrator, an experimental representative computational system that enables users to assess the trustworthiness of digital information, we must first decide on a trust model. A serviceable trust model should enumerate the conceptual components that form a possible solution to the trust decision problem at hand, and describe also their relationships and interactions on a conceptual level. Such a trust model can serve as a starting point for comparing possible alternatives against the actual solution, for documenting the scope and other assumptions underlying it, and for deriving a concrete solution architecture and implementation of the focal demonstrator.

In terms of initial scoping, we have aimed at keeping a certain amount of flexibility around the actual trust model, and not prescribe in it aspects of human behavior that may not be universally held, but are treated as subjective by real individuals. Decision making in general, and therefore also trust decision making, is in part inherently emotional and subjective. There are candidate elements for a possible all-encompassing trust model (that is, one including subjective human behavior) available from psychology, sociology, ethnography, and also mathematics (e.g. tit-for-tat strategy from game theory). But it is currently unclear how these manifold contributions could be successfully combined into a coherent aggregate, without at least partly contradicting each other. E.g. the history of artificial intelligence suggests that an attempt still exceeds the capabilities of current science, especially so if the targeted problem domain is as broad as helping to decide whether to trust a (*any*) piece of electronic information that is placed in front of us.

As a corollary, this also suggests that a more realistic and useful approach consists of an initial trust model that establishes a reliable basis in such a way that it does not contradict more detailed models (to the extent that non-experts can be aware of their full spectrum) and that confines itself mostly to areas where common overlap can be presumed to exist. Ideally, such a model later gives rise to experimentation with some of the more detailed models and theories from social sciences. This experimentation should occur in an incremental and evolutionary fashion, so that the detailed models that serve best in each situation could be chosen and tuned based on context-specific and personalized empirical evidence.

When translated into the functionality of a computational system based on such an actual trust model, the relative reduction in scope amounts to a solution that does not attempt making fully automatic trust decisions on behalf of users, but instead presents relevant (primary and secondary) information to them in such a way that they can conveniently and efficiently interpret it as part of the still mental task of arriving at final trust decisions. This act of interpretation can then include whatever additional subjective considerations they wish to apply. In other words, the solution aids users in forming their human trust decisions; it does not replace or incapacitate them.

On a coarse level, our initial trust model comprises these few mandatory components:

- There is *information* (also referred to as primary information) on which judgment as to its trustworthiness is required. Arriving at an appropriate trust decision constitutes solving the trust decision problem. (Can some information be trusted in a given context: yes or no?)
- A *consumer* (or consumer of primary information) is a human who is confronted with taking a trust decision on said information, and to whose current context and circumstances it should be appropriate. Usual definitions of trust imply that the person who trusts puts herself at risk with respect to outcomes that are controlled by others whom she trusts. If the trust decision concerns information, as is the case here, this risk amounts to possible concrete effects from acting on supposedly trustworthy information.
- There is *meta-data* (also referred to as secondary information or annotations), which is information with the specific additional property that it is about the primary information. Here we don't make a distinction between how many steps of referencing occur between the secondary and primary information, but just insist that it has to be at least one. As a result, for the purposes of this overview meta-data and meta-meta-data are treated as the same. In practice, meta-data (such as an assessment of the form "I can vouch for this piece of information") and meta-meta-data (such as evidence about "my" online reputation or professional credentials) may receive different treatment. As a case in point, the two kinds form different parts of the visualization that is employed in the current implementation of the focal demonstrator (see Figure 1).
- An *author* (or author of meta-data) is a person or other cryptographic principal who creates and binds a given instance of meta-data. Unlike with (primary) information, we assume that

all pieces of meta-data carry this authorship information, and that a firm and permanent binding is established e.g. by explicit cryptographic signing of the duple. This is part of a larger assumption, whereby we require that meta-data is more structured than arbitrary information. (By not extending this structural assumption also to the primary information, we can readily employ a single solution also for the assessment of legacy web content, which is generally unstructured on the semantic level. Notice that our use of the term author refers only to authorship of meta-data, and that other authorship roles, such as those in relation to outside reference material that an annotation may point to for improving the information's verifiability, are not touched by this trust model.)

Like other network-of-trust-inspired systems, the resulting trust model assumes that in cases when the consumer does not know whether to trust the primary information, she can triangulate by taking other, better-known authors and their meta-data into account. Notice that software agents could also serve as principals and therefore authors. This means that automatically created meta-data (e.g. from reading sensors, or from translating other information as is described in section 3.4.2) are well within the model's scope.

In addition to these mandatory components, there are several optional ones, such as trust policies. These are artefacts whereby consumers can declare requirements on meta-data that must be met before a pursuant piece of information can be seen as trusted. Since individual users may or may not wish to delegate larger authority for automatic trust decision making on their behalf to software agents that apply such trust policies (see also the earlier argumentation about the most desirable initial scope) trust policies are considered only as non-mandatory in this trust model.

Overall, situations that are in scope of the trust model are conceptually similar to what often happens also outside the domain of electronic information. The trust model therefore replicates actual social processes between people, an arrangement which we deem as desirable and in fact a necessary prerequisite for the success of social networking software. We want such software to mimic actual social processes but in the digital realm.

As a concrete example, we may refer to a human resources manager who evaluates information submitted by a job applicant by considering both the primary information (resume, etc.), plus secondary information about this primary information (support letters making direct reference to parts of the job history). Both the primary and secondary information count, and it is of high importance where and on whose authority the secondary information came about, i.e. that it is from a person whom the evaluator knows and can trust. In cases where the evaluator does not know the candidate directly (as will frequently be the case), considering the secondary information and its source helps her in arriving at a sound trust decision by allowing that a certain amount of trust bestowed on the source of the secondary information is extended towards passing judgment on the (primary) information as well.

Trust in content is a multi-layered concern. Making a distinction between the following aspects also indicates that different aspects are accessible to automation in a computational system in varying degrees. The order is from concrete and relatively deterministic (therefore allowing straightforward software support) to more abstract and cognitive. A reasonable scope for a computational system, such as the focal demonstrator, negotiates and includes some of these aspects approximately in order.

- *Integrity* of information guarantees that the information has not been tampered with since its creation (or its last legitimate editing).
- *Binding* of information concerns keeping records on which parties participated in the creation or editing of information. In other contexts we also use the term linkability for the same concept; however, we avoid it here, in order to guard against possible mix-up with the distinct use of linkability in relation to zero-knowledge proofs [CL02].

- *Context* of information provides further evidence supporting that a consumer should trust any of these participating parties, especially given the type of information concerned and the type of its intended use.
- *Accuracy* of information concerns whether said information is internally consistent and externally verifiable.

1.2 Blogging scenario

The focal demonstrator concentrates on a blogging scenario. Blogs are a representative type of Internet content that is continuously updated by many individuals and organizations. Updates occur by adding new time-stamped articles. For instance, news headlines and reports from news organizations are now commonly available as blogs, and numerous individuals are maintaining what resemble online diaries in the form of blogs. We interpret the term blogs in a relatively broad sense, i.e. not just encompassing individuals online journals, but all content that is "pushed" over RSS or Atom protocols, and other similarly structured content (e.g. from electronic mailing lists) that is easily transformed into the common format.

The issue of whether online information can be considered trustworthy is especially urgent when new information arrives that has to be acted on quickly. This may well be the case with blog articles that can convey important political, economic, or other news; yet these may arrive from an initially unfamiliar source of origination. By providing consumers with a technological means for not only viewing the primary information online, but in the context of related assessments by others whom they are acquainted with, and who in turn may be better acquainted with the primary information, we can facilitate more educated trust decisions that are of benefit to consumers.

Regarding the desirable conceptual scope of the focal demonstrator, we repeat that trust is ultimately a personal decision. Different individuals may make different choices even when presented with the same "objective" evidence, and not everybody is able or even willing to express what exact considerations go into their respective trust decisions. This partly gray area has to do with fundamental questions about the degree of rational vs. "gut" decision-making, which we cannot attempt to settle during this research effort (see also section 1.1).

The focal demonstrator rather concentrates on bringing the known pieces of evidence to a consumer's attention so she can take them also into subjective account. As an important prerequisite, the system ensures that a user that consumes meta-data can objectively know that it is related, who authored it (in an absolute or pseudonymous sense), and that it has not been tampered with. The focal demonstrator assumes that the human consumer then proceeds from this evidence and makes an actual "binary" trust decision by inspecting the evidence so visualized. However, the trust model also does not rule out that later on trust decisions could be formed e.g. by applying data mining techniques for automatic classification (see section 4.6).

We have been investigating two specific scenarios where blog usage can create tangible value for stakeholders, and where this value can be enhanced by increasing software support around ascertaining trustworthiness.

In a first instance we are looking at a population of employees of a multinational company (or members of another organization) who consume news from similar sources in the form of blogs. Not each individual participant may have the capacity to judge each piece of information in its self-contained form (for a start, it may be phrased in a foreign language), yet the entire "crowd" of members can form an enhanced overall view for "inside" members on augmented "outside" information. This version of the use case is of immediate relevance to the PrimeLife consortium member who was tasked with implementing the focal demonstrator, and it has the considerable

advantage that it allowed an agile development style, where feedback from first-hand use was able to inform some incremental design decisions and hence to improve the deliverable's quality. The scenario has already been tested inside a large corporate intranet during the lifetime of PrimeLife, and experiments have led to first qualitative results.

In a second instance we are looking at private individuals who are consuming health-related information (e.g. consider treatment options adjacent to interviews with their physicians), and who have obvious warranted interest that this information be trustworthy (e.g. "Is it advertisement? Is it a rumor? What does my insurance company say about it? What is it?"). This instance of the use case is especially relevant because it may concern each individual citizen, and furthermore it directly points to the importance of including privacy-friendly technology. Experience shows that individuals somewhat differ in their judgments as to the most desirable and practical levels of privacy [Br99] based on cultural background, politics, age, and other factors; yet privacy is generally held an undisputable right and value when it comes to information that concerns personal health. Relative importance of each pro-or-con argument may shift again as new developments occur in societies, so in order to stay maximally focused on the enabling technology that PrimeLife is about it seems to make particular good sense to concentrate on a health-related version of the motivating use case as well.

Chapter 2

User Experience

For describing the user experience that is enabled by the focal demonstrator we will refer to users that occupy either one of two roles. First, users acting in the role of *consumers* need to establish the trustworthiness of some online information. Second, users acting in the role of *authors* support the first by augmenting pieces of information with meta-data, according to their expertise and motivation. These two roles also constitute components of the trust model that was introduced in section 1.1.

In a larger context, the role of authors has to be considered in tight coupling to their possible incentives. In the short run this is necessary in order to gather enough meta-data and to successfully bootstrap this system, as well as any other social networking system (see also Metcalfe's law, which states that the systemic value of a network of compatibly communicating devices grows and accelerates with the network's size). Even more critically, in the long run this also constitutes a challenge resulting from asymmetric information [SCW01] effects. Following the principal-agent model from economics, the principal (consumer, in this case) wishes to delegate a task to agents (authors, in this case), who possess relevant information in private (meta-data, in this case). In order to prevent conditions of moral hazard (e.g. fake meta-data) she must device contracts between herself and agents in such ways that rational agents will stick to the contracted terms, even if they take the relative information deficit of the funding party into account. The theory of incentives and other results from economics allow e.g. a quantitative valuation of incentives that would form parts of suitable contracts.

However, it is not obvious how some of the assumptions made for the ideal economic models could translate into constraints encountered in an actual use case such as the blogging scenario. For instance, by which mechanism would consumers express their valuation of trustworthy information, or how could such contracts be reliably enforced in an online environment? For the time being, we just record that a body of established research on the economics of information could inform a principled approach to providing much-needed incentives, and we note this as a possible area of future work (see also chapter 6).

An important principle for the design of the user experience was to amend familiar ways of working with online information, and to augment them as necessary without disrupting existing practice. Two representative ways that are very familiar to EU citizens and many other users from young ages up form the foundations of the two kinds of extended user experiences that we

provide. As will be recognized as relevant in practice, these correspond to reading blog articles either with a web-based aggregator or with a specialized program. We therefore refer to their two augmented versions as a browser-based user experience (section 2.1) and a reader-based user experience (section 2.2) respectively.

2.1 Browser-based user experience

The browser-based user experience assumes that a consumer has chosen a web-based aggregator (such as Google Reader) for reading blog articles or consuming any other web page. This user experience is enabled by a Firefox extension (see section 3.4.1), and it makes the whole spectrum of supported functionality available to users acting both in the roles of consumers and authors. Figure 1 shows a concrete example of the resulting visualization (see also section 3.4.1).

Figure 1: Mashup presentations blends information and pursuant annotations.

2.2 Reader-based user experience

The reader-based user experience assumes that a consumer has chosen a dedicated feed-reader software (such as Mozilla Thunderbird), as opposed to a normal web browser (such as Firefox). Because of limitations concerning actually encountered software the system can only make a fraction of its supported functionality available as part of this user experience. For instance, many feed readers are unable to handle HTTP forms (required to support users in the role of authors of free-form annotations) or HTTP authentication. However, these readers can delegate certain tasks to web browsers, so that in practice this user experience consists of a combination of employing both a feed reader and a web browser (the second quasi as co-processor).

Figure 2 depicts an instance of a display that can be achieved inside a standard feed reader (Vienna RSS/Atom newsreader in this case); this normally approximates what is also conveyed in the browser-based user experience. When blog articles are listed by titles, those are also augmented in such a way that they quickly indicate the presence of meta-data that is related to the

titled article (functionally equivalent to color-coded icon in browser-based user experience; see section 3.4.1).

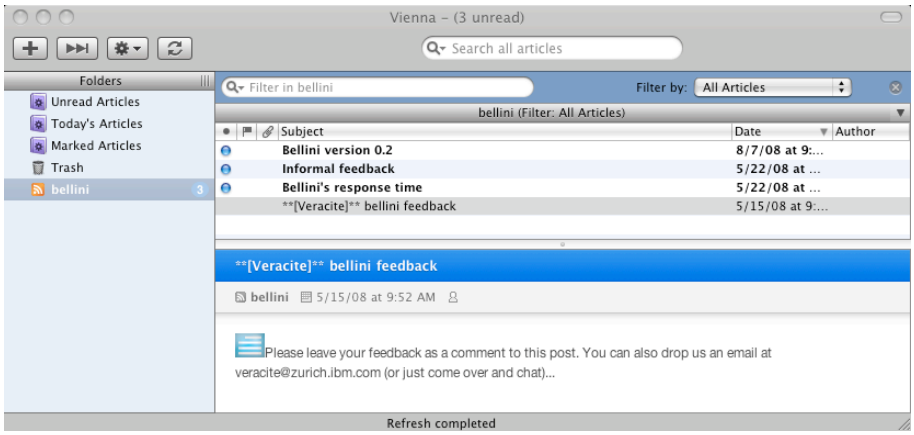

Figure 2: Reader-based user experience.

Chapter 3

Building Blocks

This section gives a brief summary of the software architecture for the code base of the focal demonstrator. Section 3.1 introduces the main components and their interplay. It also informs about the currently assumed deployment environment. The remaining dependent sections sketch the purpose and coarse responsibility of each identified high-level component.

3.1 High-level architecture

Figure 3 depicts the high-level component architecture of the focal demonstrator. Since it delivers a web-based user experience to users both in the roles of consumers and authors (see chapter 2) a natural decomposition into server-side and client-side components applies. Server-side components are drawn below the dashed line in Figure 3. A Firefox extension forms the only client-side component (in addition to regular web browsers or feed reader software that are assumed as generic and given, and which users can pick on preference from a palette of standard-conforming choices).

As befits a web-based solution, server-side and client-side components interact using protocols that reside atop Hypertext Transfer Protocol (HTTP) transports. Our architecture favors standard protocols (e.g. Really Simple Syndication, RSS), or otherwise employs the architectural style REST (Representational State Transfer) [Fi00] for specifying few proprietary ones.

The high-level architecture suggests dividing components into three broad categories: middleware components, front-end services, and other components. Sections 3.2 to 3.4 look into each of these categories in sequence, and chapter 4 identifies and describes the public interfaces, which are also indicated in Figure 3. The public interfaces provide the principal handles for integration with other, also third-party components.

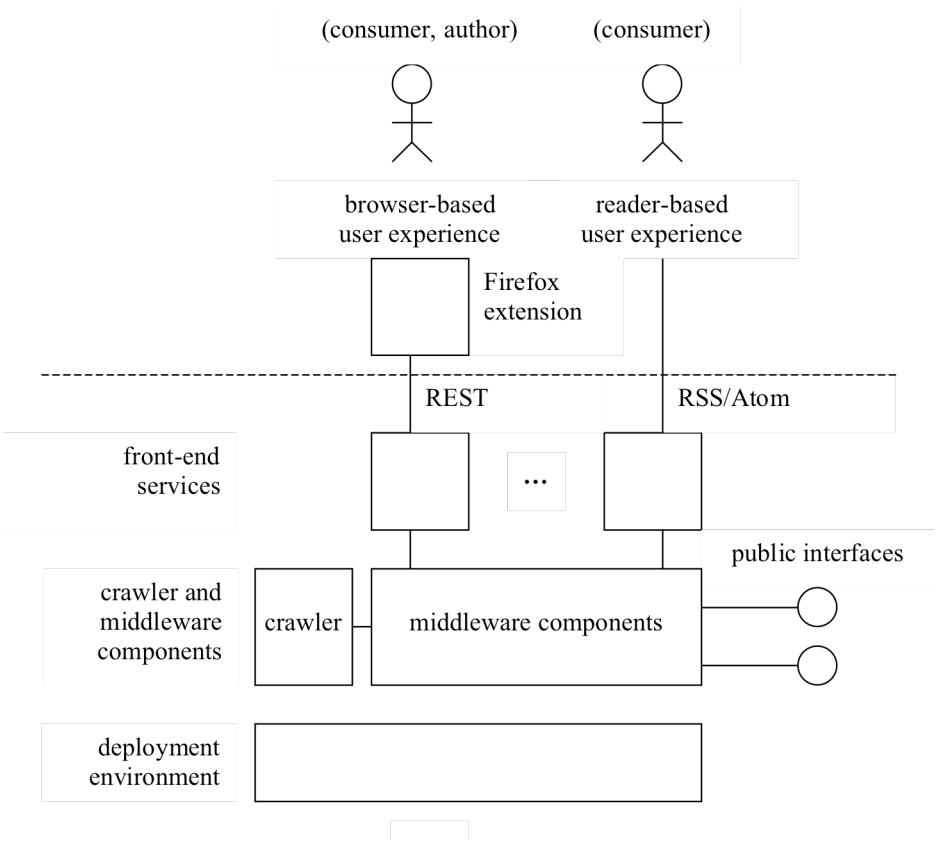

Figure 3: High-level component architecture.

3.1.1 Deployment environment

The focal demonstrator is largely deployed inside a web application container (Apache Tomcat) and assumes a relational database system (PostgreSQL). Only the crawler resides outside this container and is configured as a (Linux) system service.

The entire server-side code base can execute on a Java Virtual Machine (mixed Java and some Scala source code) and offers a correspondingly high degree of portability. More detailed assumptions e.g. as to the presence of a specific Linux operating system distribution (Gentoo) are confined to few configuration scripts.

The Firefox browser extension follows the usual conventions for Mozilla Firefox components (JavaScript), and has been tested with versions up to 3.0.3.

3.2 Middleware components

Middleware components exist mainly in the form of Java packages and libraries. They group pieces of focal-demonstrator-level functionality that become reused in multiple other places.

3.2.1 Secure annotation support

This component supports the secure binding of information to its pursuant meta-data. For each invocation, it expects a duple consisting of information and (optionally) its meta-data as input, and it yields a *bound URI* (BURI) as output. BURIs are opaque strings of characters that follow the

standard syntactic conventions for Uniform Resource Identifiers (URIs) under the HTTP scheme. Each BURI is chosen such that it identifies a unique pair consisting of a piece of information and its meta-data each.

The process for arriving at a BURI consists of two or three steps, having to do with normalization, versioning, and binding respectively. (Even if the third step is skipped we refer to the result also as BURI, which in this case represents a versioned and normalized piece of information alone.) The mechanism for constructing BURIs combines encoding certain parameters inside an XML document whose location corresponds to part of the BURI, as well as encoding its cryptographic hash value in another part.

- *Normalization* is concerned with the fact that different forms of representation may amount to the exact same content. At the lowest level byte-representations-related normalization involves tasks such as rearranging white space (e.g. during XML canonicalization). At a more abstract level it may require stripping information off all parts in its byte representation that are not immediately apparent to human inspection. For instance, embedded pieces of JavaScript code may fall into this category. By binding those to meta-data, an author may enter unintentional and undesirable liabilities (e.g. when embedded code is allowed to modify the display of surrounding information in such a way that certain clauses remains hidden to an author at the time when she applies her signature). It is therefore important to protect authors from such hidden consequences using normalization.

  The supported normalization mechanisms also address alias naming, such as in the case where same blog articles are advertised both under permanent identifiers (permalinks), as well as more convenient shortcut names.

- *Versioning* is concerned with the situation that normal Uniform Resource Locators (URLs) refer to pieces of information whose logical Internet address remains constant, while the content stored behind can change at any future time. As a case in point, news organizations' brands are associated with mnemonic URLs, but the corresponding web sites' contents are obviously expected to change with timely new developments. Versioning therefore comprises means such that URL-type references can address pieces of information that are constant both in terms of Internet address location and in terms of their exact content.

  Practical versioning mechanisms can either prescribe that a single version remains entirely unmodified (technology choice: cryptographic digest values), or otherwise limit changes to well-understood semantic manipulations that authors can allow before revoking their existing meta-data (technology choice: policies). In the second case, authors can trade larger convenience from not having to re-bind some of their annotations against a similar increase in their possible liabilities. The employed policy language must be sufficiently straightforward, such that meta-data authors can clearly understand the exact nature of these trade-offs and indirect exposures that may result over the lifetime of a piece of modifiable information.

  All supported references can either relate to web pages (such as blog articles) as a whole, or to fractions thereof (technology choice: XPointer).

- *Binding* refers to forming an aggregate consisting of both a piece of information in its normalized as well as versioned form on one hand and its pursuant meta-data on the other. The architecture relies on a binding mechanism in an abstract sense (see section 4.4) and for instance does not require that a particular digital signature algorithm be used.

3.2.2 Crawler support

Crawler support consists of a library for the handling of relevant data formats and protocols (Atom, RSS, OPML), plus implementations of related heuristic methods of several kinds:

- heuristics for distinguishing relevant hyperlinks from similar "noise" (see section 3.4.2)
- heuristics for inferring authorship of a blog article (or entire blog) in cases where authorship by principals is not explicitly encoded

3.3 Front-end services

The front-end services comprise HTTP protocol end-points of two kinds:

- JavaServer Pages (JSPs) for furnishing web-based user interfaces (e.g. management portal)
- servlets for offering REST-based access to platform components (e.g. RDF store)

Front-end services serve as mediators between users of the system and integrated components belonging to the system on one hand, and core functionality surrounding the middleware components on the other. The components that are currently exposed as REST services are not yet fully aligned with the conceptual public interfaces in as listed in chapter 4.

3.3.1 Management portal

The management portal offers a web-based user interface for giving access to more detailed functionality that addresses the supported needs of both consumers and authors. Figure 4 depicts a screenshot of the management portal.

For the browser-based user experience, users can install the Firefox browser extension from this portal (see section 3.4.1). For the reader-based user experience they can subscribe to specific aggregated feeds (see section 3.3.2). They can also list annotations and manage (edit, delete) their own annotations (see section 3.3.4). Finally, they can view further concise online information about the demonstrator and leave feedback.

Figure 4: Management portal.

3.3.2 Feed aggregation

As part of the reader-based user experience (see section 2.2), consumers can subscribe to modified versions of blogs. The modification consists of replacing each contained original blog article with an augmented version that ensures that a visualization of its meta-data is also in place when the primary information is viewed. This is accomplished by requiring that the consumer subscribes to a new blog feed address, which is similar to the one she is originally interested in. Consumers can do so without much effort: the management portal can automatically convert from known feed addresses to corresponding new ones (see Figure 4: "subscribe to annotated feed").

As a result of this modified subscription, the feed aggregation component gets involved whenever the user's feed reader program inquires if new blog articles have become available for the blog she is interested in. The component logic then has the opportunity of replying with a modified (RSS or Atom) feed document, which in turn refers to a modified version of each eligible article. As a result of this modification to the feed document, the page visualization component (see section 3.3.3) gets involved whenever the feed reader program inquires a particular new blog article, and it can therefore effect a mashup-style visualization of the type that is depicted in Figure 1 and Figure 5.

3.3.3 Page visualization

This component performs the actual mashup-style presentation that combines the display of information with an indication of its meta-data. If performs its task by forwarding to a dedicated JSP component and leads to a similar visual result as depicted close to the left margin of the window in Figure 1 and Figure 5.

3.3.4 Meta-data management

This component is concerned with providing convenient access to stored RDF meta-data (see section 4.5) for the purposes of storing, obtaining, and deleting meta-data both in terms of its binding to information (see section 3.2.1), physical storage, and life-cyle management.

The focal demonstrator also offers a degree of privacy for consumers (in addition to optional privacy for authors; see section 4.1) that prevents them from having to expose their browsing behavior as a result of subsequent requests for meta-data as they continue browsing. A privacy-unfriendly (let alone malicious) remote component could collect all submitted addresses and thereby gain information that many users would not expect to share with a remote party. The afforded protection consists of phrasing requests in approximate rather than exact terms. If an answer exists, i.e. a sought piece of meta-data is stored, a component under more direct control of the consumer (currently the Firefox extension that resides strictly locally on her computer) can detect the fact from the approximate answer received from a remote component. The remote component thereby remains ignorant as to exactly what she was looking for, and this remains in effect her private concern. The exact mechanism consists of a manipulation of certain message digest values and a suitable protocol for sending remote queries.

3.3.5 Security and authentication

The security and authentication component is in charge of performing authentication for all web-based user interfaces and protocol end-points that belong to the focal demonstrator. Client components can contact and invoke this component using URL/HTTP redirection. The specific authentication style employed is not prescribed as fixed and can range over a number of alternative options (see section 4.1). Therefore different degrees of user privacy can be supported as a configuration concern.

3.4 Other components

This catch-all category comprises remaining high-level components that do not fit into either of the two previously discussed categories.

3.4.1 Firefox extension

A Firefox browser extension for the focal demonstrator delivers the browser-based user experience (see section 2.1). It allows users to navigate the Internet as they normally would with Firefox (or alternative web browsing software).

In a first instance, the extension provides visual clues about whether any annotations for the displayed web page do exist. This summary visualization consists of an icon in the browser's navigation bar that changes its color between orange and blue (see rectangular, horizontally striped icon left inside the navigation bar in Figure 1 and Figure 5). Orange color indicates that no annotations are available. Blue color indicates that annotations are available.

In a second instance, clicking on the icon brings up a more detailed page with a mashup visualization consisting of both the web page's original content (primary information) and its annotations (secondary information), if applicable. At this point authors may also enter new free-form annotations. Figure 1 and Figure 5 depict the visualization of example meta-data alongside the left margin of the window. Exact visualization styles can differ according to the type of

annotations concerned (see section 4.3). The Firefox extension delegates the actual rendering task to the page visualization component (see section 3.3.3).

3.4.2 Crawler

The feed crawler is a stand-alone software component in the high-level architecture that is tasked with creating new (crawler) annotations. We have seen from experience that bootstrapping a social networking service that depends on user-generated content can be challenging, especially when no brand perception, nor peer pressure, nor other direct incentive that gain authors reward do yet exist. Relying on an early adopter community alone is risky at best, and can also lead to certain biases. We have recognized the need for incentives, and in chapter 6 argue that stronger privacy protection can be understood and also positioned as one kind of relevant incentive that is of special interest within the wider scope of PrimeLife, beyond this deliverable.

Another alleviating counter-measure to incentive problems consists of a deliberate early design choice that gives equal weight to both automatically generated and manually provided meta-data. The crawler is a representative software agent that automatically generates annotations, and as such by definition is unconcerned with human incentives.

If a blog article refers to another blog article (or any other web page) by including its address as an outgoing hyperlink, this can be interpreted as an annotation of the second article by the creator of the first one. The exact semantic relation that is thus conveyed can differ. Depending on circumstances, it could e.g. amount to either criticism or endorsement. (Observation suggests that endorsement is the typical default.) In principle, techniques that are also of relevance to search engines could be employed for inferring actual semantic relations with some precision (e.g. analyze natural language in proximity to a hyperlink). Furthermore, in practice there occur also complications from hyperlink "noise" in the form of online ads or other hyperlink-laden content from third parties that dilute the sought annotation "signals" inside blog articles and especially mashups, and that need to be filtered out e.g. by applying heuristics.

The implemented crawler currently takes a simple approach. Its configuration takes a list of blog feeds (as OPML configuration files), which are then regularly scanned for new blog articles in a first step, and for "signal"-type hyperlinks in a second. For each hyperlink thus identified, the crawler creates a new crawler annotation instance in the RDF store. The current ontology for crawler annotations (see section 4.3) models a marker for this type (next to properties that all types of annotations commonly inherit according to the core ontology), but does not attempt to discern more fine-grained semantic meanings.

The crawler possesses heuristic means for identifying the owner of each blog, and hence the author of each piece of meta-data that was added in this way (see section 3.2.2).

Chapter 4

Public Interfaces

In order to allow integration with other components both from within the PrimeLife project (e.g. Idemix-type identities and electronic signatures) and beyond (e.g. open source components) the focal demonstrator was designed with a number of public interfaces in mind. Some of them are already represented in the code base; others for now exist only on a conceptual level. In addition, detail modifications may later occur based on experience from further integrations.

An interface-centric approach to integration was deemed especially appropriate, because delivery of this focal demonstrator is scheduled relatively early during the lifetime of the PrimeLife project. Since it occurs before most other work threads have concrete deliverables that might be integrated, the immediate goal was integrability rather than integration.

On a deployment level, the interfaces are or will be realized as RESTful web services, except in cases where HTTP protocol overhead is prohibitive. Local library calls constitute the alternative in such cases.

4.1 Identity of users

The notion of trustworthy information that is assumed by the focal prototype emphasizes checking additional evidence in the form of meta-data about a piece of information, especially in situations where the information itself is unascertained, yet the authors of related annotations are sufficiently familiar. In a computational system knowing authors reliably means that it must be possible to identify them to a sufficient degree e.g. for the purpose of checking their signatures on meta-data.

In addition, the system provides a consumer of information with a visualization that indicates her (social) relations to the authors who provided relevant meta-data. Therefore it should also be possible to identify the user who consumes information in the system.

The overall system design leaves open what specific technologies for identifying users are employed; it only prescribes a contract in the form of a public interface that allows different implementations according to different needs. For instance, it may not be desirable to fully identify users as real-world individuals, but only as members of defined abstract sets (roles) or owners of defined (possibly anonymous) credentials. In this way, authors do not have to release highly personal information that is not relevant to their task; yet the consumers still obtain enough

relevant information e.g. for judging degrees of expertise or seniority. As a case in point, PRIME's Idemix-type credentials can be employed to accomplish this effect.

The current focal prototype identifies users by their IBM intranet credentials (enterprise-wide unique identifier and password). Providing these credentials is optional for consumers, and users who opt against logging in can still use the system with reduced functionality.

4.2 Reputations of users

While the system does not perform actual trust decisions on behalf of users, its visualization component (see section 3.3.3) attempts bringing evidence to the consumer's attention, by placing it as ambient clues around the primary information display. In order to present this evidence to users in a meaningful way, the full presentation must include both the annotations as such ("what does the evidence suggest?"), plus human-interpretable additional information about their authors ("where did the evidence arrive from?").

Reputation systems already provide access to such human-interpretable information, by summarizing opinions expressed by their users about each other. The reputation systems that form part of online commerce sites such as Amazon.com or eBay are known as frequently cited examples. Also social networking sites offer mechanisms for reputation management, with the tracked properties and metrics depending on the target application or audience. For instance eBay tracks its sellers in terms of percents of customer satisfaction. The business-oriented social networking site LinkedIn tracks members partly in terms of the numbers of recommendations received from other accredited members.

Since reputation systems are already manifold and spreading, the focal demonstrator deliberately does not attempt launching another new one, but instead assumes access for the purpose of obtaining relevant information from an existing one or a federation of several. Accordingly, the design of the focal demonstrator does not prescribe a fixed recommendation system, but instead foresees another contract and a public interface, which eligible reputation systems must be able to fulfil and implement respectively.

The current implementation of the focal demonstrator mainly uses one characteristic that is tracked by an IBM-internal reputation system. It describes membership of individual employees in tagged interest communities (example tags: "security", "lifescience"). Hence the context in which the focal demonstrator places evidence consists of the indication whether an author belongs to the consumer's larger circle of acquaintances (employee of same company), and whether she belonged to a smaller circle of acquaintances (overlapping memberships in interest communities for a consumer-author pair). It also displays some additional information from a corporate LDAP directory (this approximately resembles information commonly found on business cards). This integration between the focal demonstrator and the corporate-internal reputation system has been completed and is fully functional as part of the existing intranet deployment that underwent review.

Overall, use of these simple membership tags by the focal demonstrator is a representative example of how more encompassing information can be federated in from reputation systems at large.

4.3 Structure of meta-data

An early design decision for the focal demonstrator opted for structured annotations. This is in seeming contrast to simple tags, which appear as very popular unstructured annotations on the

present Internet. Knowledge about structure can substantially aid processing, for instance when it comes to visualizing annotations or applying policies. Moreover, unstructured annotations seamlessly fit under the same paradigm, because they can be seen as reduced to the most trivial form of structure (comprising only a single sub-element of type string).

With structured annotations it is possible to sort annotations into different types, depending on structural properties, and to model a corresponding type system. Since the focal prototype in turn models annotations following the standardized Resource Description Framework (RDF) [BHL01], it is natural to employ ontologies as type systems and for establishing structure in the form of instantiations of such types. The system then becomes extensible to new types of annotations, because (most of) types' characteristics can be expressed declaratively in ontology definitions, which can then be imported and interpreted at run-time.

Again, the focal demonstrator does not prescribe a single or fixed set of ontologies, but foresees another public interface for integration and extension. Also this interface can be instantiated as a configuration concern; the corresponding contract prescribes the language for expressing ontologies (the Semantic Web's ontology language OWL) but no restrictions on the terms defined in new ontologies. (The only exception to the second concerns certain assumptions contained in a small core ontology.)

The current focal demonstrator employs three ontologies, a core ontology plus two specific example ontologies (each of these two forming an extension of the core ontology) for two different types of annotations. The first type is *free-form annotations*, as a representative for manual creation of meta-data (and also resembling much-familiar tags). The second type is *crawler annotations*, as a representative for automatic creation of meta-data (by the crawler in this case). Some of the dependencies on the two example ontologies (such as visualization patterns) are currently wired programmatically into the code base, and therefore are not yet characterized in a declarative way, as would be the case if they formed an integral part of the ontology definitions. Thus the respective amount of flexibility exists on the conceptual, and not yet on the implementation level.

Figure 5 shows an example case where a single resource (blog article) has been annotated with two annotations, one being a crawler annotation and the other a free-form annotation.

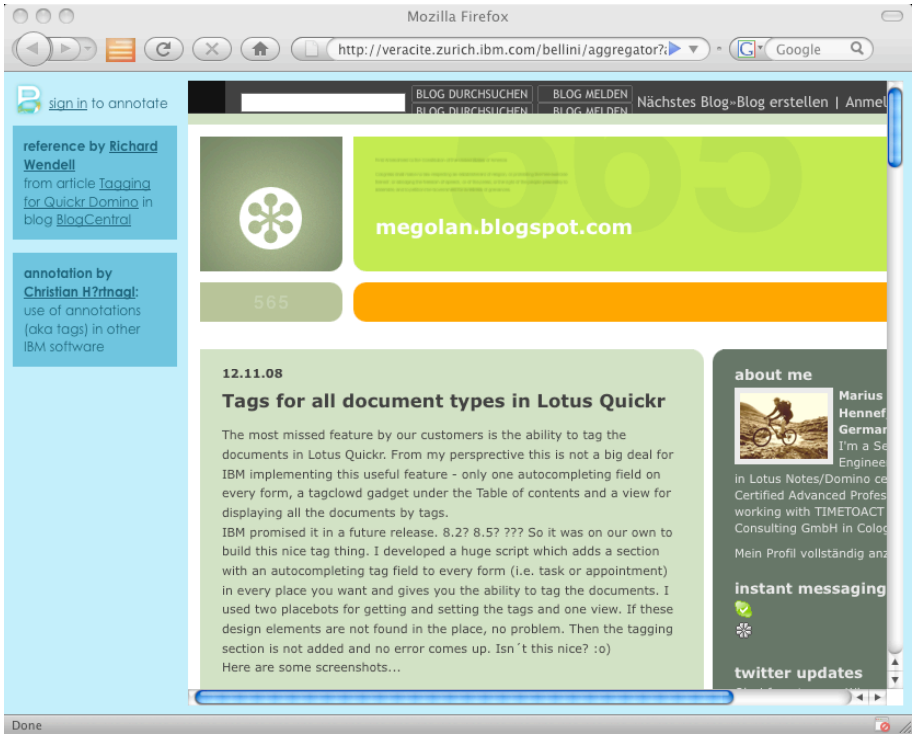

Figure 5: Annotations of two different types placed on a single resource.

4.4 Binding of meta-data

Secure binding of meta-data to information requires means for applying a principal's digital signature to a message that combines both. The design of the focal demonstrator treats the capability for applying such signatures as another abstract building block. A similar rationale as mentioned in section 4.1 with respect to anonymous identification applies here as well.

It is for instance possible to employ Idemix-type signatures (based on non-interactive zero-knowledge proofs) [CL02]. This has the enhanced effect that consumers can reliably check on certified properties held by authors, while not necessarily learning about their full individual identities. Anonymous certificates release more narrowly defined (if still highly relevant) personal information about authors as would be the case e.g. with conventional X.509 certificates, and this gives a greater degree of online privacy to authors. In this way the design of the focal demonstrator already covers some privacy-preserving mechanisms. When alluding to this keeping of authorship information, we use the term *binding* (e.g. over signing) in order to emphasize that even schemes that may not electronic involve signing could apply under the same contract.

The current prototype does not yet instantiate the corresponding interface. The rationale for this is that at this time all annotations are managed centrally and stored in a single RDF store, where data corruption is less of an issue. A concrete interface specification could refer to the XML Signature standard, which itself includes extension elements e.g. for expressing different kinds of electronic signatures on XML-formatted content, and which covers issues such as XML canonicalization and serialization of relevant data structures.

The focal demonstrator also addresses the normalization of content before its signing (similar to XML canonicalization, but on a semantic instead of syntactic level; see section 3.2.1).

4.5 Storage of meta-data

The distinction between information and its pursuant annotations (meta-data) is partly an artificial one. Meta-data also comprises information, and can itself be annotated by recursive meta-data. From a practical perspective, the focal demonstrator does not require that information possesses a fixed structure. Any resource that can be identified by a URI can serve as eligible and potentially trustworthy information, regardless what form of content resides "behind" the unique identifier. In contrast, it assumes that meta-data is structured as a well-formed RDF model. This design decision makes constructive sense, because it allows the annotation of all existing web pages and other legacy content, while still facilitating type-dependent processing of annotations. Therefore, one pragmatic distinction between information and meta-data can be drawn depending on structure.

A similar range of possible distinctions also applies to where information and meta-data each are stored and located. In a setting that emphasizes use of legacy data sources (such as with web pages' in the focal demonstrator) it is natural to keep the two apart, i.e. to leave information as is, and manage meta-data separately (for example in an RDF store). On the other hand, Semantic Web standards provide technological means such that both information (expressed as XML) and meta-data (expressed as RDF model) can reside in single XML documents. RDFa is one emerging standard with relevant characteristics.

Again, the conceptual design of the focal demonstrator does not require a particular location or choice in favor of co-location of annotations relative to the pieces of primary information concerned. A conceptual public interface is in place to cover a variety of different arrangements.

The current focal demonstrator opts for a central RDF store that is implemented atop a relational database. We employ the corresponding capabilities of the Jena Semantic Web framework, and have augmented them by a RDB-based rapid index for optimizing the most prevalent types of queries.

4.6 Trust valuation

Trust valuation refers to the process of condensing all available meta-data that belongs to a piece of information. If forms part of a process that ultimately leads to a binary trust decision on the information whose trustworthiness is under consideration. We have already emphasized that trust valuation is considered mostly outside the scope of our initial trust model (see section 1.1), and therefore the decision process will typically be a human's mental process to a large degree. In it a consumer receives the visualization of evidence and then tries to make sense of this evidence in an individual way. (This is in contrast to a fully automatic process, where a consumer would only register a binary decision outcome that was calculated by software alone.)

However, we also expect that in certain situations a larger part of the process can be automated. For instance, if frequent annotations of a given type are expected, each consumer may receive enough incentive for formulating a trust policy, i.e. stating a set of rules that describe the trust valuation process in explicit terms. A concrete instance of such a policy for filtering trustworthy news articles on a given topic may require that an article from a certain newspaper website is marked as "noteworthy" by a certain acquaintance, or by anybody else with certain credentials. (The topic itself would also be conveyed by similarly supported meta-data.)

Alternatively, there may be opportunities for supporting trust valuation by treating the decision process as an automatic data classification task. Given enough annotations of a certain type, it may be possible to train an automatic classification system from user feedback, and over time at least calculate suggested default choices that may amend or facilitate the consumer's mental process.

Data mining (and privacy-preserving data mining) can come into play here. In order to allow these kinds of extensions, the conceptual design of the focal demonstrator allows one more degree of freedom as to the type of automatic trust valuation applied.

The current focal demonstrator implements a minimal trust valuation that visualizes verbatim evidence with additional context information (see section 4.2 and Figure 1), but does not yet attempt automatic classification or summarization.

Chapter 5

Related Work

The focal demonstrator for milestone M2 shares some characteristics with project Annotea [Ko05] from W3C. In particular, both use RDF-based annotations for augmenting existing web content, and both employ Firefox extensions as part of their gathering and visualization of meta-data. In terms of offered functionality, this demonstrator puts relatively more design emphasis on security and privacy aspects and related implications.

It for instance considers secure binding of meta-data to its pursuant information and also with anonymous signing mechanisms. This is especially important in situations where pieces of information are repeatedly edited and aggregated to form part of distributed online resources, as opposed to when all parts remain secure inside a central meta-data store. A recent incident, where an erroneous news article led to a transient drop in the sock price of a large air carrier, illustrates how this could matter in a tangible way.

The focal demonstrator also ensures that consumers of meta-data are protected against revealing their online behavior as an unseen side effect. This can for instance be important in a situation where individual EU citizens seek trusted online information before making personal health decisions. Overall, the focal demonstrator puts more emphasis on filtering information (and doing so in a privacy-friendly way), while Annotea was more concerned with spotting and searching for new information.

The focal demonstrator is also loosely related to systems that perform authentication based on networks of trust (e.g. Pretty Good Privacy – PGP, Simple Public Key Infrastructure – SPKI), as well as a body of research on data lineage (a.k.a. data provenance) [MGM08].

Chapter 6

Outlook

The PrimeLife project emphasizes that individuals in an information society must be able to protect their autonomy and retain personal information when using online services. A large part of the ongoing technical work program in PrimeLife is devoted to resulting privacy challenges. For instance, some technical work is devoted to pushing the technology frontier in the area of privacy-enhancing cryptography.

The focal demonstrator that was developed for PrimeLife milestone M2 (see chapter 1) has focused on assessing the trustworthiness of online information, as was planned [PL07]. For the limited purpose of work item 1.1, privacy-support was not yet a focal concern, although it is already reflected as a preliminary in several aspects of the conceptual design.

The chosen trust model (see section 1.1) aids individuals in arriving at personal trust decisions by presenting blog articles (or other online information) together with relevant meta-data inside an otherwise familiar application environment (see section 2). The designed solution maintains a strong permanent binding between information, meta-data and further secondary information on who has provided the meta-data, so that consumers may ascribe some of the context-relevant meta-data to principals whom they are familiar with and therefore trust in a given context (either directly as individuals, or in the abstract as constitutional entities). Such strong binding between information and meta-data can anchor their own mental (and however informal) reasoning about transitive trust relationships in a similar way as axioms would do in a fully formal setting.

The general topic of enabling users to assess the trustworthiness of blog articles remains highly relevant, especially since such information often has a limited lifetime and may require actions quickly after its arrival. For instance, it is clearly conceivable that information of importance to personal health decisions (such as about traveling to certain world regions on business, or considering a new treatment option while fighting disease) may reach individuals first via blogs or similar electronic channels. This is partly the case because newspaper and journal articles are by now distributed as blog articles (in the sense that they use RSS or Atom protocols for distribution), and so are dispatches in the electronic online journals of many individuals who may have pertinent first-hand experiences e.g. on a given health matter.

Preliminary experience from deployment of the focal demonstrator on a corporate intranet suggests that there remain challenges about providing enough authors with sufficient incentives

for making their efforts on behalf of others (the potential consumers) worthwhile. Evidence shows that otherwise the quantity and quality of meta-data may not be sufficient for creating a sustainable market-like exchange and for making the particular use case effective and worthwhile in practice. If left unmitigated, a chicken-and-egg problem may hinder successful adoption.

So far it has emerged that viable mitigative incentives can be categorized into four kinds:

- Stimulating incentives in the form of monetary payments (e.g. micro-payment schemes)
- Stimulating incentives in the form of other valuations (e.g. reputation schemes)
- Stimulating incentives in the form of side-effects (e.g. games with a purpose [vA06])
- Protective incentives in the form of privacy protection

Because of the overall scope of PrimeLife and its privacy-related main goals, an investigation of the effects of stronger privacy-protecting mechanisms with respect to web applications similar to this focal demonstrator forms one possible area of future work.

A corresponding refactoring of the code base could commence around the public interfaces that already provide a measure of privacy support (see sections 4.1 and 4.4). Promising research questions could include the relation of such protective incentives to serendipitous models from economics (see chapter 2), and resulting trade-offs in relation to data mining (see section 4.6).

Anonymously signed instruments could also serve to support electronic payment and micro-payment schemes for offline use. E-cash of this kind could directly enhance the present or a similar PrimeLife use case by offering one kind of stimulating incentives.

E-cash payment options have not been frequently offered to consumers until now, partly because conventional credit cards still cope well enough with their larger e-commerce transactions, while small dues (such as for online access to commercial newspapers sites or hosted applications) tend to be funded indirectly in the form of online advertisements. The current global economic crisis has already led to a reduction in consumer spending, and this may in effect also lead to a similar reduction in consumer advertisement, a resulting shift in fundamental business models, and a possible reconsideration of e-cash and electronic micro-payment schemes as possible complements and alternatives.

Online advertisement-financed services almost by their definition run counter to the expectation of strong personal privacy, because the margin-generating promise of this new advertisement channel lies exactly in its enhanced targeting of individuals' circumstances (and even some of their hidden inclinations, as suggested by data mining techniques). Traditional cash payments of small amounts by individuals, on the other hand, have always been anonymous insofar as tracing single bills or coins has never seemed practical. This older pattern has both important privacy-related advantages (spending and privacy touch in many significant ways in each individual's life, for instance again when it comes to health-related choices) as well as disadvantages (money laundering, etc.).

It would therefore seem important that a possible strengthening of (offline) e-cash payments should factor the expedient privacy-related properties of paper money in as well as guard against known disadvantages (e.g. by privacy revocation under controlled circumstances), and get privacy right from the beginning of the design and maturation. PrimeLife technology that is already under development could be used and demonstrated to this effect [CHL05]. Giving individuals a way to spend future e-cash in a privacy-friendly way (and one that does not disrupt their normal expectations from handling paper money in the real world) is another possible contribution towards protecting the autonomy of individuals in an information society and retaining their personal information, as expressed in this project's goals.

References

[BHL01] Tim Berners-Lee, James Hendler and Ora Lassila. The Semantic Web. Scientific American, May 2001.

[Br99] David Brin. The Transparent Society: Will Technology Force Us to Choose Between Privacy and Freedom? Basic Books, 1999.

[CHL05] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact E-Cash. Eurocrypt 2005.

[CL02] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. International Conference on Security in Communication Networks (SCN), 2002.

[Fi00] Roy T. Fielding. Architectural Styles and the Design of Network-based Software Architectures. Ph.D. thesis, University of California, Irvine, 2000.

[Ko05] Marja-Riitta Koivunen. Annotea and Semantic Web Supported Collaboration. Workshop on Users Aspect of the Semantic Web (UserSWeb), Heraklion, Greece, 2005.

[MGM08] Luc Moreau, Paul Groth, Simon Miles, et al. The Provenance of Electronic Data. Communications of the ACM, April 2008.

[PL07] PrimeLife Consortium. Privacy and Identity Management in Europe for Life (PrimeLife). FP7-ICT-2007-1, Version 3, 09/10/2007, Grant Agreement GA no. 216483.

[SCW01] Ines Macho-Stadler, J. David Perez-Castrillo, Richard Watt. An Introduction to the Economics of Information: Incentives and Contracts. Oxford University Press, 2nd edition, 2001.

[vA06] Luis von Ahn. Games with a Purpose. IEEE Computer Magazine, June 2006.