Rich Identity Provisioning

Jos van Dijk
jos.vandijk@os3.nl

University of Amsterdam
Faculty of Science, Informatics Institute
System & Network Engineering

July 10, 2014

Abstract

A digital identity represents an entity on the Web. Users expect privacy and security together with ease of use regarding control of disclosure of personal information demanded for access to protected resources. Service providers however, expect dependable identities that allow for personalization.

Today's increasing activities on the Web urge for context-specific identities that ensure on the one side minimal disclosure of private information and rich sharing on the other. Rich Identity Provisioning implies a user-centric identity management environment.

In this report an architecture is proposed that fits best to the requirements that apply to Rich Identity Provisioning. Secondly, open source components are evaluated to determine whether these components fit to the architecture.

Contents

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Research Questions . . . 1
1.3 Related Work . . . 2
1.4 Scope . . . 2

2 Literature and Technology Review **2**
2.1 Terminology . . . 2
2.2 User Identity Models . . . 3

3 Identity Provisioning **3**
3.1 Goals and Issues . . . 3
3.2 Concepts and Terminology . . . 3
3.3 Identity Provisioning Mechanisms . . . 4

4 Rich Identity Provisioning - Architecture **5**
4.1 User services . . . 5
4.2 Identity Data services . . . 8
4.3 Data Storage services . . . 9
4.4 Access & Policy services . . . 9

5 Rich Identity Provisioning - Open source Components **9**

6 Conclusion **13**

7 Future Work **13**

1 Introduction

Identity (ID) provisoning is a persistent subject. A user, intending to access remote protected resources will be checked for permission. This decision process is called authorization. Access privileges however, are associated with a digital identity. Prior to using the privileges, a user has to prove that she represents the digital identity in question. This process of proving the association of an entity with a digital identity is called authentication.

Today, people increasingly rely on remote resources. Service Providers (SPs), offering services and resources, often demand user's personal information to create a relationship. This user account is site-specific and isolated. In this setup, many accounts have to be maintained due to increasing visits to different SPs. Users experience a cumbersome procedure that provokes wrong behaviour. For example, using short passwords, reuse of credentials and bad protection of credentials are ill-considered actions. Identity fraud is encouraged by this phenomenon. Besides that, identity fragmentation occurs, scattering identity pieces all over the Web.

A second concern involves privacy violation. SPs have great commercial interest in surf and consumer behaviour. Tracking and profiling of users on the Web is common. "Do-Not-Track-me" browser options or add-ons will hardly help as they have no authority.

Users should be in-control of the amount of personal information to be shared with SPs. This amount of information is context sensitive. Transactions in banking or health insurance require different identities from social web access. The main goal of a user-centric identity provisioning environment is to keep the user in control, managing multiple IDs to be used in various contexts.

Section 2 covers general information on topics relevant to this project. The next section discusses the most prominent aspects of identity provisioning. Section 4 presents the proposed architecture that answers the Rich Identity Provisioning requirements, followed by the section that evaluates relevant open source components to determine whether they fit to this architecture. Finally this report is concluded by answering the research questions and recommendations for future work.

1.1 Motivation

Privacy and security affect all users surfing the Web irrespective of their awareness. Identity fraud (or identity theft or impersonation) is widespread and comes in many flavours.[1]. The impact is severe due to its diversity. An identity provisioning system that contributes to a better understanding of the risks and impact of careless identity management and provides means to improvement is valuable.

1.2 Research Questions

For this project the following research questions are put

1. *What architecture fits best to a user-centric identity provisioning system regarding Web access?*

2. *What open source components fit to such a system?*

[1] http://en.wikipedia.org/wiki/Identity_theft

1.3 Related Work

Identity provisioning, which is a part of identity management is a vast and persistent topic. A lot of research has been conducted on issues regarding identity provisioning.

The counterpart of an authority backed user-centric solution is a decentralised environment with independent self-determining entities that control their own data and identity.[1]

1.4 Scope

This project intends to propose an architecture that provides a polyglot environment, solutions that are context-based. The open source components that will be evaluated fit to this principle. An 'all-in-one' enterprise solution doesn't fit to all contexts but contributes at most. Such solutions are out of scope as they are not within reach of all users.

2 Literature and Technology Review

2.1 Terminology

The *Internet*, a global network of interconnected networks, makes many resources and services available. Protocols like ftp, smtp, bittorrent and xmpp are used to access such resources and services. The *Web* is a different approach of accessing Internet resources. It uses the http protocol, browsers, web pages and hyperlinks to access information.
Some resources and services are protected and users must be granted permission.

Identification is the process of distinguishing entities by using *identifiers*. An identifier value is unique within its context. A *personal* identifier links to the real entity. An example of a personal identifier is a passport number. An *anonymous* identifier represents an identifier that can't be linked to a personal identifier. A random string of characters is an example of an anonymous identifier. An anonym is normally used once. A *pseudonym* is an anonym that is being reused.[2]
Attributes represent characteristics of an entity. Attribute values are not unique within their context. An example of an attribute is the eye color. Multiple identifiers, accompanied by multiple attributes, might be used to create a single *identity*.[2]

Authentication is the process of proving an identifier. Identity authentication proves that an identity is associated with an entity. Showing a passport that contains a person's photograph provides this proof. Anonymous authentication doesn't disclose a user's real identity.

Authorization is the decision of granting access to resources, based on authentication.
Anonymous access is in use where resources don't have to be protected or where anonymity is enforced by law.

In a digital environment, a paper or plastic passport is ill-suited; *digital identities* are used here. A well-known means to prove the association between a digital identity and an entity is using a digital certificate.[2] The certificate represents an identifier. Conceptually, a digital identity doesn't differ from a non-digital identity: both represent a set of identifiers and/or

[2] http://www.tldp.org/HOWTO/SSL-Certificates-HOWTO/x64.html

attributes that builds up an identity. Using a username and password however, is the most widespread authentication method.

Identity, anonyms and pseudonyms relate to *privacy*: *"The right to privacy is our right to keep a domain around us, which includes all those things that are part of us, such as our body, home, property, thoughts, feelings, secrets and identity. The right to privacy gives us the ability to choose which parts in this domain can be accessed by others, and to control the extent, manner and timing of the use of those parts we choose to disclose."*[3]

User identity models should take privacy concerns into account.

2.2 User Identity Models

In a setting where authorization is applicable, many models exist. A taxonomy is given in [4] ranging from the isolated model in which a separate user identity is stored (called silo) in every single SP to *federated identity* where a group of SPs share and trust entitlements within a federated domain and *centralised identity management* where only a single Identity Provider (IdP) exists. Such an IdP functions as a Trusted Third Party (TTP). Federated identity allows for Single Sign On (SSO); once authenticated, a user is granted access to multiple services without user interaction regarding authentication.
Contrary to these authoritative models, the user-centric model resides. In a user-centric model a user has control over her identities. This characteristic is considered one of the 'Laws of Identity'.[5] Separation of user identities will be discussed in subsection 4.1.

3 Identity Provisioning

3.1 Goals and Issues

In a consumer - producer (user - SP) setting, both parties have different goals. Users accessing services which have been made available by SPs demand privacy, security, usability and control.[4, 6, 7] SPs however, rely on accurate user information in order to be able to authorize properly on the one hand and provide personalization on the other.[3] This difference in concern might cause an area of possible conflict.

3.2 Concepts and Terminology

To be 'in control' is the main characteristic of a user-centric identity management system. Users must have control over which ID to use and what attributes to disclose to their counterparts. Web activities in which resources are accessed happen in many contexts. These contexts may differ in the amount and kind of personal information that will be disclosed. A bank transaction for example, requires different identity information compared to the information required in accessing social media like Facebook. *Partial identities* is a concept that meets this differentiation among identities thereby reflecting a single entity as shown in figure 1 taken from[4]. Partial Identities allow for context depending IDs.

Representation formats need to be agreed upon by parties to be able to correctly interpret the exchanged attributes. *Vocabularies* or *Ontologies* are used to define the concepts (classes)

[3] http://www.marketingcharts.com/wp/online/consumers-say-they-want-more-relevant-website-experiences-35416/

[4] http://www.fidis.net

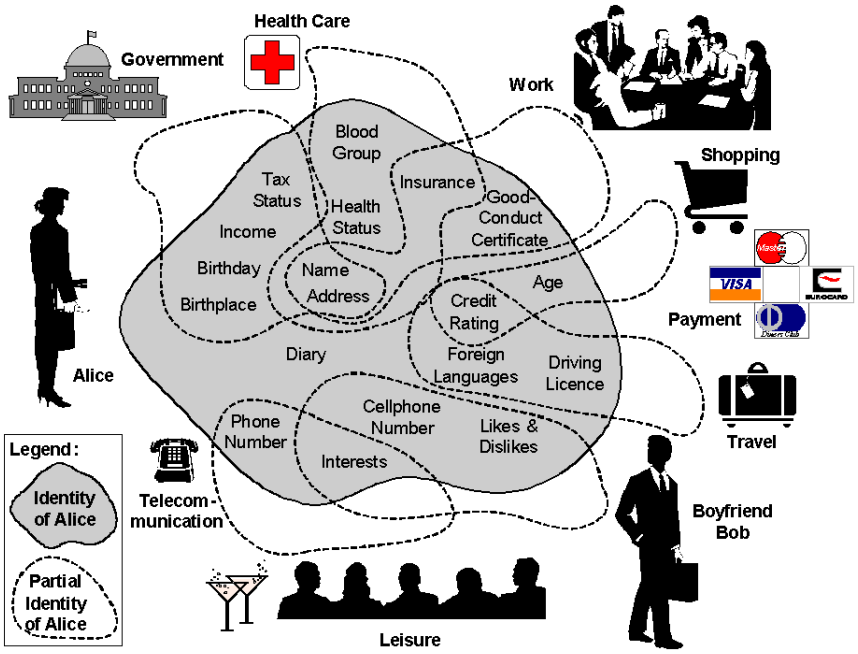

Figure 1: Partial Identities allow for context depending IDs.

and relationships used to describe and represent an area of concern.[5] In this context, the area of concern is a person. Ontologies are powerful as they define both attributes of the entity itself and relationships to other entities. Figure 2 shows a graph of the 'FOAF' ontology.

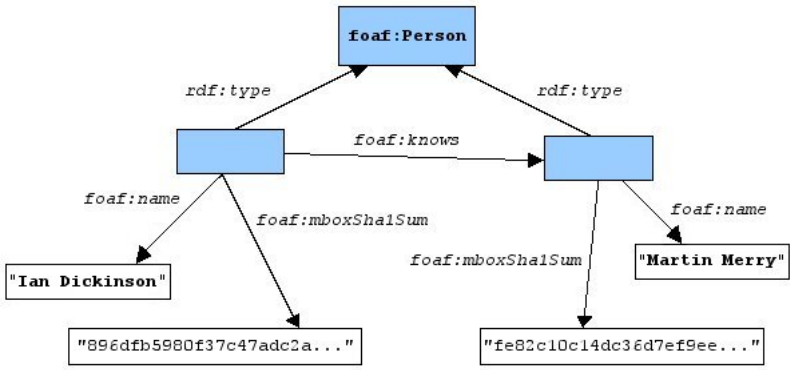

Figure 2: An ontology describes a context.

3.3 Identity Provisioning Mechanisms

Two distinct use cases in identity provisioning have to be recognised: a setting in which minimal disclosure of attributes is intended i.e. just authentication on the one hand and rich sharing on the other. Both approaches have their own characteristics.

The *minimal disclosure* approach aims for attribute stripping as far as possible. An example of a user-centric solution is using existing email addresses. Email addresses have a context-

[5] http://www.w3.org/standards/semanticweb/ontology

specific structure (name@work.com, name@home.com, name@membership.org) and there is no need for new identifiers. The potential association between an email address being used and the owner's entity depends on the email address composition. Authority issued email addresses might reveal one's real identity i.e. `jos.vandijk@zuyd.nl` but personal email addresses won't. Minimal disclosure is not synonymous with anonymity. Using techniques like TOR and GNUnet will improve anonymity.
TOR[6] represents an overlay network of 'Onion Routers' that provides random and encrypted paths between user and SP. It is assumed that an adversary has no view of the entire network. Together, TOR and an anonymous email service provide an environment that disables identification and mitigates tracking and profiling.
GNUnet[7] represents a secure Peer-to-Peer filesharing network but differs from TOR by assuming that an adversary might be in control of part of the network. Anonymity is accomplished by using public key identifiers and making the user's traffic indistinguishable from the traffic that she routes for others.

The *sharing* approach requires an environment that allows for rich data exchange. Using an URI is obvious as one can be found and referenced easily and social relations on the Web can be established.
Implementations of both approaches show up in section 5.

4 Rich Identity Provisioning - Architecture

Identity provisioning aims for both dependable identities and privacy consideration of the associated entities. Rich identity provisioning extends this principle with the support of context depending identities.
An architecture that accommodates Rich Identity Provisioning (RIP) is presented in figure 3. The RIP architecture, comprising four components, mediates between the user and the protected SP resources. An Identity Provider is involved somehow. A brief functional description is given below and depicted in figure 4. Details are discussed in the next subsections.

`User services` holds all tools that a user needs to manage her (virtual) identities as well as to control disclosure of identity information. Means to verify proper operation are included.

`Identity Data services` includes identity representation and consistency.

`Data Storage services` comprises all data needed by other services.

`Access & Policy services` contains all protocols needed to perform the necessary operations.

4.1 User services

In a user-centric identity provisioning system, a user should be `in-control` of her identities. This part of the `User services` component affects almost all remaining parts as illustrated in figure 5.

`inControl` of `Identity`. A user must be able to manage her identity information, represented by attributes, in a store. Two major implementations exist.
cloud: a cloud service is offered to the user. A user might run her own server or rely on a provider (TTP). An example of such a solution is Higgins' Personal Data Store (PDS)[8]. A user manages the store using a GUI. Partial Identities are created by using *cards*. This mechanism is part of the component `Identity Data services` and is discussed there. Vocabularies are

[6] https://www.torproject.org/
[7] https://gnunet.org/concepts
[8] http://www.eclipse.org/higgins/

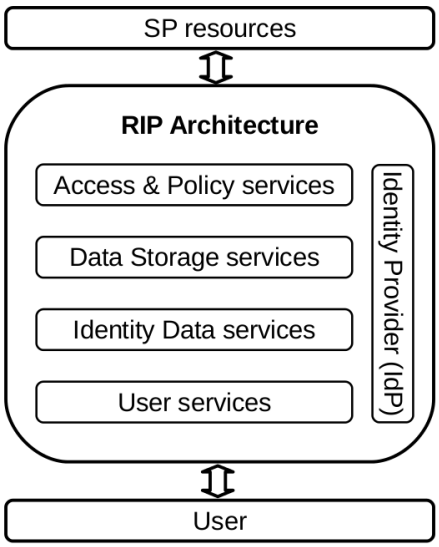

Figure 3: Global architecture of a Rich Identity Provisioning system.

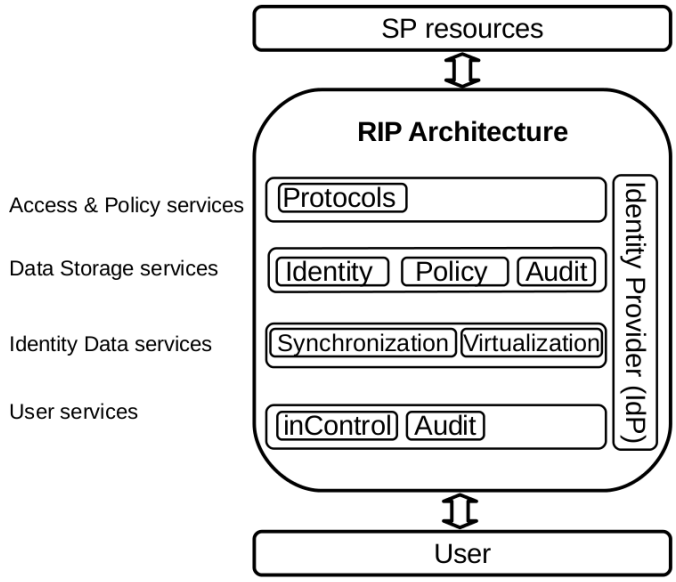

Figure 4: Services of the RIP-architecture.

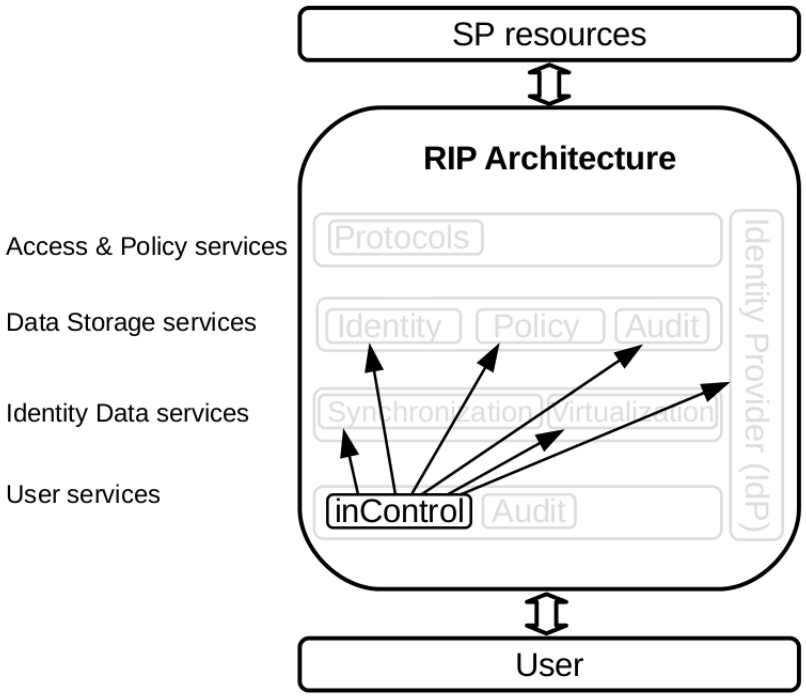

Figure 5: A user-centric approach.

used to meet interoperability requirements, an export facility allows for 'offline support' and 'managed IDs' is supported. 'Offline support' is useful in case of connection failures to the cloud service. 'Managed IDs' occur where IDs are issued (and controlled) by an authority.

card: a smartcard provides a secure means to store personal information. Smartcards introduce a trusted module[8] and are an obvious solution to offline support as well.[9]. Combining both approaches is exemplified by figure 6. In case a smartcard is no option, a nfc (near field

communication) crypto tag may come in place.

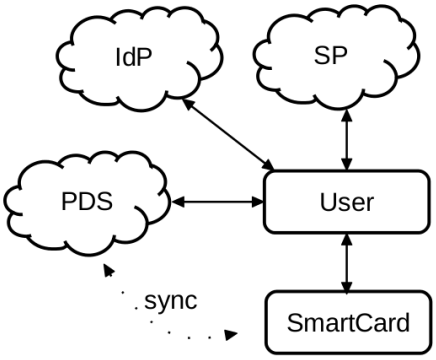

Figure 6: `Identity` solution comprises a smartcard and cloud storage.

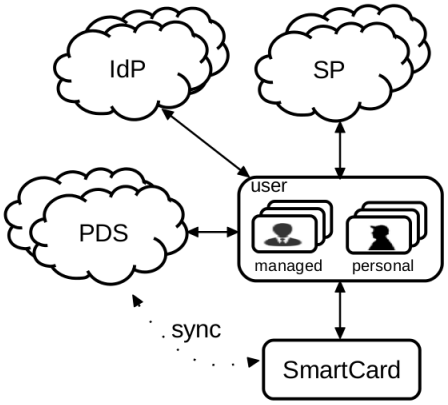

Figure 7: Managed cards versus personal cards

`inControl` of `Virtualization`. Representation of identity information is closely related to the previous topic. Minimal disclosure of personal information is achieved by Partial Identities, holding a context-specific set of attributes. A common approach to manage Partital Identities is using *cards* as discussed before; every single Partial Identity is represented by a card as shown in figure 7. In an integrated environment, both personal and managed card exist. Personal cards are created and owned by the user, managed cards are issued by providers. Government, health and insurance companies are representative contexts for managed IDs.
A second use case for virtualization concerns device virtualization. A user may apply different devices for accessing the Internet i.e. PC, laptop, tablet, smartphone. A GUI provides an interface for managing multiple physical devices, mapping cards onto devices and presenting a single virtual device. This approach, shown in figure 8, contributes to enhanced privacy as disclosure of device attributes is controlled.

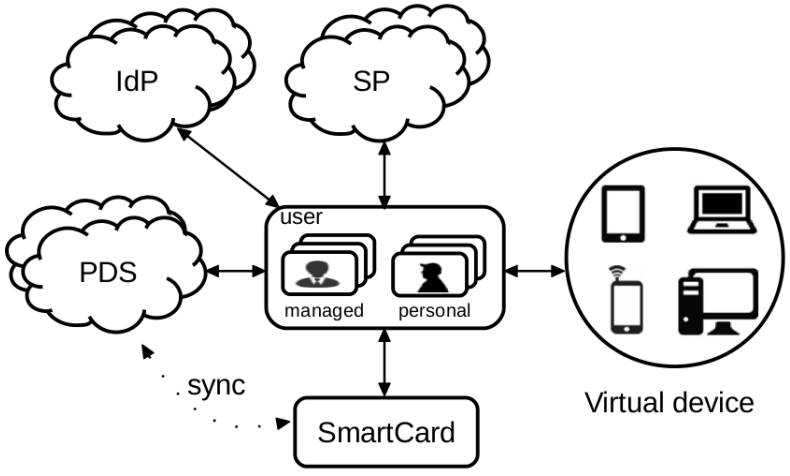

Figure 8: Privacy enhancement by device virtualization

`inControl` of `Policy`. Policies that control disclosure of attributes are stored here. Applying policies enhances user awareness regarding privacy. In a minimal disclosure setting, a SP might send a 'AttributeQuery' request, containing the list of attribute values that are demanded for

service access. The proposed set of attributes is evaluated against all available Partial Identities i.e. cards. Cards that meet the requirements are shown in the GUI. A wise policy would be user confirmation of the card to be used. Using a PIN might be an additional policy.
In case no existing card answers the SP requirements, a new virtual ID might be composed out of the available attributes and shown to the user. It's up to the user to decide to add and use this new ID and to apply additional policies. Defining trust levels between digital identities and devices helps in thoughtful policy implementation.[7]

`inControl` of `Audit`. Means to audit the proper disclosure (and use) of information must go hand in hand with enforcing policies. Disclosure of attributes should be logged and put into a context: which provider requests what attributes? This context has to be verified against the enabled policies. A second audit aspect is the use of attributes. A user should be informed about the use of her credentials by the SP. A secure logging module at the SP collects all user attribute related processing. The results are audited by a TTP. Privacy violations will be reported to both the user and the SP.
Compromized IDs are common today. Detecting misuse of IDs depends heavily on audit capabilities. A compromised ID should be revoked as soon as possible. Deanonymization by the TTP audit is needed in case of criminal activities. Logfiles must be handed over to authority entities (law enforcement). Policy management and audit provisioning are shown in figure 9.

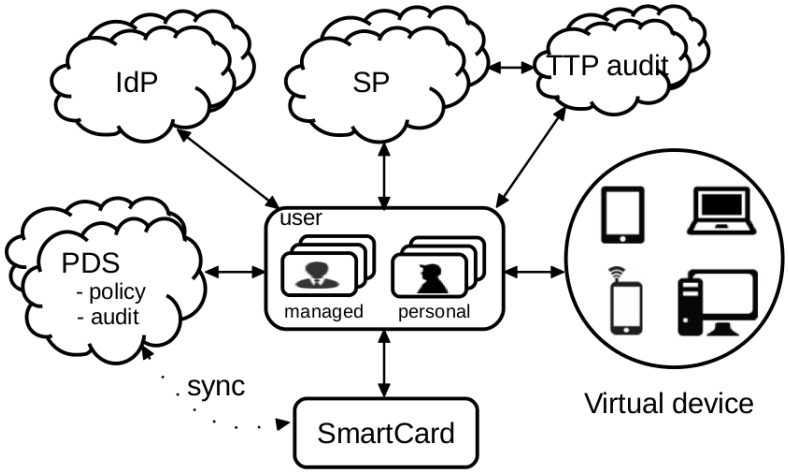

Figure 9: Policy management and Audit provisioning

4.2 Identity Data services

The `Identity Data services` component comprises two building blocks: `Virtualization` and `Synchronization`. Identity and device virtualization are discussed in the `inControl` section.
`Synchronization` deals with keeping the data stores `Identity, Policy` and `Audit` up to date. As mentioned in the previous section, a compromised ID should be revoked instantly. But also an employee leaving a company should be refused access to the company's resources according to pending policies. Processing such events should occur event-based triggered.
A second approach to keep the data stores consistent is achieved by a Validating process.[8] A smartcard could be lost or stolen. By setting expiration dates, such a card will be automatically disabled. Frequent revalidation assures consistency.

4.3 Data Storage services

The purpose of the data storage services is clear and discussed in the `inControl` section. Characteristics of this storage are secure, available and interoperable.
Security may be obtained by using secure storage techniques like a crypto container.
Redundancy improves availability. By synchronizing between a secure cloud store and a smartcard, availability is improved. This solution doesn't apply to all use cases. A smartcard can't be connected to a smartphone instantly. Replacing the smartcard with a nfc tagged crypto processor might be an acceptable substitute.
Interoperability is related to data representation. Synchronization demands for standardized data formats.

4.4 Access & Policy services

The glue that connects the various components and external parties together is represented by the `protocols`. SAML 2.0[9], SCIM 1.1[10] and XACML 3.0[11] are well-known protocol specifications in the context of identity provisioning. Open source solutions like Shibboleth Identity Provider[12] and WSo2 Identity Server[13] implement these protocols. These specifications however, are out of scope as they fit to the 'all-in-one' solutions.

5 Rich Identity Provisioning - Open source Components

The diversity of contexts in services as well as devices require tailored solutions. This section evaluates relevant open source implementations of such solutions. Due to the unique characteristics, they fit to different contexts. References to the proposed architecture are provided where appropriate.
Subsection 3.3 denotes two distinct use-cases: *minimal disclosure* and *sharing*.

Minimal disclosure solutions

`BrowserID (PERSONA)` implements the BrowserID specification[14]. In short, a user's email address is used as an identifier. Email addresses are suitable for authentication as the owner has to confirm to be in control of this email address by activating a link that is sent to the mailbox by the IdP. The benefit of this solution is the use of existing identity information. In a BrowserID-enabled environment, a single password is needed to manage the entire set of email addresses to be used. For each email address, an asymmetric cryptographic key pair is generated and associated. Email address and associated public key are stored at an IdP. This TTP responds to an authentication request by issuing an *Identity Certificate*: a signed statement that confirms the association of the owner and the email address. Prior to this confirmation, the user authenticates by using her private key. Authentication is processed by the browser. A message diagram is shown in figure 10.
This solution is user-centric as the user is in control of the email addresses to be used. Multiple addresses are allowed to serve the various contexts. No additional identifiers nor attributes are

[9] http://saml.xml.org/saml-specifications
[10] http://www.simplecloud.info/
[11] https://www.oasis-open.org
[12] https://shibboleth.net/products/identity-provider.html
[13] http://wso2.com/products/identity-server/
[14] https://github.com/mozilla/id-specs/blob/prod/browserid/index.md

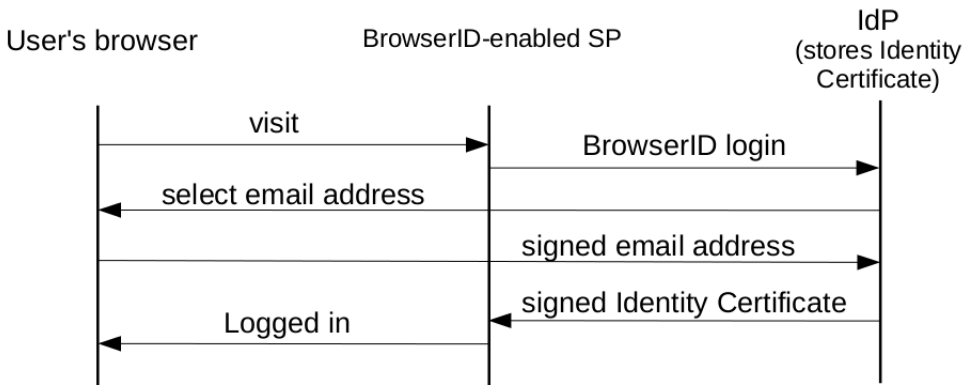

Figure 10: BrowserID Message Diagram.

required. Persona claims privacy by only sharing an email address for login purposes.[15] Figures 11 and 12 show how this environment looks like.

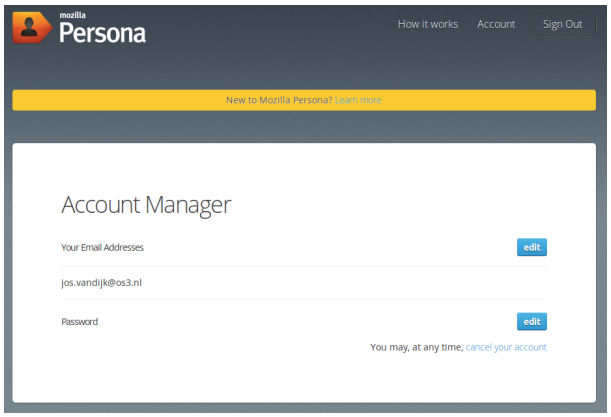

Figure 11: BrowserID account.

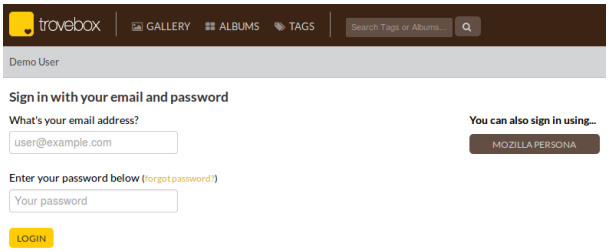

Figure 12: BrowserID enabled Website

`SQRL`[16] and `TiQR`[17] are authentication protocols using QR-codes. Both solutions are designed to authenticate using mobile devices that allows for cross-device authentication: the device that requests the service differs from the device that is used for authentication.
TiQR apps are available for Android and iOS. A user needs to create an account on TiQR-enabled websites. Upon account creation, these sites present a QR code representing the identifier to be read by the mobile device. A PIN code is associated with this account. A user attempting to login will scan a QR-code, now representing a challenge, and responses using her credentials. From a user's point of view this is a convenient way of authentication.
A major drawback is the need for new identifiers. Account creation for every single site provokes reuse of credentials. As this approach produces silos, it doesn't fit to the RIP architecture. SQRL however, doesn't use any pre established relationships. For every SQRL-enabled site, the client generates an (site-specific) asymmetric key pair. This approach improves anonymity (at the application layer). As creation of new identifiers is not the case here, SQRL differs completely from TiQR. SQRL fits to the architecture and is user-centric for the minimal disclosure

[15] http://www.mozilla.org/en-US/persona/privacy-policy/
[16] https://www.grc.com/sqrl/sqrl.htm
[17] https://tiqr.org/

use case. A detailed comparison between these two protocols is presented in[10]. As SQRL is published rather recently, no mature implementations are provided yet.

`U-Prove` is a crypto technology owned by Microsoft.[18] It is the underlying architecture of the Claims-based identity implementation, deployed in Windows.[11] A claim is a synonym for attribute. A token is made up of a set of attributes and digitally signed by the issuer. By composing different sets of attributes, users control the amount of disclosed personal information. Tokens are considered unlinkable due to the use of cryptographic wrapping of the attributes. This mitigates profiling and tracking. Microsoft released the core U-Prove specifications under the 'Open Specification Promise'[19] However, the released SDKs are not completely independent of proprietary software.[20] U-Prove can't be considered being an open source solution.

OpenID is a specification that associates an URI to an entity. A user creates an OpenID account by selecting an OpenID provider. This single ID is to be used for all visited OpenID-enabled SPs (called Relying Party). The OpenID provider mediating between user and SP is called the TTP. The SP's website shows a form that allows for OpenID authentication. The user applies her URI which is examined by the SP and the request is redirected to the OpenID provider. The user authenticates to the OpenID provider which in turn confirms her identity to the SP. As the specification doesn't prescribe any authentication methods, a user might use multiple methods to serve different contexts. OpenID is widely adopted now. A detailed message diagram is presented in figure 13 taken from[10]. A comparison of characteristics among `SQRL, TiQR` and OpenID is discussed in[10] as well. OpenID fits to the RIP architecture: a user might use multiple providers, is in control of her identities and authentication methods. Users that prefer to run their own OpenID Identity Provider are able to play this role.[21]

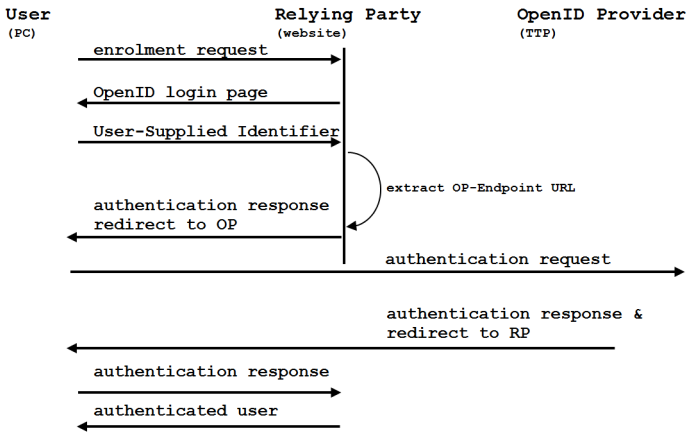

Figure 13: OpenID Message Diagram.

Sharing protocols.
Protocols intended to share rich identity information tend to use URIs because these are readable and discoverable. Examples of valid URIs are: `http://www.os3.nl` and `mailto:jos.vandijk@os3.nl`.

[18] http://research.microsoft.com/en-us/projects/u-prove/
[19] http://www.microsoft.com/openspecifications
[20] https://uprovecsharp.codeplex.com/
[21] http://wiki.openid.net/w/page/12995226/Run your own identity server

`WebFinger` is a protocol (`rfc7033`) which can be used to discover information about entities on the Web using standard http methods.[22]. Users provide personal information intended to be shared, like profiles, among other entities. The SP involved marks this information for publication via WebFinger. A client issuing a WebFinger request, by using the entity's email address receives a document that may contain link relations, attributes and other information. An email client might retrieve a user's vcard. Google's `@gmail.com` addresses are accessible for `WebFinger` as shown in figures 14 and 15.

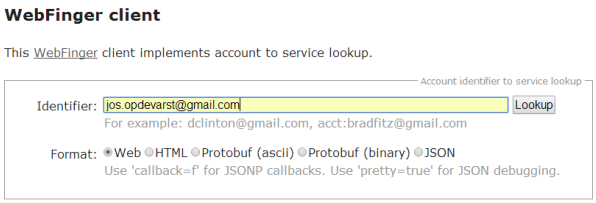

Figure 14: WebFinger query.

Figure 15: WebFinger response.

How well this solution fits to the RIP architecture depends on the 'level of control' a user has. A user running her own WebFinger server is in-control. Using a corporate server is comparable to the 'managed' cards, discussed in subsection 4.1.

`webID` uses an URI to uniquely identify an entity.[23] It provides a central repository for all user information (attributes). This user-centric approach resembles the Higgin's PDS. An asymmetric key pair is linked to this WebID to assure secure operation. A Web of Trust is built by using vocabularies (i.e. `FOAF`) to create relations (i.e. linking information).
`WebID` is able to disclose rich identity information. An example of the available identity space provided by a WebID provider is shown in figure 16. WebID fits to the RIP architecture.

A summary of the evaluated open source components is presented in figure 17.

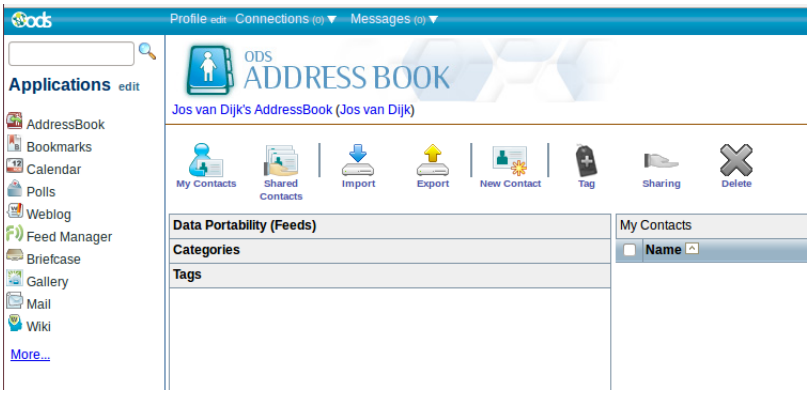

Figure 16: Open Data Space providing a WebID data store.

[22] http://tools.ietf.org/html/rfc7033
[23] http://www.w3.org/wiki/WebID

Implementation	Identifier used	Fits to RIP	User centric	Minimal Disclosure	Rich Sharing
BrowserID	e-mail address	√	√	√	
SQRL	site-specific key	√	√	√	
TiQR	QR code	-	√	√	
U-prove	key	No open source	No open source	No open source	No open source
OpenID	URI	√	√	√	
WebFinger	e-mail address	√	√		√
WebID	URI	√	√		√

Figure 17: Overview of evaluated open source solutions.

6 Conclusion

During this project an architecture has been composed that meets the requirements on a user-centric identity provisioning system.
This architecture provides mechanisms to ensure that the user is in-control. Data stores are controlled by the user, management of context-based personal virtual identities and devices included. Association of various Identity Providers to these context-based virtual identities completes user-centric operation.

Open source components are evaluated to determine how well they fit to the proposed architecture. Results show that solutions fit to various contexts:

solutions designed for *minimal disclosure* of personal information

solutions designed for *rich sharing* of personal information

solutions designed to be used by mobile devices

solutions designed to apply context-based authentication methods

The focus of this evaluation is context. Other relevant characteristics like security, privacy and anonymity are not taken into account.

Offering users a polyglot environment i.e. context-based solutions, brings additional responsibility. All offered solutions must meet requirements regarding security, privacy and anonymity despite the fact that these requirements depend on the context as well.

The evaluated open source solutions cover only parts of the proposed architecture. In a user-centric environment a user should get feedback on proper operation. Audits on applied policies are part of identity provisioning.

7 Future Work

Open source solutions that are part of the offered polyglot environment, must be evaluated on other relevant aspects like privacy, security and anonymity.

User-awareness of proper management of virtual identities, devices and Identity Providers must be educated/integrated. The role and capabilities of TTPs might be fuzzy.

Not all evaluated solutions are 'active'. Currently, no implementation of SQRL is available. Developments in this area should be watched.

References

[1] Mark McLaughlin, Gerard Briscoe, and Paul Malone. Digital identity in the absence of authorities: A new socio-technical approach. *arXiv preprint arXiv:1011.0192*, 2010.

[2] L Jean Camp. Digital identity. *Technology and Society Magazine, IEEE*, 23(3):34–41, 2004.

[3] Yael Onn et. al. *Privacy in the Digital Environment (Haifa Center of Law Technology, Niva Elkin-Koren, Michael Birnhack, eds.)*. 2005.

[4] Audun Jøsang and Simon Pope. User centric identity management. In *AusCERT Asia Pacific Information Technology Security Conference*, page 77. Citeseer, 2005.

[5] Kim Cameron. The laws of identity. *Microsoft Corp*, 2005.

[6] Tewfiq El Maliki and J-M Seigneur. A survey of user-centric identity management technologies. In *Emerging Security Information, Systems, and Technologies, 2007. SecureWare 2007. The International Conference on*, pages 12–17. IEEE, 2007.

[7] A Matos. Gap analysis and architecture requirements. *SWIFT Deliverable*, 202, 2008.

[8] Jan Vossaert, Jorn Lapon, Bart De Decker, and Vincent Naessens. User-centric identity management using trusted modules. In *Public Key Infrastructures, Services and Applications*, pages 155–170. Springer, 2011.

[9] Ronald Marx, Hervais Simo Fhom, Dirk Scheuermann, Kpatcha M Bayarou, and Alejandro Pérez. Increasing security and privacy in user-centric identity management: The idm card approach. In *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2010 International Conference on*, pages 459–464. IEEE, 2010.

[10] Jos van Dijk. A closer look at sqrl. 2014.

[11] Claims-based identity for windows v2. 2009. `http://www.davidchappell.com/writing/whitepapers/Claims-BasedIdentityforWindowsv2.pdf`.