A design to extend the claims identity model to include geo-data

INFORMATION CARDS AND A DESIGN TO EXTEND THE CLAIMS MODEL TO INCORPORATE GEOLOCATION

Matthew Evans

A thesis submitted in partial
fulfillment of the requirements for the
degree of

M.Sc. (Information Technology)

University of Cape Town

2009

Supervised by: Dr Andrew Hutchison

Date _____

A design to extend the claims identity model to include geo-data

UNIVERSITY OF CAPE TOWN

ABSTRACT

INFORMATION CARDS: A DESIGN TO EXTEND THE CLAIMS MODEL TO INCORPORATE GEOLOCATION

By Matthew Evans

Supervisor: Dr Andrew Hutchison

The rapid adoption of the internet has occurred despite the lack of a ubiquitous identity meta-system. The status quo is a patchwork of proprietary security systems. A number of security issues have arisen as a result which threaten to lead to a loss of trust in the internet, and may limit the scope of applications built on it; effectively constraining the potential of the internet as a platform for business and services. Current initiatives by a broad consortium of industry leaders promise a vastly improved landscape with a set of interoperable protocols and systems, built on open specifications, and guided by a set of core identity principles, enabling a more secure online experience.

Simultaneously there have arisen a large number of location aware web application and services which detect and use a user's location to enhance their application experience. These advances, although useful, present new security and privacy issues.

This paper investigates the operation of one of the new identity technologies, information cards, and proposes extensions to the existing supported schemas to incorporate recent advances in geo-location technology. The proposal is supported by reference to existing open source implementations.

A design to extend the claims identity model to include geo-data

Table of Contents

LIST OF FIGURES

(All data-flow, and swim-lane diagrams are the author's own)

1. Acknowledgements

I would like to thank the following individuals for their willingness to answer queries and proposals, and to discuss the issues around security and interesting applications thereof in the context of information cards:

1. Kim Cameron: Identity Architect at Microsoft and a steering member of the Information Card Foundation community who took the time to respond to my emails on the topic, and who affirmed my original idea of the concept of integrating geo-data into the schema defined by the claims security model.

2. Raoul du Plessis: IT Architect at Nedbank South Africa who discussed the issues around Cardspace adoption in the enterprise, and the possibilities of Federation.

3. David Hislop: Mobile development and research at MWEB South Africa for discussing some of the issues around OpenID, and enterprise applications of identity toolsets.

4. Axel Nennker: T-Systems (Germany), and a steering member of the Information Card Foundation community, who helped me with some issues I was having compiling the Openinfocard project to which he also contributes.

2. Introduction

The evolution of distributed computing networks such as the internet has created a problem with identity: the HTTP protocol, upon which much internet based data communication takes place, was designed to be stateless[1]. Between client requests a server process retains no specific client state information. This is reconstructed on each request, using data passed with the request, typically in the form of a cookie header, or stored in memory on the server and accessed via session identifiers passed in the URL or a hidden form field.

This is not of itself problematic; in fact the simplicity, and resource efficiency of the model has contributed in no small measure to the success of the protocol. However the artificial nature of the way in which state (and identity) is maintained through a session impacts the ability of the protocol to incorporate a common tokenized identity 'layer' (in the same way that local network protocols, such as Active Directory, or Novell's NetWare support it).

As a result of this, there has evolved a patchwork of proprietary, custom authentication, authorization and single sign-on solutions, largely based around the paradigm of user name and password, and employing HTTP cookies, our URL based session identifiers. These solutions are often insecure[2] and usually mutually incompatible.

The problems with this status quo are numerous and well documented, and include phishing, password fatigue and poor data management policies. Data is routinely compromised, identity theft is increasingly common and there is a very real threat to the internet as a platform for services and commerce.

The problem was identified some time ago, and a number of proposed solutions have been implemented and attempted. Among them is Microsoft's Passport (originally part of the 'Hailstorm'[3] set of services) which was positioned as a platform agnostic XML based API for

[1] http://tools.ietf.org/html/rfc2616#section-5.1.1

[2] http://conferences.sigcomm.org/sigcomm/2010/papers/sigcomm/p435.pdf

[3] http://www.microsoft.com/presspass/features/2001/mar01/03-19hailstorm.mspx

user authentication and profiling, freely available to any web sites who wished to use it. Although this solution in many respects was very successful (there are millions of daily authentications processed by it), it never became the common standard for web authentication outside the Microsoft stable of web sites and services, and some high-profile early adopters such as http://www.ebay.com dropped support for it relatively quickly[4]. The generally cited reason for this failure is consumer unwillingness to entrust their personal data to a third party without justification for holding it.

As a result of lessons learned from these deployments it became clear that a new initiative was required to establish the identity layer, and numerous collaborations were begun to achieve that goal. Among them were:

- OpenID[5], a method of establishing identity based on proving 'ownership' of a unique resource identifier, such as an internet address The unique qualities of these resources (such as URL's, or extensible resource identifiers) allows it to function as an identifier., providing that a user can establish 'ownership'. The resource then represents the user online. This is conceptually similar to the PayPal system which uses an email address as a globally unique identifier, and indeed PayPal is a corporate member of the OpenId Foundation, along with Google, IBM, Microsoft and Yahoo. The OpenId technology is examined in greater detail in chapter 2.1.3.1.

- The Oasis Security Services Technical Committee (SSTC), in conjunction with the Liberty Alliance (a consortium of industry groups including British Telecom, Intel, Novell, Sun, Oracle, and AOL) in 2002 developed the highly regarded SAML (Security Assertion Mark-up Language 2.0 specification)[6]. SAML is essentially an HTTP and XML based set of specifications which describe identity data exchange scenarios. In addition the Liberty Alliance has been responsible for a number of related specifications, privacy and best practices documentation. They also support an

[4] http://www.crm-daily.com/perl/story/13038.html

[5] http://openid.net/foundation/

[6] http://www.oasis-open.org/committees/download.php/11785/sstc-saml-exec-overview-2.0-draft-06.pdf

interoperability testing program to ensure that organizations which implement the SAML protocol do some in a consistent manner. SAML is discussed in greater detail more in chapter 2.1.3.2

- Information Cards: a more recent initiative by Microsoft, Novell, Oracle, Google, Deutsche Telekom, Intel and others to create a user-centric, standards based meta-system which embraces the current set of technologies, and provides a consistent internet user agent, and operating system integrated interface for managing and supplying multiple identities, each of which is modeled as a set of 'claims' which the user presents, either self-asserted (e.g. 'my nationality is South African') or 'managed' (e.g. a claim managed by a third party. So for instance the Department of Home Affairs might issue a card, containing a claim to the effect that a certain user's Id number is a given sequence of numbers. This claim is 'managed' by Home Affairs. The mechanism for this interaction is discussed in Chapter 2).

At the same time as these advances in identity management have been occurring, an unrelated set of specifications were being developed by the W3C, specifically the geolocation API specification[7], which at the time of writing has the status of an 'informal proposal'. Geolocation is a new technology which is understood to mean the identification of the actual geographic location of a connected user agent (and, by implication, its user). It is usually expressed as the set of latitude and longitude co-ordinates, collectively described as geo-data. A full set of geo-data may well include other 'precision' indicators, as well as data expressing altitude, and heading (i.e. direction of travel). Despite being in its infancy, there is already a client implementation ('Geode') available for the popular Firefox browser[8], offered by the Mozilla foundation. In addition numerous API's and libraries have been published which offer similar functionality: essentially the ability to geo-locate the client user-agent using IP address lookups, or more advanced tools.

This is novel functionality, and many devices and browsers will support the ability to geo-locate oneself, and numerous potential applications will emerge to take advantage of this

[7] http://dev.w3.org/geo/api/spec-source.html

[8] http://labs.mozilla.com/2008/10/introducing-geode/

functionality. However it raises important concerns about privacy and consent. The geode implementation handles these by the use of an 'information bar' which prompts the user to 'share' his geo-data with a requesting site.

This is a workable solution, but it is suggested that this personal data falls squarely within the concept of a 'claim' as understood in the information card metaphor, and that the sharing of it is very well handled by the sending of an information card. The consistency of user experience and the strong encryption used to protect card data are additional good arguments in favour of doing so.

There are some issues with this proposal. Since geo-data is liable to change reasonably often (e.g. with a laptop / mobile device user), there needs to be a seamless, uncomplicated way to synchronize this data to a user's card. Secondly the current claims schema is constrained to a set of claims which do not include latitude and longitude. Thirdly the current paradigm for cards is as a mechanism for authentication, whereas the proposal of this work is to extend that use case to allow for simple consensual data sharing.

This hypothesis of this thesis is that this proposal will enable a set of useful implementations and these will be illustrated with reference to some extensions to an open source, cross-platform information card implementation, the openinfocard project[9].

The paper is structured as three further chapters. The first (Chapter 2) outlines the key technology areas of interest. Firstly, digital identity, being the electronic representation of a user (being either a human being, or in a broader sense, an organization or a system entity, such as a device). Also discussed in this context are the identity domain problems of authentication (the requirement of reliably establishing who the user is), and authorization (establishing what electronic acts a user is permitted to perform). The paper examines traditional solutions to these problems, outlining the established weaknesses, and then moves on to critically discuss a number of currently emerging technologies. There is particular focus on the notion of claims based security, embraced by the emerging Identity meta-system. Thereafter the concept of geo-location (introduced above) is further examined with a discussion of relevant technologies, and related privacy issues.

[9] http://code.google.com/p/openinfocard

A design to extend the claims identity model to include geo-data

Chapter 3 moves on to focus on the claims based security model and schemata, as they are currently implemented, and discusses the implications for extending them with further claims such as geo-data. Example schema changes are outlined, and an argument is made for using the claims model, and the identity meta-system tools for sharing geo-data online. Also discussed are mechanisms for populating, and synchronizing these claim sets in the context of online / browser interactions. The chapter then outlines a series of hypothetical use cases which would be supported by such an implementation. Some attention is given to the question of vulnerabilities, and privacy.

The final chapter brings the research together as set of findings, and focuses the requirements of the second chapter into recommendations. Areas for future research are outlined, and high-level conclusions of the research are documented.

3. Previous Work – an analysis of the landscape

This chapter serves to describe the current situation in terms of the technology and related research in the field, and in this way outline the possibilities of the main proposal of this work.

Although the field of identity is not new, most of the technology around the idea of the identity meta-system, claims based security, and information cards, is in its infancy. Initial implementations of these tools are in their first or beta versions: Microsoft's beta version of its Security Token Server (Geneva) was released in November 2008. [10] Version 1 of the Microsoft Cardspace information card implementation was released in 2006. The IdentitySelector project which is an information card implementation for the Firefox browser was given an alpha release in December 2006[11].

Accordingly this is not as yet a highly researched area (in academic terms at least). Much of the innovation and discussion of the area is emerging from various industry consortiums and work published in a less formal manner, for example online journals and blogs.

In this section the intention is to discuss some of the concepts with reference to these publications and online resources, and to cover the emergence of the requirements for a ubiquitous identity system, and address some of the standards and technologies which are gaining traction. This process will implicitly generate a case for the proposal of this work to extend the current claims subset to incorporate geo-data.

3.1. Digital Identity

2.1.1 The Problem

The internet (and digital environments generally) are context poor environments: human system actors are typically not in the presence of the systems or other human actors with whom they interact. Credentials are easily spoofed, or impersonated. It is difficult without resorting to non-standard technologies (bio-metrics, digital certificates) to unequivocally identity these users (and even these technologies are subject to exploits of various kinds).

[10] http://blogs.msdn.com/card/archive/2008/11/04/geneva-server-beta.aspx

[11] http://www.codeplex.com/IdentitySelector/Release/ProjectReleases.aspx?ReleaseId=18845

The converse of this is that it is relatively easy for a software system (such as a website) to impersonate another one. This is the basis for the phenomenon known as phishing, where spoof sites are establishment to which users are lured, in an effort to steal user credentials which are thereafter used to access protected, high value resources. The risk is theoretically mitigated by various security mechanisms built into modern browsers (SSL support, phishing detection), but statistics indicate that the threat is simply growing The service hosted at http://www.phishtank.com which validates suspected phishing sites reported 11,247 valid phish reports in December 2007[12] and 19, 332 in December 2008 [13] representing an increase of 71 percent..

The result of this is a limited ability to unequivocally identify the users, or the systems involved in digital interactions, and a consequent widespread problem of digital identity.

As an example, in the context of online banking, a credit card transaction is referred to as a 'cardholder not present' / CNP transaction[14]. These are easily repudiated by credit card holders because of the fact that the merchant is typically unable to supply any evidence (beyond the presentation of a primary account number (PAN), which is not conclusive) that the cardholder was in fact the user who initiated a transaction. The merchant does not see the card, nor is it able to validate a signature against that contained on the back of the card. One might, with some justification argue that this evidence is of limited value too, but the fact is that user not present interactions contain little of the context which real world ones do.

2.1.2 Traditional Solutions

2.1.2.1 Passwords

The standard solution to internet based resource authentication and authorization control is a system based on username and password. This is a so-called 'shared secret' security implementation. A number of factors conspire to make this a poor solution for authenticating users[15].

---

[12] "Stats > December 2007", http://www.phishtank.com/stats/2007/12/

[13] "Stats > December 2007", http://www.phishtank.com/stats/2008/12/

[14] http://www.lloydstsbcardnet.com/merchant_account/card_not_present.asp

[15] And the password is...fundamentally insecure, Randall Stross, International Herald Tribune http://www.iht.com/articles/2008/08/11/technology/digi11.php

First is the emergence of phishing, and snooping software (keystroke loggers, packet sniffing on insecure data transports e.g. HTTP) has meant that these credentials are often readily compromised.

Second is so-called 'password fatigue' which describes the phenomenon where users, weary of attempting to maintain unique passwords for multiple online resources simply resort to using one. This fact combined with the fact that web sites and services typically require an email address as the username (because it is guaranteed to be unique) , mean that when this combination is compromised, such a user may have his / her digital identity stolen in multiple contexts. The exercise of compromising these credentials is often made easier because of the fact that many users rely on naive, predictable passwords ('password', '11111' etc.)

Thirdly, inadequate data storage and management processes and security mean that this data is often compromised on the provider side, exposing thousands to identify theft.

Some systems rely on multiple shared secrets, and implement policies that force these secrets to be 'strong', in other words hard to predict. These systems are more robust but by no means totally immune to the issues outlines above.

2.1.2.2 Multi-factor authentication

As a result of the problems of shared secrets identity implementations, multi-factor authentication technologies emerged. The phrase multi-factor means a combination of the following:

1. Personal factors: (" Something you know"), e.g. a shared secret, such as a password
2. Technical factors: ("Something you have") e.g. a phone number, PDA, smart card, or digital certificate.
3. Human factors: ("Something you are") a measurable human characteristic typically bio-metric data such as voice, fingerprint or DNA attributes.

Sites and services offering multi-factor authentication require more than one factor to authenticate a user, or authorize a transaction. So for example a user will be required to provide a password, and a digital certificate, or a one-time password (OTP) sent to him / her in an out-of-band channel e.g. on a registered mobile phone number, or perhaps use a fingerprint reader allowing the user to submit digital bio-metric data.

Multi factor authentication raises the bar, and makes fraudulent impersonation considerably more difficult. As a result the Federal Financial Institutions Examinations Council, an American banking regulation body, issued guidance in October 2005 to the effect that all banks in its jurisdiction should implement multi-factor authentication by December 2006.[16]

While these represent a leap forward, and offer a number of flexible options for a variety of requirements, they are not immune to some of the issues mentioned earlier, and have a few unique problems of their own too[17].

Firstly, multi-factor techniques are vulnerable to so called man-in-the-middle attacks. A typical scenario might work like this: a user is enticed to surf to a phishing site where he enters his username and password, the phishing site uses this data to automate a logon to the real site, this 'initial' logon causes the real site to issue a one-time password to the user's mobile phone (or requires the user to use an issued device to generate one). The attacker then enters this password into the phishing site, which simply passes it on, authenticating to the site. Out of channel solutions do not solve this problem. This is illustrated in Figure 1 below.

[16] http://www.ffiec.gov/pdf/authentication_guidance.pdf

[17] Two-factor authentication: Too little, Too late: http://www.schneier.com/essay-083.html

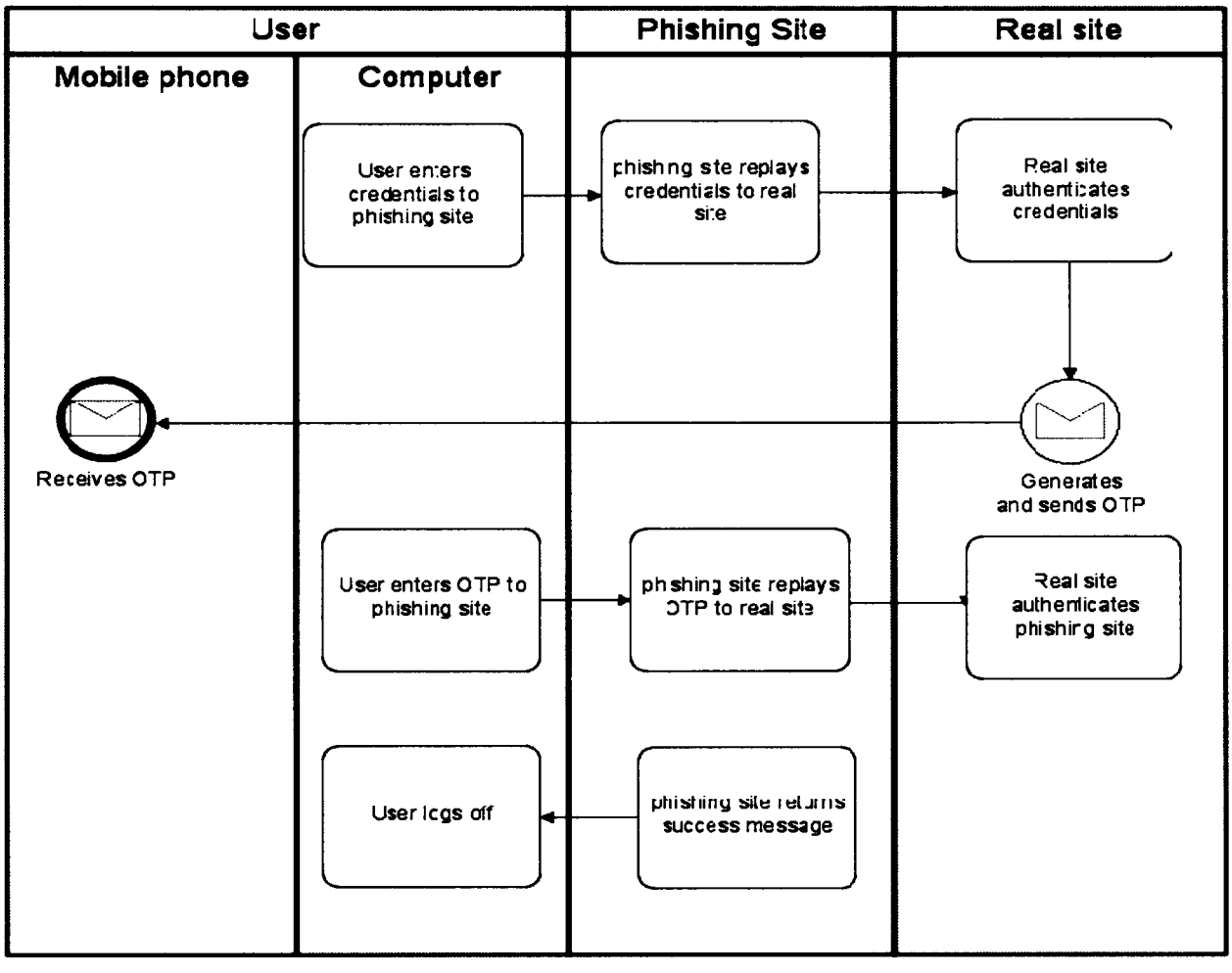

Figure 1: Swim lane diagram of man in the middle attack with multi-factor authentication

Another potential issue attaches to biometric-data, which is easily digitally stored, and cannot be changed after being compromised (like a password could).

The most robust of these technologies impose a significant cost overhead on the process of authenticating users. This may be as a result of additional hardware or software requirements to provide an additional technical 'factor'. This cost will logically, ultimately be borne by the users themselves. Additionally they introduce complexity to the process. Nonetheless, it can be argued that these are acceptable costs in light of the value of the resources being protected

2.1.2.3 The status quo

Clearly, digital identity has many challenges to face. Traditional solutions are often insecure, ineffective, proprietary, expensive and complicated from a user's point of view. In the view of some even the 'best' current solutions will simply be expensive failures[18].

2.1.3 Emerging Solutions

In the light of these obstacles a new set of protocols have emerged. They have similar design goals. They are defined by open specifications[19] to enable interoperability[20], and have limited technology requirements It is important to note that these are protocols; a set of rules defining communication between endpoints to convey identity information. Protocols are by definition open. They are not proprietary technologies which require to be licensed, or which expose only a limited API for public consumption. These are simply useful skeletons for constructing a set of tools to support digital identity.

2.1.3.1 OpenID

OpenID is a set of specifications which support establishing identity with reference to 'ownership' of a resource identifier, generally a universal resource identifier (URI), but also an extensible resource identifier (XRI). An XRI is a "URI-compatible scheme, and resolution protocol"[21] unique identifier issued and resolved by an 'i-broker', in a similar fashion to domain names. An XRI would resolve to something like: "= matthew.evans" (an i-name) or =! 1000.1ab2! 93d2! 8c73 (an i-number, conceptually similar to a machine and router friendly DNS address[22]). These identifiers have conventions for personal, corporate and network identifiers making them extensible to the requirements of the environment. Additionally they are location transparent, in that they are not tied to a specific location (like a postal address), device (like a phone) or service (like an email address). They are therefore resilient to changes to this underlying data, and they satisfy many of the privacy requirements of a digital identity.

---

[18] Two-factor authentication: Too little, Too late, http://www.schneier.com/essay-083.html

[19] The OpenId Specifications, 2007. OpenID Foundation. http://openid.net/developers/specs/

[20] The Identity Selector Interoperability Profile V 1. 5
http://schemas.xmlsoap.org/ws/2005/05/identity/Identity_Selector_Interoperability_Profile_V1.5.pdf

[21] http://www.xdi.org/xri-and-xdi-explained.html

[22] Marc Mercuri, Beginning Information Cards and Cardspace, 2008,New York, Apress, 57

An i-broker hosts a personal contact page, which mediates authentication requests, from parties requesting identity confirmation. The identity standard for XRI is maintained by the Organization for the Advancement of Structured Information Standards (OASIS) [23]

The OpenID specification defines three principal actors: a Relying Party (RP), the User-Agent, and the OpenID Provider (OP)[24]. The Relying Party is a "Web application that wants proof that the end user controls an Identifier", the User-Agent is the "end user's Web browser which implements HTTP/1.1", and the OP is the "An OpenID Authentication server on which a Relying Party relies for an assertion that the end user controls an Identifier".

The protocol defines the format of authentication requests from a relying party, supported encryption standards, security standards and requirements, the format of number-used-once ('nonce') identifiers and the supported scenarios. These scenarios include immediate or deferred authentication, deferred or non-immediate authentication is defined as authentication which requires the end-user to interact with the OpenID provider (typically by authenticating to it). All data transfer takes place using standard HTTP redirects and GET /POST actions.

A typical scenario follows. A user navigates to a website via a browser. When attempting to access a protected resource, the site responds with an authentication challenge, indicating that is recognizes certain OpenID providers (for example, the popular http://www.myOpenID.com).

2.1.3.1.1 Discovery

The user enters his claimed identifier for that provider, and the relying party performs HTML discovery on that identifier. This requires it to ascertain the provider's endpoint URL (i.e. the address at which it will accept HTTP post parameters). This information is contained in an HTML link tag within the head element of the HTML document contained at the claimed identifier. So, the document accessible via HTTP at http://mattevans.myOpenID.com/ contains the following mark-up within the head element:

---

[23] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xri

[24] OpenID Authentication 2.0 – Final, Terminology, http://openid.net/specs/openid-authentication-2_0.html#terminology

<link rel="OpenID2.provider" href="http://www.myOpenID.com/server" />, which is the providers endpoint URL.

2.1.3.1.2 Association

Using this endpoint, the relying party will establish an association with the server, which implies agreeing a signature algorithm (either 160 or 256 bit key length secure hashing algorithm (SHA), with 256 bit preferred), and the session type, which "defines the method used to encrypt the association's message authentication code in transit.". Here either 160 / 256 bit Diffie-Hellman SHA encryption is supported. Once the association is established, the authentication request is actioned.

2.1.3.1.3 Authentication

Typically the user is then redirected to the OpenID provider to complete the authentication. The specification is not prescriptive as to how authentication should occur, but generally this occurs via a username / password login process. However it occurs, once the OpenID provider is satisfied that the end-user is the owner of the URI (the 'claimed identifier'[25]) it issues a positive assertion in the defined format, and POSTS a set of parameters back to the URL defined by the relying party (the 'return_to' parameter of the request) which include the encoded signature, a number-used-once parameter (to prevent replaying of the assertion at the relying party).

At this point the interaction is complete and the user is redirected to the resource. Numerous variations on this use case are possible, but this is a typical scenario.

The figure below illustrates the process outlined above.

[25] OpenId Authentication 2.0 – Final, Request Parameters http://openid.net/specs/openid-authentication-2_0.html#anchor27

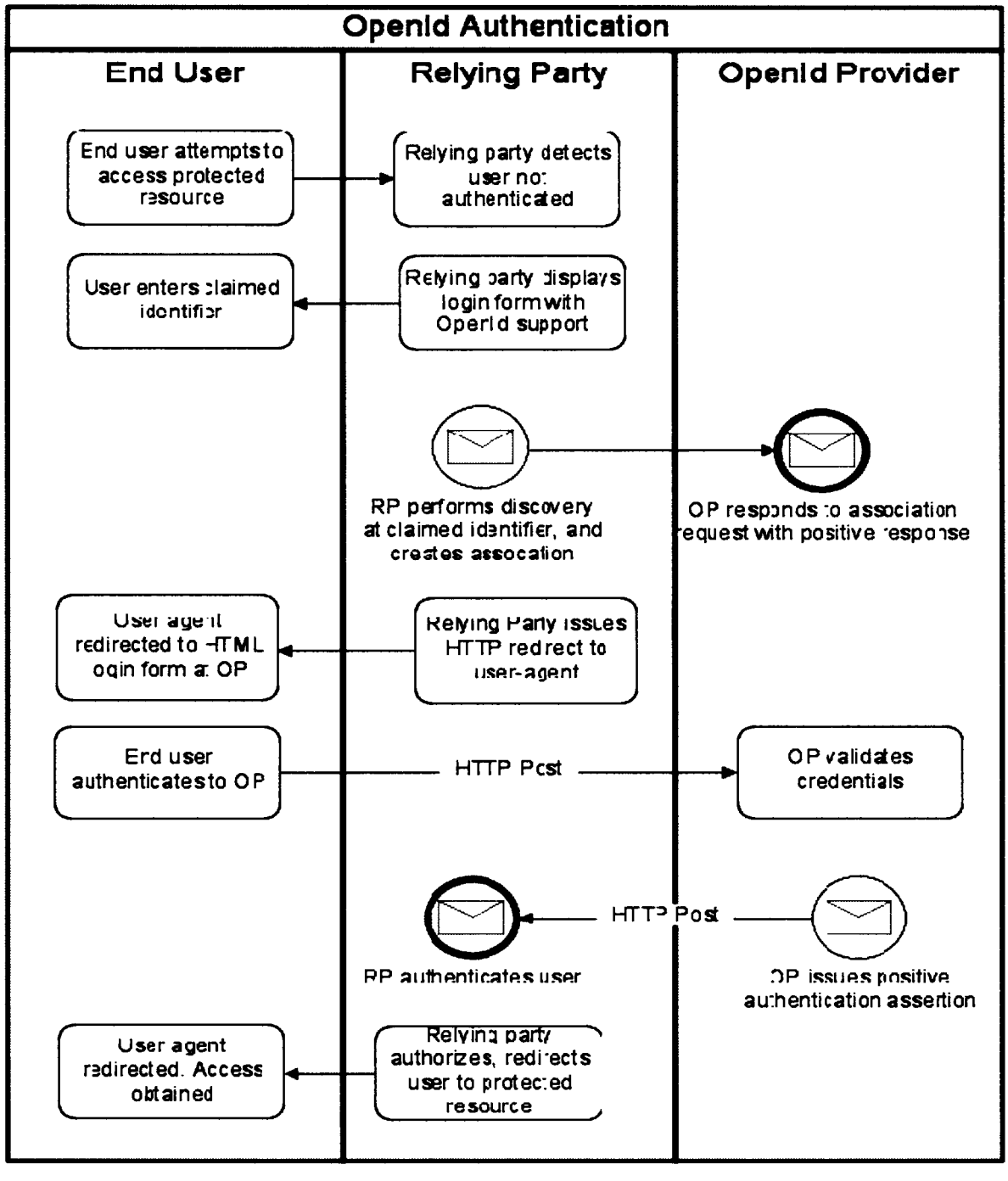

Figure 2: OpenID authentication process

2.1.3.1.4 Problems addressed by this protocol

The OpenID model was conceived to overcome the problem of multiple identities. Simply put, you have one OpenID, and you re-use it everywhere you require to authenticate yourself (wherever it is supported). Additionally you are not required to maintain multiple sets of data at all of the services you use. Your identity is maintained (and secured) by one provider. You are not required to remember usernames and password for a plethora of locations. There are now multiple OpenID providers including Google and Microsoft[26].

2.1.3.1.5 Strengths

The model is particularly potent in that it leverages ubiquitous protocols and standards (DNS, HTTP/S, and HTML) which are well established and easily understood. It is relatively simple to set up an OpenID server (although to become a trusted open identity provider is another matter). It is decentralized and enjoys all the benefits of that (network resilience, reliable availability). Additionally the single identity concept is very compelling to users suffering from password fatigue.

2.1.3.1.6 Weaknesses

The same factors which make OpenID so widespread (as least in implementation terms) operate against it. There are hundreds, if not thousands of Open Id providers[27]. Because the specification admits of multiple different scenarios, and defines support for a variety of different technologies and encryption standards, it is unlikely that any the implementations across providers will be uniform. Supporting multiple implementations is difficult and expensive for relying parties[28], and in the end probably not viable. Additionally, how should a relying party decide which providers to trust authentication assertions from?

The single identity concept is alluring, yet probably flawed. It is in fact preferable to allow multiple identities. One's identity, understood as the digital data which define me uniquely, as far the South African Revenue Service is concerned is not an appropriate set of identity data

[26] Microsoft announces support for OpenId: http://www.identityblog.com/?p=1026

[27] http://openiddirectory.com/openid-providers-c-1.html

[28] OpenID Gets Explained, Maligned, and Dropped http://ostatic.com/blog/openid-gets-explained-maligned-and-dropped

to reveal to, for example, an online shopping site. Multiple identities are the reality of the offline world, and so the preferred online solution should support this.

More significantly, because of its reliance on username / password, OpenID is still vulnerable to phishing[29]. The demonstration site at http://idtheft.fun.de/, illustrates how easy it is to screen-grab HTML from any OpenID provider selected and generate a phishing site on-the-fly, which then harvests one's OpenID credentials. The seriousness of this is exacerbated when considering the implications of a single identity. Once compromised, this identity could be impersonated across multiple different relying parties. This risk is further elevated by virtue of the fact that OpenID providers keep a record of activity by the end user. Here's an example from my OpenID profile (with relying party domains removed)

RECENT ACTIVITY

When	IP Address	Event
6 days, 1 hour ago	41.241.120.221	Approve for http://
6 days, 1 hour ago	41.241.120.221	Sign in by password
2 weeks, 3 days ago		Sign in by session expired
1 month ago	41.241.37.83	Approve for http:
1 month ago	41.241.37.83	Sign in by password

More activity...

Figure 3: OpenID history

This fact would allow a successful phisher to access a user's profile, and see a list of sites which the user had visited, thereby allowing him access all of them, with the freshly phished OpenID. It also represents a privacy issue. Why should an identity provider need to know which sites a user has accessed? This is referred to as 'linkability'.

[29] Gone Phishing, Mike Jones, 2008 http://self-issued.info/?p=73

Clearly there are security and privacy issues with OpenID, and despite a lot of mainstream support, it has come under heavy attack for these[30]. Some propose that it is suitable for low value identity use cases[31]. Proposals to mitigate some of these risks have emerged, and recently been implemented. These will be discussed later in this chapter.

2.1.3.2 Security Assertion Mark-up Language

SAML, developed under the auspices of the Organization for the Advancement of Structured Information Standards (OASIS) "defines a common XML framework for exchanging security assertions between entities"[32]

Conceptually similar to OpenID in that it operates in a framework where an Identity Provider (IdP) makes various identity assertions to a service provider (SP), these claims are released on authentication typically via a user redirect (as with OpenID), although this local authentication mechanism is again left to the provider.

The framework itself is more rigorous, and relies on XML for data exchange.. It has gained a lot of traction in the enterprise with IBM, Sun, Oracle, and VeriSign already providing products which support SAML[33]. These products are subjected to compatibility testing via the Liberty Alliance Project[34], a consortium of industry members whose focus is the "establishment of open standards, best practices and guidelines for federated identity" [35] and identity based web services.

2.1.3.2.1 Design Goals

Version 2.0 of the standard was approved on February 1, 2005[36] and as reflected in the Technical Overview published prior thereto, the drivers were fourfold[37].

---

[30] OpenID: Phishing Heaven, http://www.links.org/?p=187

[31] Why OpenID leads to Cardspace http://www.identityblog.com/?p=923

[32] Security Assertion Markup Language (SAML) 2.0 Technical Overview, Hughes and Maler, 2004, 4

[33] Debunking SAML myths and misunderstandings http://www.ibm.com/developerworks/xml/library/x-samlmyth.html

[34] https://www.projectliberty.org/liberty/about

[35] Marc Mercuri, Beginning Information Cards and Cardspace, 2008,New York, Apress, 55

[36] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml#XACML20

1. To address the limitations of browser cookies. A cookie is an opaque piece of data either stored in a browser's internal memory (a session cookie), or persisted to disk (a permanent cookie). Encrypted data stored in this cookie is often used to store authentication state information. On each protected resource request this state is checked to ensure a user is logged in and entitled to access the resource. This is often used to allow so-called Single-Sign-on (SSO), when a user is accessing a number of different resources. However cookies are only accessible via the browser by the application at the domain name which issued them, so their ability to work cross organizational boundaries was limited.

2. Generally SSO technologies are proprietary, which is limiting.

3. Web services security mechanisms were limited at the time, restricted to the Web Service Extensions (WSE) standards, which were not baked into SOAP.

4. Federation. Federation is the provisioning of trust relationships across organizational boundaries.

2.1.3.2.2 Data Format

As mentioned SAML transfers data in the structured XML (Extensible Markup Language) format. This allows for semi-structured, hierarchical data which is defined by published schemas. XML is well supported and there is wide availability of tools to parse and validate it. This makes for a robust, richly expressive data format which can store data defined in multiple schemas simply by referencing their relevant namespaces.

2.1.3.2.3 Front Channel SAML

The fact that XML is the data format of choice does not imply that in all instances of SAML authentication a web service (SOAP) interface is invoked. SAML 2.0 explicitly added support for HTTP redirect / POST binding. This is described as a front channel exchange. All interactions are mediated by the end users browser which displays XHTML forms, which

[37] Security Assertion Markup Language (SAML) 2.0 Technical Overview, Hughes and Maler, 2004, 5

include the defined request and response XML tokens in hidden form fields. Transport level security is mandated in the form of SSL 3.0

This flow is defined as the Web Browser SSO Profile[38], and a typical interaction would proceed as follows:

A user requests a protected resource via a browser. The relying party supports SAML 2.0. The relying party detects that the user is not checked in, and serves a XHMTL[39] form to the user. This form contains a hidden form field with the identifier 'SAMLRequest'. This value of this field is the encoded XML for a SAML request. Here is very basic example of such a request.

```
<SAMLRequest RequestID="urn:SAMLID:17862301" Version="100">
  <AssertionID>urn:SAMLID:29100231XA</AssertionID>
</SAMLRequest>
```

This SAMLRequest is requesting an assertion with a defined identifier[40]. An assertion is simply a statement issued by an identity provider about some aspect of the digital subject. It may be an attribute statement, expressed as name-value pair, or an authentication statement confirming that the subject authenticated at a specific time, or a combination. The assertion returned is defined by the request assertion identifier. The structure of requests, responses and assertions are all defined in schema managed and hosted by OASIS.

Instead of requesting a specific assertion id, the request might instead contain a query type. These correspond to the categories of assertion statements which are possible. So a request might explicitly contain an authentication request[41]. The request would contain an Assertion

---

[38] Security Assertion Markup Language (SAML) 2.0 Technical Overview, Hughes and Maler, 2004, 14

[39] XHTML stands for Extensible Hypertext Markup language, it is simply a dialect of HTML which complies with XML rules regarding well-formed mark-up. This makes its format predictable and susceptible to validation by schema, and therefore parsing. This is important since form elements contain the relevant SAML XML which need to be parsed out during the exchange. See "What is XHTML?", http://www.w3.org/TR/xhtml1/#xhtml

[40] Draft SSTC protocol discussion, Mclaren and Mishra, http://www.oasis-open.org/committees/security/docs/draft-sstc-protocol-discussion-00.doc , 4

[41] Example SAML 2.0 request and response http://rnd.feide.no/content/example-saml-20-request-and-response

Consumer Service URL which defines the endpoint at which the SAML response is to be returned, and a destination attribute which can issue valid assertions in respect of the request.

The user's browser then POSTS the form data to the identity provider (in earlier SAML specifications this was referred to as an inter site transfer service (ITS)) defined by the destination, and if the user has no security context at the destination, he / she authenticates (by normal password authentication, but conceivably in another manner e.g. by certificate). This authentication creates a security context at the identity provider.

The user is then sent another XHTML form with a SAMLResponse hidden form field. This has a destination attribute which corresponds to the consumer service URL contained in the original request. The browser posts this form data to the Assertion Consumer Service URL defined in the request and the user is thereafter provisioned a security context at the service provider. This would typically be persisted by a cookie.

Finally the user is redirected to the protected resource. Subsequent requests to this service provider would detect that the user is authenticated. Should the user navigate to another service provider in the same realm (trust domain), the browser would execute the same process, and detecting the security context on the server return a authenticated SAMLResponse to the new service provider, without the requirement for the user to authenticate again. Thus the Single sign-on use case is achieved.

Note: this SAML specifications also defines a 'redirect' binding which relies on HTTP GET actions instead of HTTP POSTS (in this case the SAML request and response would be passed to the browser not in a hidden form field but in a query string SAMLRequest parameter)

The process is illustrated in the following diagram.

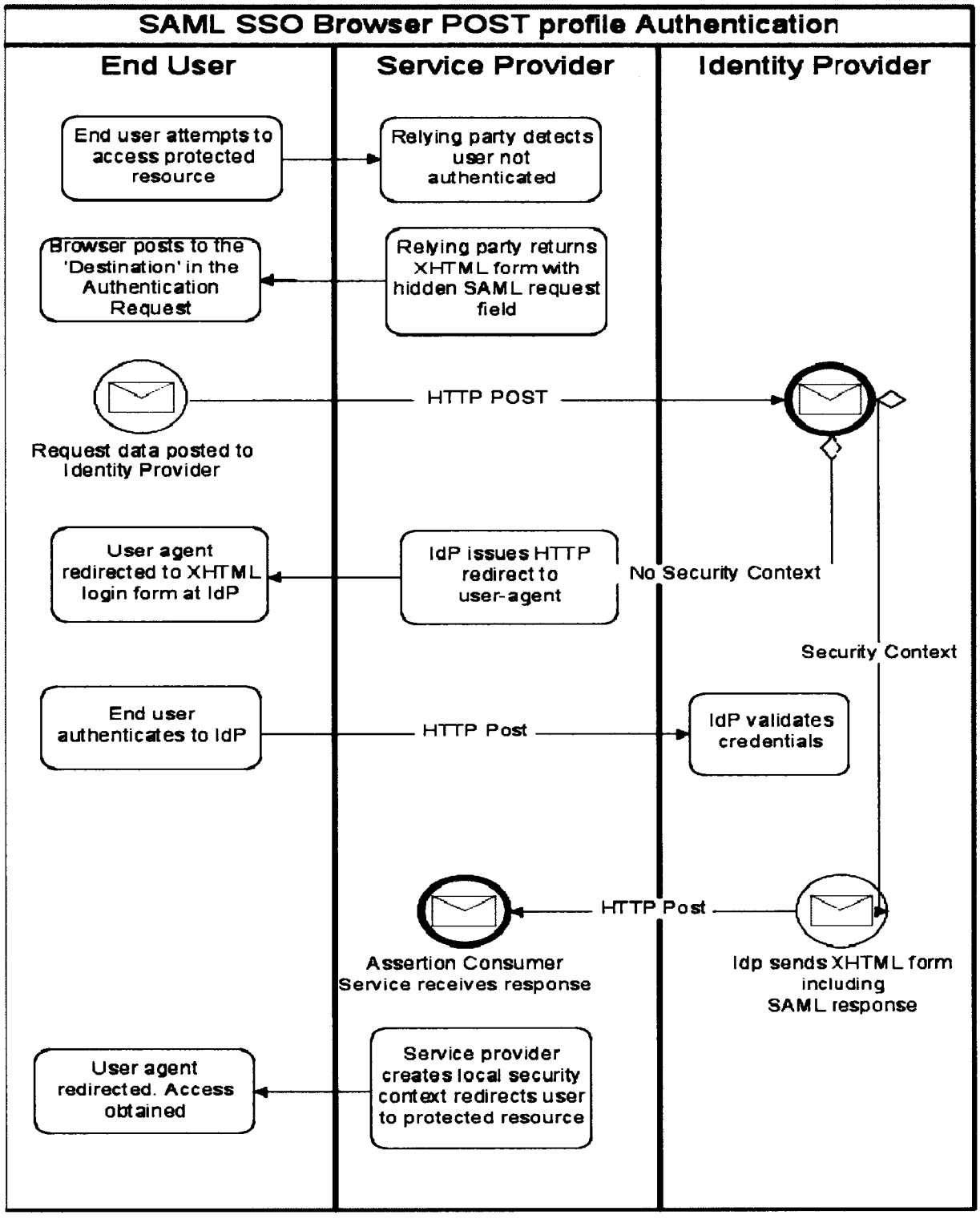

Figure 4: SAML Sign-on with POST binding

2.1.3.2.4 Back Channel SAML

Back channel exchanges, on the other hand rely on direct communication between the relying party and the Identity provider, using the published SOAP over HTTP Binding. This enables a relying party to issue an authorization request to the identity provider via the back channel (i.e. not via the user interface) and retrieve the soap response including a status code to authenticate a user

2.1.3.2.5 Additional use cases

The primary use case supported by SAML's authentication request protocol is single sign-on, but its richness supports multiple other scenarios too. Authorization queries (a particular type of SAML request) defined in the assertion and request protocol support answering queries such as "Should action(s) Y on resource Z be allowed for subject S given evidence E?" [42]. Additionally, other protocols support other use cases. A Single sign-out protocol, for instance, defines the exchanges required to nullify a security context at a given provider.

2.1.3.2.6 Strengths

Based on XML, SAML is highly expressive of multiple different scenarios. XML allows the use of multiple schemas in combination, allowing the continuous extension of these schemas and the establishment of a rich inheritance hierarchy. The schema language (XSD) supports the concept of extension explicitly which allows the model to expand, and to build upon previous versions. As an example the schema definition of the 'AuthnRequestType' has at is base attribute the 'RequestAbstractType'. This brings a level of object orientation to the model, allows for polymorphic request handling, and defines a base for future query types.

SAML is less permissive about implementations. There is a rich set of best practices, and as should be clear the protocols are considerably more rigorous as to what is permissible and how various data values, binding combinations are to be interpreted. This makes

[42] Draft SSTC protocol discussion, Mclaren and Mishra, http://www.oasis-open.org/committees/security/docs/draft-sstc-protocol-discussion-00.doc , 9

implementation and integration across organization boundaries more predictable. This, it is suggested, is what has lead to what appears to be increased adoption across the enterprise[43]

2.1.3.2.7 Weaknesses

The SAML specification is silent on how an end-user should authenticate to an identity provider, it defines the structure of the various protocols (authentication, logout) various bindings (SOAP, HTTP), and the appropriate combinations thereof for the various use case profiles.

The result of this is that a SAML implementation may support username / password as its authentication mechanism. In this case the predictable phishing vulnerability exists.

Another issue is related to the SAML supported redirect, and artifact bindings (which support passing data handles allowing parties to an exchange to interact and exchange data asynchronously to an authentication request). These bindings, it has been argued, allow an identity provider to establish information about the user (the service provider used, for instance) which it has no business knowing. This is a privacy concern.[44]

2.1.3.3 The Identity Meta-system

Arguably the most ambitious identity initiative is the so-called digital identity meta-system which is the subject of ongoing work by various industry leaders who collectively have released many specifications and schemas. This technology is the subject of open patent promises by the various involved parties including Microsoft[45], and IBM[46]. The primary idea of the meta-system is to enable disparate systems on different platforms to speak the same digital identity protocols, irrespective of their underlying authentication architecture. This is a

---

[43] Google, NTT and the US GSA Deploy SAML 2.0 for Digital Identity Management http://projectliberty.org/liberty/news_events/press_releases/google_ntt_and_the_us_gsa_deploy_saml_2_0_for_digital_identity_management

[44] http://www.identityblog.com/?p=815

[45] http://www.microsoft.com/interop/osp/

[46] http://www-03.ibm.com/linux/opensource/ispinfo.shtml

tall order, and the authors of the document outlining its aspirations[47] describe the 'practical considerations' and design goals of attempting this exercise:

2.1.3.3.1 Goals

1. Improved security and privacy. Clearly the system should lead to improved internet security and consumer privacy.

2. The meta-system would need to be inclusive of existing technologies, rather than a substitute for them. In this context the authors explicitly mention digital certificates and SAML. This is the meaning of the word "meta-system" in this context; it is a system of systems.

3. Inclusive of scenarios. To obtain broad acceptance the meta-system would need to support a broad range of sometimes conflicting requirements.

4. The system would need to be incrementally deployable, and not require some kind of 'flag day' upgrade.

2.1.3.3.2 The Laws of Identity

The meta-system additionally is subject to a set of formulated Identity Laws[48] which have been widely debated and restated. At this point they are not unilaterally agreed upon, but most authors seem to agree that they (or a subset of them) are reasonably sound.

As restated by the original author, the laws are as follows (quoted verbatim)[49]:

1. People using computers should be in control of giving out information about themselves, just as they are in the physical world.

---

[47] Design Rationale behind the Identity Metasystem Architecture, Kim Cameron and Mike Jones, Microsoft Research, January 2006. http://research.microsoft.com/~mbj/papers/Identity_Metasystem_Design_Rationale.pdf

[48] The Laws of Identity, Kim Cameron, http://www.identityblog.com/stories/2004/12/09/thelaws.html

[49] http://www.identityblog.com/?p=1007

2. The minimum information needed for the purpose at hand should be released, and only to those who need it. Details should be retained no longer than necessary.

3. It should NOT be possible to automatically link up everything we do in all aspects of how we use the Internet. A single identifier that stitches everything up would have many unintended consequences.

4. We need choice in terms of who provides our identity information in different contexts.

5. The system must be built so we can understand how it works, make rational decisions and protect ourselves.

6. Devices through which we employ identity should offer people the same kinds of identity controls - just as car makers offer similar controls so we can all drive safely.

Admirable as these may be, some have expressed criticism of the real life viability of some of them[50]. A more concise set has been described by Ben Laurie[51], which requires that for an overarching meta-system to be useful, and protective of privacy it should support assertions that are verifiable, for instance, simply asserting that one is 18 years old, is not adequate proof to allow a user to order alcohol via the internet. The assertions should be minimal, which means that disclosure should be limited to the necessary information. This corresponds to point 2 of the original set of requirements. To continue the proof of age example, a minimal assertion would simply be one which asserts that the user is in fact over 18, it should not require the user to disclose his / her actual age, or Id number, for instance. Finally the assertion should be unlinkable by relying parties, or identity providers to other assertions. This would defeat the minimal disclosure requirement.

---

[50] Revisiting the Laws of Identity, Clayton Donley
http://blogs.oracle.com/clayton/2008/08/revisiting_the_laws_of_identit.html

[51] http://www.links.org/?p=15

2.1.3.3.3 Roles

The design rationale document defines three roles:

1. Identity Providers: the entity which issue digital identities, in the same way that a bank issues a credit card

2. Relying parties (analogous to the service provider concept defined in the SAML vocabulary). The ubiquitous example is a web site, or service.

3. Digital Subjects: the entities to whom identities are issued, such as individuals or organizations.

2.1.3.3.4 Claims

In the meta-system model an identity is comprised of a set of claims (a claim-set) about various identity related properties of the subject. These properties are defined in a schema[52] and range from given name and surname to mobile phone number and gender. The schema is extensible to new claim types should these be identified.

Claim sets are represented by the concept of an information card, which is analogous to the kind of card one might encounter in the real world, such as a credit card, or a library membership card. A specific card might contain claims regarding my name, surname and country.

2.1.3.3.5 Information Cards

Cards may be of two types, self-issued or 'managed'. A self issued card is one where the claims are simply asserted by the presenter. This might seem strange, but conceptually it is no different from what happens when a user registers on a web site. The user makes a set of assertions about his / her identity, and the website simply chooses to believe them. This is fine for low value resource access, but high value resources would require a card to be issued by a card authority, for example, a bank, or a government organization, and such a card is referred to as a managed card. The 'authority' is referred to as an Identity Provider (IdP).

---

[52] Standard claim types defined by the information card model: http://schemas.xmlsoap.org/ws/2005/05/identity/claims.xsd

Authentication of the provider is effected by digital signing as outlined in the XML-Signature Syntax and processing specification[53]. Another way of looking at this is to say that self-asserted (or personal cards) are simply asserted by a local token server (STS), while managed cards use the remote STS defined in the card.

The identity selector interoperability profile[54] defines how cards should operate, be issued, validated and stored. Essentially they are simply digital data, encrypted and stored in a card store on a user's machine. The storage mechanism is secure against the loss of the device on which the cards are stored, or the removal of the card store from the device[55], by encrypting them with the user's key (a hash value created by the user's credentials) and a machine key. Optionally a user may elect to pin protect a card.

A user would typically create multiple cards and re-use them across sites and services for which they were appropriate.

2.1.3.3.6 Identity Selectors

The creation and management of cards is performed using the identity selector software installed on his / her device. Microsoft's identity selector implementation has been dubbed Cardspace, It runs on Windows XP, Vista and Windows 7 operating systems, and is compatible with the Internet Explorer Browser. There are, however, cross platform, and browser implementations for various popular Linux distributions, and Apple Mac operating systems which support the popular Firefox and Safari browsers. A number of them are supplied by the Digital Me project run by Novell which, is functionally equivalent to the Cardspace implementation[56]. There are also a number of other open source implementations, for example, the openinfocard project targeted at Mozilla / Firefox[57]. The source code for

[53] http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/Overview.html

[54] The Identity Selector Interoperability Profile V 1. 5 http://schemas.xmlsoap.org/ws/2005/05/identity/Identity_Selector_Interoperability_Profile_V1.5.pdf

[55] "What happens when my laptop gets stolen?", Garrett Serack, http://www.fearthecowboy.com/2007/01/what-happens-when-my-laptop-gets-stolen.html

[56] http://www.novell.com/news/press/bandit-projects-cross-platform-card-selector-gives-users-control-of-their-internet-identities/

[57] http://code.google.com/p/openinfocard/

this project is used for the prototype implementation which forms the proposal of this submission.

2.1.3.3.7 Browser Integration

Identity selectors are invoked when a user clicks an information card login icon. These icons overlay an HTML object tag with a type attribute of "application/x-informationcard". Clicking on it invokes the installed identity selector (either via ActiveX[58], or the plug-in architecture of the browser). The user is then confronted with a dialog which looks like this:

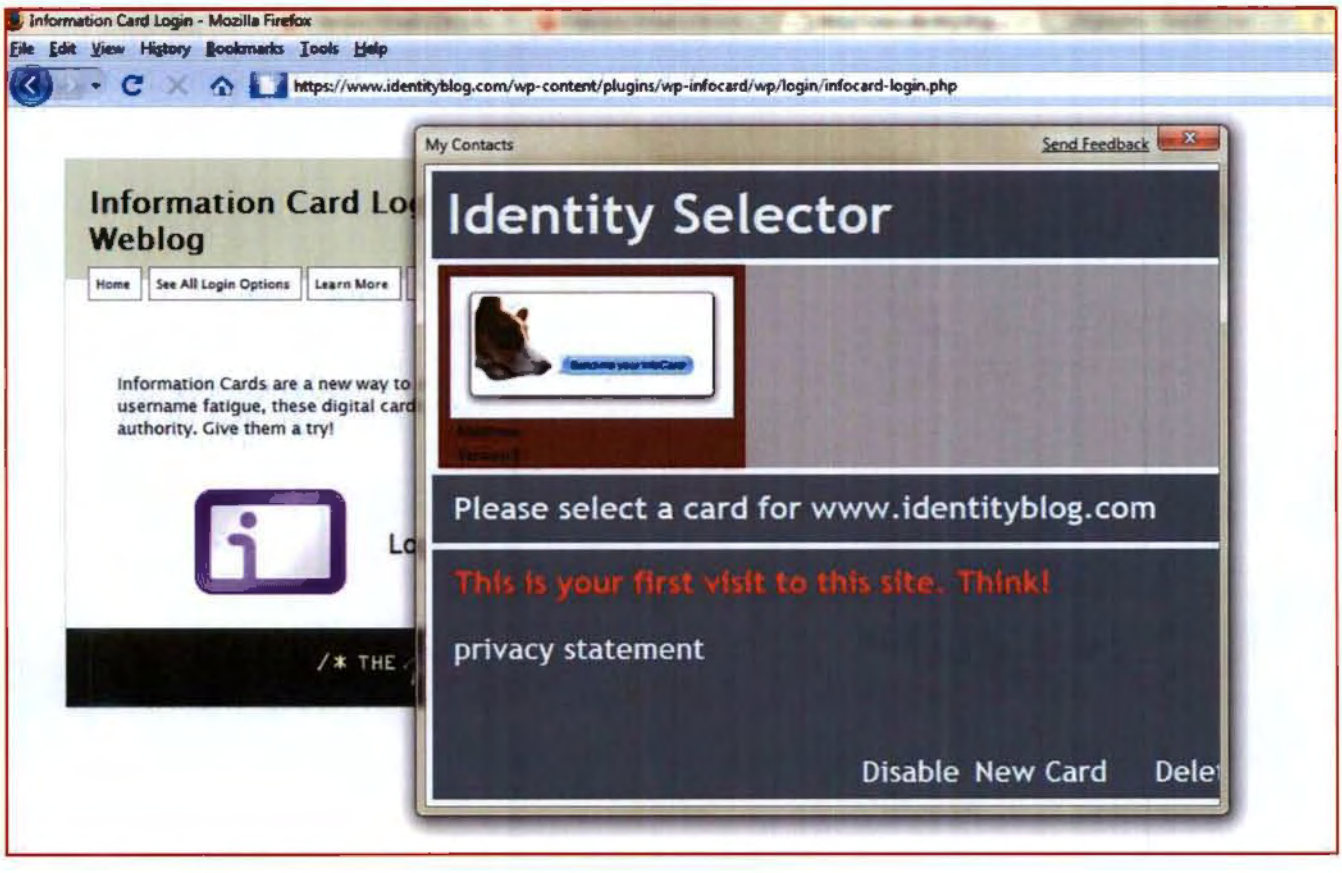

Figure 5: Firefox OpenInfoCard Identity Selector

[58] Marc Mercuri, Beginning Information Cards and Cardspace, 2008,New York, Apress, 72

This figure shows the identity selector interface displayed when clicking the information card icon at www.identityblog.com. Existing cards are displayed and may be selected, edited, deleted or submitted.

The identity selector maintains a history of sites which it has supplied claims to, and, if appropriate, advises the user that this is the first visit to the site. There is also a link to the privacy policy hosted by the site. The Cardspace implementation goes further and warns the user explicitly about any issues with the site's SSL certificate, or privacy policy.

The user has the option to add a new card, which allows them to add a self asserted card and define the claims which it exposes, or to import a managed card (as defined earlier). Additionally the selector allows a user to create an OpenID backed card, by defining the relevant OpenID URI. This is an example of the integration of an existing technology into the meta-system.

2.1.3.3.8 The standard use case

Once a card has been selected and created, it is submitted to the relying party. This triggers a series of SOAP communications, during which a token is generated by a security token service and sent to the relying party. The security token service is either part of the identity selector implementation, in the case of a self asserted card, or alternatively, hosted at a metadata exchange endpoint, in the case of a managed card. Additionally a public / private key pair is generated. This is sent along with the card, and used to validate subsequent communications.

The relying party must be capable of accepting and parsing these communications. Existing open source implementations for this requirement exist for most major web server stacks, including Java[59], LAMP[60] and .Net.

Once the token is received it is parsed out, SSL signatures are checked, and the claims are imported. A unique identifier (the so called private personal identifier) is also passed in as a

[59] https://xmldap.org/relyingparty/

[60] http://www.pamelaproject.com

claim. This in combination, with the private key sent along, is used to generate a unique, non-predicable identifier which allows the site to reliably re-identity repeat users[61].

Once the relying party has parsed the claims, inspected them, satisfied itself as to the validity of the attached cryptographic keys and signatures and (in the case of a managed card) the identity provider, it will validate the user's authentication request. This is illustrated below:

[61] Me and my PPID: Can I rely on it?, Garrett Serack, http://www.fearthecowboy.com/2007/01/me-and-my-ppid-can-i-rely-on-it.html

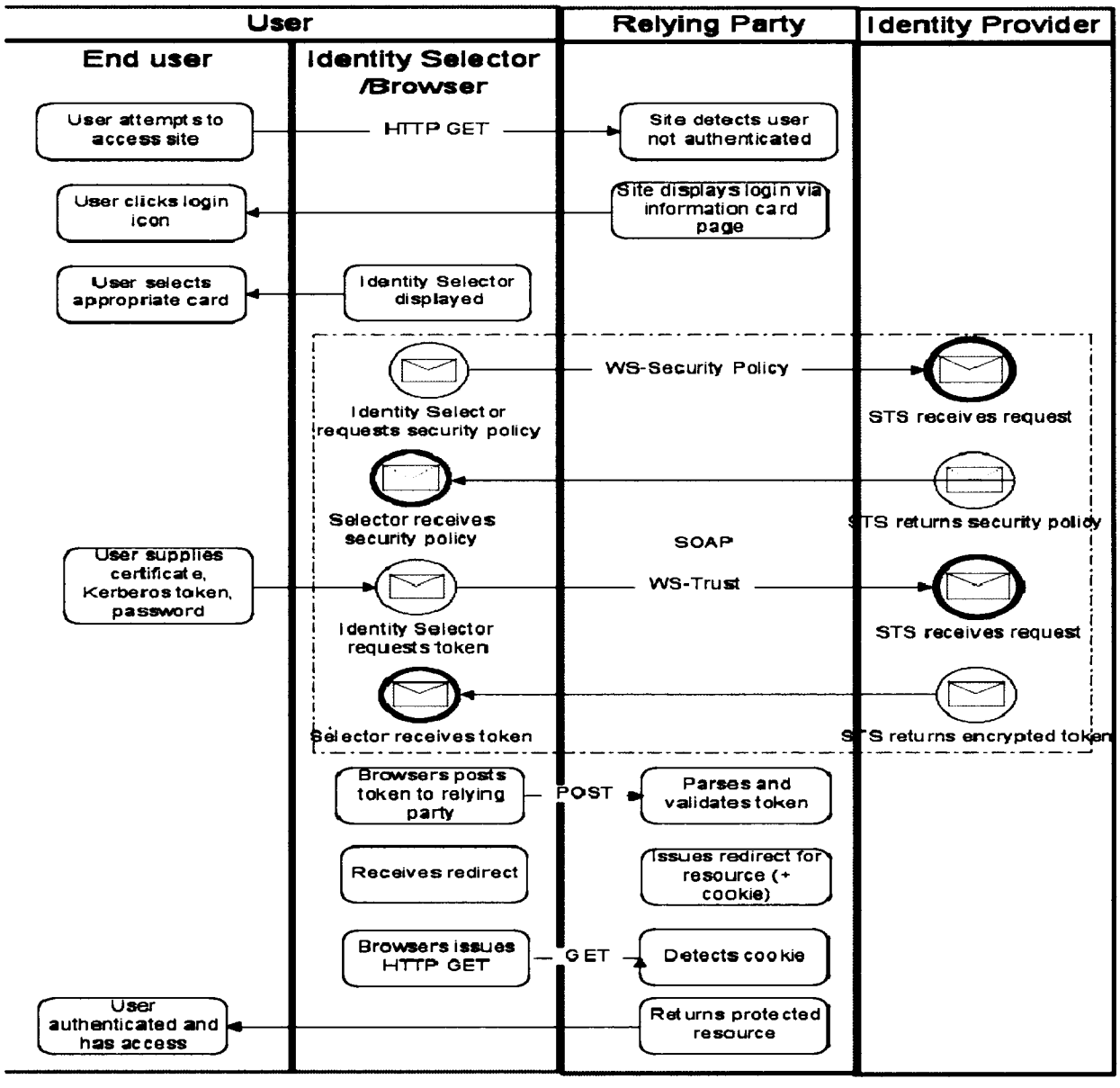

Figure 6: Information card authentication

What should be noteworthy is the fact that the user never enters a username or password; he or she simply selects an appropriate identity (card) and submits it. If a user was lured to submit the card to a phishing site, the token submitted would contain no password that could be stolen. To authenticate to a security token service for a managed card, three authentication measures are supported. X.509 certificates, Kerberos, and username / password. The username / password option may seem to open up a phishing attack vector, but it should be

remembered that the data is entered into the identity selector interface. There is no third party interface with which the user interacts at this point. This minimizes the phishing attack vector. Token replay is also mitigated by a set of factors, discussed later.

2.1.3.3.9 Relying party Policy

Secondly it is also noteworthy that the identity selector (via a security token service) might return any number of tokens. It might be a SAML token, or an OpenID one. This is defined by the policy of the relying party. A policy is expressed as part of the syntax associated with the information card object tag in the relying party's HTML. See below:

```
<OBJECT type='application/x-informationCard' name='xmlToken'>
<br/>
   <PARAM Name="tokenType"
Value="urn:oasis:names:tc:SAML:1.0:assertion">
<br/>
   <PARAM Name='requiredClaims'
value='http://schemas.xmlsoap.org/ws/2005/05/identity/claims/privateper
sonalidentifier
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress'>
<br/>
   <PARAM Name='optionalClaims'
value='http://schemas.xmlsoap.org/ws/2005/05/identity/claims/webpage'>
<br/>
   <PARAM Name='privacyUrl' value='http://www.identityblog.com/wp-
content/plugins/wp-infocard/site-messages/privacy'> <PARAM
Name='privacyVersion' value='1'>
<br/>
</OBJECT>
```

Inspecting this mark-up illustrates the site's identity policy. The site requires a SAML token which supports four claims (private personal identifier, given name, surname, email address). Optionally it requests a web page address. Finally it includes a URL where its privacy policy is hosted.

2.1.3.3.10 Strengths

The identity meta-system has several compelling arguments, the absence of passwords, the consistent modal user interface, the correlation of the card metaphor with real life scenarios, and the use of strong cryptography. The fact that it leverages existing technologies (X. 509

certificates, Kerberos, SAML, OpenID all have a role to play in appropriate scenarios), means that existing security implementations can be adapted to use it as enterprise acceptance grows.

Additionally the application of the principle of "Minimal Disclosure" is highly protective of privacy. An identity provider never need have knowledge of the relying party. It simply responds to a request from the identity selector to issue a token response in respect of certain claims. There is no linkability problem.

The metasystem implicitly recognises that users have multiple identities, and identity requirements. Thus a user may have an OpenId identity which s/he uses for blogging, and commenting on websites. The user may have another one for submitting a tax return online. In each case the requirements differ, and the data the user wishes to transmit would not be the same. Thus the user would have a card for each identity. This card would be re-used across appropriates services, saving the time and effort currently required for multiple registration processes.

It is suggested that these factors and the willingness of the industry contributors to the technology to release all the required specifications in an open format, mean that there is great promise for the identity meta-system.

2.1.3.3.11 Weaknesses

A recent study at the University of Ruhr[62], illustrated a token replay attack vector. This relied for its effectiveness on three things, the wilful installation and trust acceptance by the user of a fake root certificate, and the 'poisoning' of the users DNS. A number of individuals have pointed out the difficulty of the first requirement[63], amongst other things. Nonetheless, it is clear that there will be continued malicious investigations of this angle.

Additionally there is some research which is critical of the fact that claims data is revealed to the relying party [64].The authors' argument is that the assertion of claims data can be

[62] On the Insecurity of Microsoft's Identity Metasystem CardSpace, http://demo.nds.rub.de/cardspace/

[63] http://idunno.org/archive/2008/05/29/quoton-the-insecurity-of-microsofts-identity-metasystem-cardspacequot.aspx

[64] Improving the Security of CardSpace, Waleed A. Alrodhan and Chris J.Mitchell, EURASIP Journal on Information Security

mathematically asserted and proved via an agreed protocol) and as such 'accepted' by the relying party without knowing the value. For instance one might assert citizenship of the European Union without having to reveal to the relying party the member state in which one lives. This is an interesting possibility, but it does not allow for those cases where the claims data itself is of importance to the relying party.

This brings to a conclusion the section on the current state of internet identity protocols and initiatives. The focus has been on OpenId, SAML, and the Internet identity meta-system, which embraces both of these (being described "token-agnostic"). The proposal of this work is to extend the claims model which underpins the meta-system to include geo-data, as defined and briefly elaborated upon in the introductory chapter. Accordingly the attention of this paper now turns to the technologies, and specifications which have begun to emerge in the area of geo-location.

3.2. Geo-Location

Geo Location stands for geographic location. It refers to the process of establishing the real world geographic location of people or objects. New technology advances have meant that many users are able to geo-locate themselves, and to submit this data relatively easily. As a result there has suddenly emerged a category of 'location-aware' applications[65]. They range from social network applications[66], to e-commerce sites which are aware of your location and suggest appropriate products and services, to broadcasting services[67].

2.2.1 Enabling technologies

2.2.1.1 GPS

GPS stands for global positioning system. It is a radio-navigation technology enabled by a network of satellites maintained and developed by the US Air Force. Originally intended for military use, in the 1980's it was made available for civilian use. GPS enabled devices have

[65] "Here come the geo-smart apps', http://www.readwriteweb.com/archives/yahoo_geolocation_api.php

[66] http://brightkite.com/

[67] Social Radio Listening BBC http://www.radiopop.co.uk/

recently become widely available. It is free and continuously available.[68] It works by triangulating a set of signals from various satellites to define the user's latitude, and longitude.

2.2.1.2 IP address services

Many services[69] have emerged which allow services to resolve a user's location by reference to the IP address passed in the headers of a HTTP request. These IP addresses are resolved by checking the IP address against a database of known IP addresses or address ranges. These services are typically able to resolve a user's location to the metro (city) level.

2.2.1.3 Hybrid Positioning Systems

So called hybrid positioning systems are available as browser plug-ins[70], or JavaScript API's[71] which work by using the wireless capabilities of a device to identity local Wi-Fi signals, as well as scanning for GPS satellites and cellular towers. This data is integrated to resolve the user's location to within a high degree of accuracy. Google provides such a service to users of its Gears API, Microsoft provides similar functionality through its Live Search Service[72]. Mozilla have recently introduced the Geode[73] browser extension for Firefox, which wraps Skyhook's LOKI functionality, and plans to support the W3C geolocation specification[74].

It is worth noting that both IP and HPS lookup based systems are only as good as the database which underlies them. In the South African context the geo-location data returned is often imprecise, but it is suggested that it is only a question of time until these databases become relatively exhaustive, and the lookups they provide highly reliable.

---

[68] The Global Positioning System http://www.gps.gov/systems/gps/index.html

[69] http://www.hostip.info/

[70] http://www.skyhookwireless.com

[71] Geolocation API, http://code.google.com/apis/gears/api_geolocation.html

[72] http://msdn.microsoft.com/en-us/library/dd251082.aspx

[73] http://labs.mozilla.com/2008/10/introducing-geode/

[74] Geolocation API Specification http://dev.w3.org/geo/api/spec-source.html

2.2.2 Privacy Concerns

All these technologies are intended to provide relying party / service provider knowledge of user geo-data. In most cases there is some recognition that there are privacy concerns, and that this is something that requires user consent:

From the Google Geolocation API: "A site's permission to use location information is separate from the permission required by other Gears APIs. Permission is granted by the user in the same way as for other Gears APIs, through a dialog."

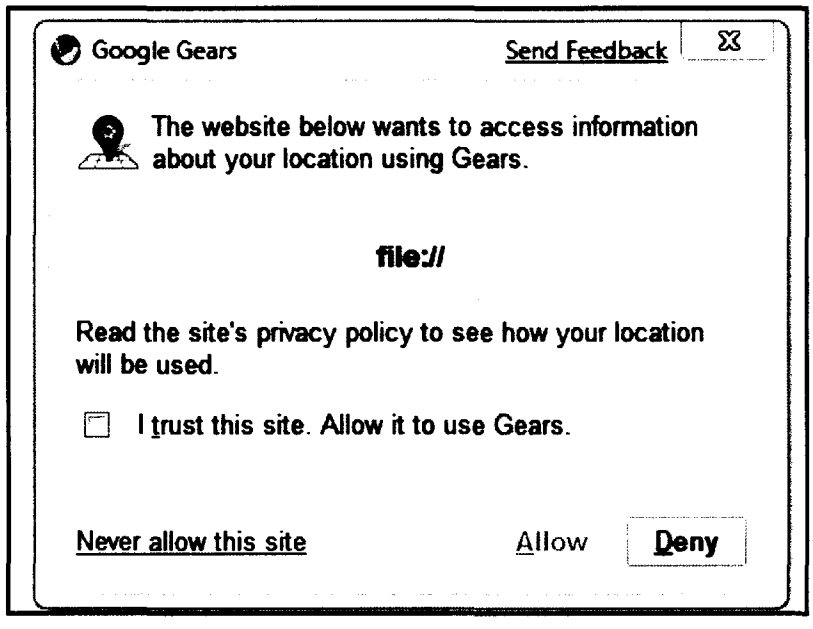

Figure 7: The Google Gears geo-data consent dialog

Mozilla have implemented their consent mechanism as follows: "With Geode when a web site requests your location a notification bar will ask how much information you want to give that site: your exact location, your neighborhood, your city, or nothing at all"

3.3. Summary

This concludes the section outlining the existing / previous work in the area which relates to the proposal of this submission. To summarize: the fragmented area of internet identity has been increasingly consolidated by a set of technologies which have as their goal the secure, reliable exchange of identity information in a way that is protective of individual privacy. The

most comprehensive and ambitious of these to date is the internet identity meta-system to which a broad spectrum of industry leaders have been contributing and which holds the promise of a highly secure, inter-operable digital identity framework. Additionally, and separately, a number of service providers and a specification have emerged supporting the concept of user geo-location. This technology is in its infancy but potentially offers much in the way of a customized, richer user experience on the internet. In the following chapter, the potential for these two technologies to be used in conjunction is examined, with reference to existing and hypothetical use cases.

4. Extending the claims model

The previous section outlined the concept of digital identity as it currently exists and introduced the concept of claims based security as envisioned in the identity meta-system. It also described some recent advances in the field of geo-location. It is argued that this analysis highlights the relative merits of information cards as an identity standard. In this context this work now focuses on a proposed extension to the claims model and some associated value propositions.

4.1. Claims currently recognized

The claims model is supported by a published schema[75] which defines the claims which are currently recognized. The list is currently restricted to 15 identity properties:

1. Given Name
2. Surname
3. Email address
4. Street address
5. locality
6. State or Province
7. Postal Code
8. Country
9. Home Phone
10. Other phone
11. Mobile Phone
12. Date of birth
13. Gender
14. Private personal identifier
15. Web page

[75] http://schemas.xmlsoap.org/ws/2005/05/identity/claims.xsd

Clearly, this represents a distillation of what the authors considered to be the standard identifiers required by relying parties. Broadly they can be divided into personal information claims (name, date of birth), location information claims (locality, street address) and contact information claims (phone number, mobile number).

4.2. Proposed new claims

It is the hypothesis of this work that the group of location information claims would be enhanced by extending it to encompass two additional ones: latitude and longitude.

3.2.1 Rationale

3.2.1.1 Privacy

The identity claims defined in the schema reflect those that the authors felt should be released only with consent, via the identity selector mechanism to sites and services the identity of which the user had explicitly been made aware.

It is clear that they considered location as a broad category of claim, however they omitted to include arguably the most granular location property: geo-location. It is unclear why this should be, perhaps it is a result of the fact that geo-location has only recently become easily accessible to internet users via the services and technologies discussed in the previous chapter.

Nonetheless, it is suggested that this is unique, personal information which should not be shared with a relying party without consent. The potential for this information to be misused seems reasonably clear.

The identity selector mechanism would make it trivial to create a geo-location information card, containing only the two claims, latitude and longitude, encapsulated in a complex schema type ("co-ordinates"), and to allow sites which require it to request that you 'log-in' with this information. (Ideally with this information only)

3.2.1.2 Existing support

Extending the claims schema with the proposed values would make it simple to express a site's policy requirement for latitude and longitude claims to be supplied to allow the user to access its geo-location services. The mark-up would look like this (relevant claims italicised):

```
<OBJECT type='application/x-informationCard' name='xmlToken'>
<br/>
   <PARAM Name="tokenType"
Value="urn:oasis:names:tc:SAML:1.0:assertion">
<br/>
   <PARAM Name='requiredClaims'
value='http://schemas.xmlsoap.org/ws/2005/05/identity/claims/privateper
sonalidentifier
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/coordinates >
<br/>
 <PARAM Name='privacyUrl' value='http://www.url.com'>
 <PARAM Name='privacyVersion' value='1'>
<br/>
</OBJECT>
```

The WS-Trust schema[76] provides a well understood open language to express SOAP requests and responses for this data between service providers and identity providers

3.2.1.3 Consistent User Experience

The identity selector interface provides a predictable, consistent and secure way to share identity data with sites requesting it. This is explicitly defined as a requirement in the draft Geolocation API specification:

"The API defined in this specification can be used to retrieve the geographic location of a hosting device. In almost all cases, this information also discloses the location of the user of the device, thereby potentially compromising the user's privacy. A conforming implementation of this specification MUST provide a mechanism that protects the user's privacy and this mechanism SHOULD ensure that no location information is made available without the user's informed consent."[77]

[76] http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3.xsd

[77] http://dev.w3.org/geo/api/spec-source.html

It is submitted that information cards present a strong candidate for the proposed 'mechanism' referred to in the specification. The alternative would be a varied set of consent mechanisms (as discussed here in the 'Privacy' heading of the subsequent Geo-location sub-chapter), these proposed measures include dialogs, information bars and would probably include standard HTML form POST mechanisms too. The potential for user confusion and poor usability is high, and mitigate in favour of a standard interface for the task.

The information card interface also defines a standardized manner for the user to access information about how the service intends to use and store this information (via the Privacy Policy link displayed by the identity selector)

3.2.2 Compatibility with the Geolocation API

This proposal is not incompatible with the Geolocation API, referred to above.

That specification essentially defines a client side script API for interrogating geo-location. It does not prescribe how that information is obtained; in fact one of the stated requirements is that the Geolocation API must be agnostic to the underlying sources of location information.

Essentially the two technologies would be complementary. Information cards would supply a robust, consistent mechanism to share geo-data with a requesting relying party, while the Geolocation API would define a standard way for the identity selector to obtain that geo-data.

3.2.3 Populating the geo-location claims

32.3.1 Self-issued cards

In the case of self issued cards, the following two mechanisms are proposed. The first is manual data entry. The 'new card' dialog would expose two additional fields, latitude, and longitude.

The following figure illustrates this, as prototyped against the openinfocard code-base:

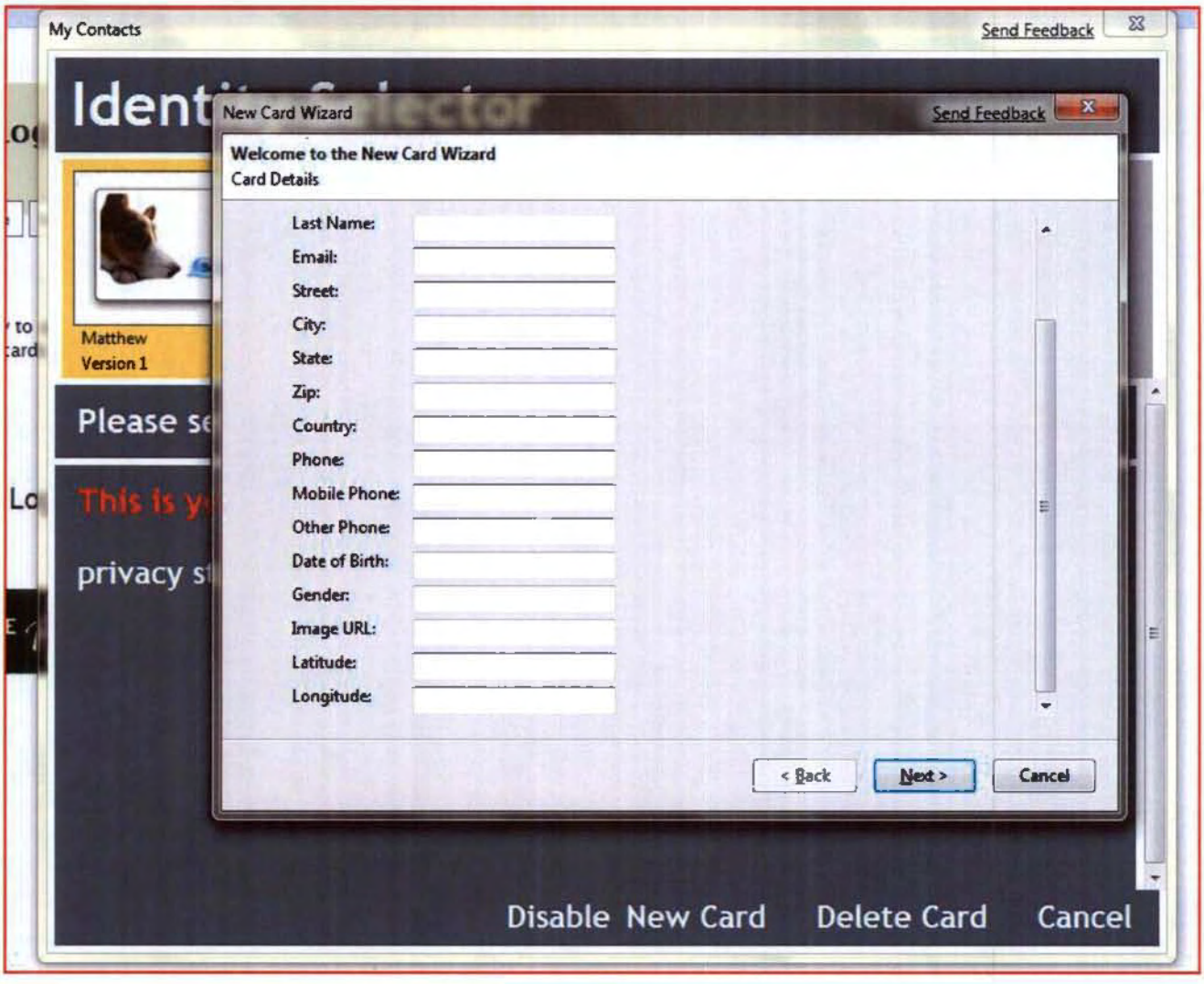

Figure 8: New information card dialog including geo-data

The second method of populating this claims data would be by invoking the geo-location API via JavaScript. This would return a JavaScript coordinates object, as defined in the Geolocation specification.

```
interface Coordinates {
  readonly attribute double latitude;
  readonly attribute double longitude;
  readonly attribute double altitude;
  readonly attribute double accuracy;
  readonly attribute double altitudeAccuracy;
  readonly attribute double heading;
  readonly attribute double speed;
```

```
};
```

Note: this requires that the user agent had some client side extension, or add-in installed which implemented the geolocation specification e.g. the geode browser extension mentioned earlier. The details of this are beyond the scope of the discussion.

The identity selector would extract the latitude and longitude data from the co-ordinates data structure and dynamically populate the data entry fields.

In either case, the form would validate for decimal values of latitude between -90 and +90, and values of longitude between -180 and +180.

The cards so created would be stored encrypted in the local card store and relying party requests for tokens including these claim types would be handled by the local security token service, which would be responsible for generating a token them.

The issue of data currency (i.e. determining whether or not the geo-location data is reasonably current) is important. A user with a mobile device or notebook is likely to use this data in multiple locations, and so it might become stale at any point. A relying party might want to express its policy that this data be no older than a certain period of time. This could be accomplished by modifying the claims schema[78] to define latitude and longitude as a complex type with the following definition:

```
<xs:complexType name="coordinates">
  <xs:sequence>
    <xs:element name="latitude" type="xs:decimal"/>
    <xs:element name="longitude" type="xs:decimal"/>
    <xs:element name="date" type="xs:dateTime"/>
  </xs:sequence>
</xs:complexType>
```

[78] http://schemas.xmlsoap.org/ws/2005/05/identity/claims.xsd

This would store this claim with an associated date time value, and return it as such in the token response to the relying party, allowing it to assess the currency of the location data presented, and decline or accept it based on their policy.

Alternatively, the information card schema includes a 'TimeIssued' element. This is a date time field[79] which could be used in a similar manner.

Another approach would be to prompt the user when selecting a card which contains these 'volatile' claims to refresh the data (via either of the methods described above), before submitting it.

3.2.3.2 Managed cards

Where a high degree of reliability in respect of the geographic data is required, a number of other interesting possibilities arise, in the case of managed cards (i.e. cards issued by an identity provider).

The identity selector interoperability profile specifies that an "Identity Provider can issue Information Cards to its users using any out-of-band mechanism that is mutually suitable."

Thus a bank might issue an information card via a USB memory device, or an email to a client user who would then import it into their identity selector software

This card could then be submitted by the user to relying parties which respected cards issued by the bank as an identity provider.

It is important to understand how this interaction works. An information card does not contain the claim data itself; it simply contains some information about what claims are supported (the SupportedClaimList element), what token services (STS)[80] exist, and what tokens it can issue; finally it defines where the STS endpoint addresses are hosted. On submitting the card the identity selector requests an encrypted, signed token containing the claim data from the STS endpoint.

---

[79] The Identity Selector Interoperability Profile V 1. 5, Nanda & Jones, August 2008, 11

[80] Marc Mercuri, Beginning Information Cards and Cardspace, 2008,New York, Apress, 249

It is also useful also to understand that card issuance is not a manual process, workflow automation processes can be implemented which issue cards on demand (obviously this requires some kind of authentication, possibly via another information card).

With these facts in mind the following scenario is possible.

1. A user requests access to a resource which requires the user's accurate geo-location data. The user has a 'hard token' device, such as RSA's SecurID product[81] (or alternatively a mobile phone, with appropriate token software installed)) issued by a location identity provider, which additionally supports GPS geo-location. Such a product does not currently exist.

2. The user issues an information card request via the device (which passes the geo-location data) to the identity provider. The communication layer might be SMS, or MMS.

3. The location identity provider authenticates the request, checks and stores the location data, and issues a card matching the request.

4. An information card is returned to the device, which is connected to the primary device (i.e. the user's laptop) via USB, or Bluetooth, and imported to the identity selector card store.

5. The user submits this card; the identity selector invokes the location provider's STS which issues a token, with a short "TimeExpires" element.

6. The token is passed to the relying party which parses out the geo-data.

This exchange is illustrated below:

[81] http://www.rsa.com/products/securid/datasheets/9651_SID800_DS_0908-lowres.pdf

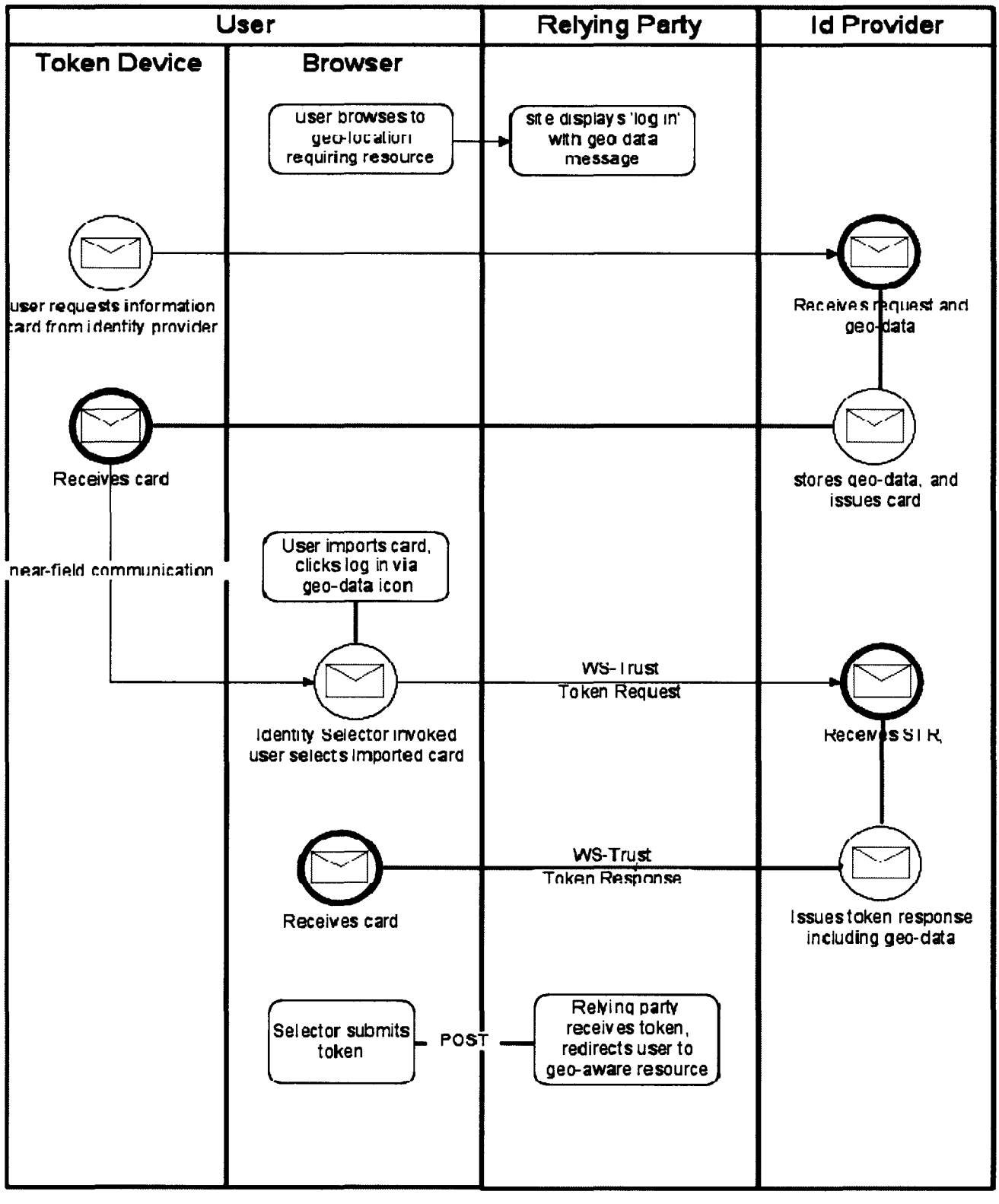

Figure 9: Location geo-data contained in managed cards

4.3. Potential applications

The use of a hardened token device (to ensure accuracy of geo-data) is a non-standard use case, requiring additional hardware, but has a number of potential enterprise applications.

3.3.1 Preventing, or restricting remote access

An example might be an application which needs to confirm a user's location (probably in addition to other authentication factors) before granting him access to resources. For example an application might need to ensure that only users at a corporate headquarters environment (as defined by a certain latitude and longitude, within an acceptable degree of accuracy) are entitled to access a resource (to prevent remote access, for example) To do so they might require an identity information card which included geo-location data. This use case would enable that scenario.

Alternatively, remote access might be restricted to certain pre-defined locations. Banking institutions or other organizations offering online access to high-value resources might issue clients who require it, with a token device which enables them to submit secure geo-data. Access to the resource would then be permitted only if the client's location fell within with a restricted set of acceptable locations.

3.3.2 Location attendance auditing

Another example usage might be location attendance auditing. To ensure that a employee actually attended a required set of locations, it would be possible to issue them with the token device and require them to authenticate their location appropriately, via an issued information card. This data could be logged and monitored. This might have applications for sales and customer service roles. It could also be useful in cases with employees in roles requiring them to attend remote locations (for example, in order to perform equipment inspection and servicing).

This type of exchange would create a requirement for trusted location identity providers who would fill the role of issuing the devices, cards and tokens corresponding to these tokens.

3.3.3 E-commerce

E-commerce retail pricing may incorporate specific rules about which prices are available in which regions. An offer available to customers in a specific geography might require them to submit an acceptable geo-data card to establish their eligibility for this offer.

3.3.4 Additional Concerns

3.3.4.1 Roaming

Although there are no current mobile operating system implementations of information card identity selectors, these are currently being prototyped[82], and will likely emerge soon. When this occurs, these scenarios scenario will be entirely possible in a roaming / wireless context. The 'card' data would be stored on the SIM card, with PIN code protection. Alternatively, and until such time as these technologies cascade to phones and mobile devices, the phone could simply be used as storage device for the cards, and the user would access the remote services via a public ('internet café') computer, browser and identity selector.

3.3.4.2 Vulnerabilities

The attack mentioned in the 'Weaknesses' section of the information cards section in Chapter 2 (i.e. the familiar token replay attack), is conceivable. As discussed there, however this attack relies for its effectiveness on two unlikely exploits (one of which requires significant user participation). That fact aside, a successful attack might allow a user aiming to 'fake' their geo-data, to do so. The implications of this depend very much on the value of the claim to the relying party. The vulnerability can be mitigated by allowing a short expiration time on the token (i.e. enforcing a very recent "Time Issued" stamp).

The user of additional hardware (as posited in the discussion above) where a secure GPS token device is required could also pose an attack vector. In this case possession of the device,

[82] http://research.microsoft.com/en-us/um/people/mbj/misc/mobile%20information%20cards%20value%20proposition.doc.

and an appropriate identity selector enabled device could allow a user to masquerade as the user to whom the device was issued, and to submit false location data.

Additionally revealing location data is an inherently risky activity. If a user were to be lured into revealing his / her location data to a relying party in error, this would constitute a serious breach of privacy. The scope for users to misjudge relying party trustworthiness in the information card scenario has been documented in an existing study[83]. This threat may be somewhat mitigated by the use of high assurance / extended validation SSL certificates.

3.3.4.3 Privacy concerns

The first law of identity is user control and consent. This can be roughly expressed as "People using computers should be in control of giving out information". Geo-data is by its nature sensitive in much the same way as an address is. However an address may be less precise than current geo-data. As a result there are legitimate concerns around privacy and the sharing of geo-data. In all current implications of the geo-location API consent mechanisms are built. However, sharing geo-data via the mechanism of information cards has several distinct advantages. Data shared via a simple HTTP post is not encrypted; data sent via an information card, however, leverages SSL to encrypt the data transfer, and is thus secure against interception during that process. Secondly the identity selector interface checks the URL, SSL certificate and Privacy Policy of a requesting site. This information is presented to the user with appropriate warnings if necessary, to allow the user to make an informed decision about sharing the geo-data. It is submitted that these facts make this solution more protective of privacy than others. Also, it should be noted that should a user elect to submit a card, the mere fact of doing so does not give a relying party ongoing access to a user's geo-data. It merely supplies a snap-shot of that information.

[83] Improving the Security of CardSpace, Waleed A. Alrodhan and Chris J.Mitchell, Journal on Information Security 2009

5. Conclusion

5.1. Findings

The hypothesis of this work centres on the viability of certain proposed additions to the existing published claims schema[84] , in order to incorporate so called geo-data. By reference to the published schemas, integration mechanisms, and a series of hypothetical applications, it is argued that the following points were established.

Through a limited set of extension to the claims identity model, it is possible to support the digital identity aspects of geo-location within the identity meta-system.

The proposed changes are of minimal, non-breaking impact to the existing schema, and are easily incorporated into the identity selector user interface.

The population of location data in an information card is possible via multiple mechanisms: manual data entry, by integration to browser geo-location programming interfaces, or by reference to third party geo-location providers.

Finally, that the emergence of a coherent identity technology which incorporates the capability to reliably physically locate an enabled end-user, exposes a number of potentially useful enterprise and commercial applications which warrant further research.

5.2. Recommendations

In order to realize these use cases it would be necessary to implement certain changes to the current information card implementation.

In specification terms, these changes are limited to the addition to that specification of a complex extensible schema definition type, 'coordinates'. This element would simply encapsulate two decimal properties (latitude, and longitude), and optionally, a date/time element, defining the time at which the co-ordinates were valid.

[84] http://schemas.xmlsoap.org/ws/2005/05/identity/claims.xsd

The inclusion of this data in the schema would impact Id selector software, relying party implementations and token services, minimally, or not at all. The identity selector interoperability profile specifies as follows:" An identity provider. "SHOULD support these claim types at a minimum. Other Identity Providers MAY also support these claim types when appropriate. The URIs defined here MAY be used by a Relying Party to specify required claims in its policy." As a result, no breaking changes to existing implementations would be entailed by its inclusion; however, future iterations of relevant software could include this functionality as a useful extension.

It is worth noting that the Geo-Location API defines in its interface additional properties relevant to geo-location. These are altitude, heading (direction of travel) and speed; in addition it defines two accuracy properties. Likewise, the 'Coords' class defined in the Google Geolocation API, supports an altitude property[85]. Taken together, these form a more accurate representation of a user's location in time and space. It is suggested that as accurate and easily interrogated API's for this data become accessible that the complex co-ordinates type be extended to encapsulate them as well, for purposes of this work, it has been assumed that useful applications can be demonstrated which rely only on latitude and longitude.

5.3. Future Research

Future research may centre on the usability of functional roaming implementations of the proposed extensions. At the time of writing there are no identity selector implementations for popular mobile operating systems (including Symbian, Windows Mobile or OS X for iPhone). These are understood to be in prototype for Windows Mobile, and the Higgins project's Java based selector will likely be ported to Linux and Mac based mobile phones in the near future.

The absence of existing software has had the result that some aspects of the hypothesis of this work (the effect of roaming and the requirement for geo-data refresh) cannot currently be prototyped. It is suggested that assuming consistent implementation of the Id selector interface, and the underlying specifications, these proposals are entirely viable; nonetheless it will be useful to assess the usability of specific aspects of the envisaged scenarios.

[85] http://code.google.com/apis/gears/api_geolocation.html#coords

5.4. Conclusion

This work has reviewed a number of new identity technologies which have emerged in response to the uniquely anonymous world of the internet. Current identity technologies have proved limited and vulnerable to a series of exploits which seem to only grow in scope.

Of these new technologies, it is herein argued that one of the most promising is that of information cards, an operating system, and browser integrated interface to a set of identity 'claims' (compositely forming 'cards') supported by a set of public specifications and current web service standards. Information cards are uniquely designed to leverage the set of existing technologies, like OpenId and SAML and X.509 certificates, require no passwords, and leverage strong cryptography to secure transferred information. They are implemented from the ground up to comply with a set of 'laws' designed to robustly protect end user data privacy, reduce redundancy and improve application security.

It has been hypothesized in this work that this promising technology would be meaningfully improved by supporting a set of limited extensions to the published specifications, which, if implemented, would position information cards as the primary interface for managing third party requests for an end user's geo-data.

How this data is provided is not fundamental to the concept, but it is argued that the W3C's emerging geo-location API would provide a standardized manner for a browser to supply geo-location data. The hypothesized scenario envisages that a request for this information would require an invocation of an identity selector (using JavaScript or the HTML 'object' tag), the selection (and population) of an appropriate card (with some associated validation) and its subsequent submission to the relying party.

The arguments in favour of doing so have been outlined as the consistent user experience offered by the identity selector interface, the robust privacy and consent mechanisms built into it, and the simplicity and standardization with which these could be integrated into a relying party's expression of the information it requires, and the way in which this information

is transmitted and parsed. The current proposed mechanisms for user consent are varied - ranging from JavaScript alerts, to browser information bars. It is desirable that a standard cross browser / cross platform interface for sharing personal information is implemented, and it is argued that information cards are a good candidate for this.

From a security perspective this novel concept of managing the consent requirement implicit in sharing geo-data using information cards is very attractive. The specifications leverage encryption technology to protect data in transfer and storage.

Potential mechanisms for populating this data are discussed and, in the context of managed (or provided) cards the scope for trusted third party geo-locating identity providers to supply this data, and authenticate an end user's location 'claims' are outlined.

Finally a series of potential enterprise applications of this functionality are discussed. Primarily these focus on the ability to securely and accurately authenticate a users location via the mechanism of a submitted information card, using a third party Identity provider and, if the ability to ensure that the geo-data is correct is critical, a hardened (i.e. tamper proof) token device to issue token requests.

6. Bibliography

1. "Google Geolocation API", Google, <http://code.google.com/apis/gears/api_geolocation.html>

2. "Hailstorm on the Horizon", Press Release, Microsoft Corporation, March 2001, <http://www.microsoft.com/presspass/features/2001/mar01/03-19hailstorm.mspx>

3. "Microsoft Open Specification Promise", Microsoft Corporation, <http://www.microsoft.com/interop/osp/>

4. "The OpenId Specifications", OpenID Foundation, 2007, <http://openid.net/developers/specs/>

5. Alrodhan W and Mitchell C, "Improving the Security of CardSpace", January 2009, EURASIP Journal on Information Security Volume 2009

6. Cameron K and Jones M, "Design Rationale behind the Identity Metasystem Architecture", Microsoft Research, January 2006, <http://research.microsoft.com/~mbj/papers/Identity_Metasystem_Design_Rationale.pdf>

7. Cameron K et al, "The Identity Selector Interoperability Profile V 1. 5", Microsoft Corporation, <http://schemas.xmlsoap.org/ws/2005/05/identity/Identity_Selector_Interoperability_Profile_V1.5.pdf>

8. Cameron K, "The Laws of Identity", <http://www.identityblog.com/stories/2004/12/09/thelaws.html>

9. Cameron K, "Why OpenID leads to Cardspace" Online Essay, February 2008, <http://www.identityblog.com/?p=923>

10. Cohen F, "Debunking SAML myths and misunderstandings" Online Resource, <http://www.ibm.com/developerworks/xml/library/x-samlmyth.html>

11. Donley C, "Revisiting the Laws of Identity" Blog Posting, Oracle Corporation, <http://blogs.oracle.com/clayton/2008/08/revisiting_the_laws_of_identit.html>

12. Fielding R et al, "Hypertext Transfer Protocol -- HTTP/1.1" Draft Standard, June 1999 <http://tools.ietf.org/html/rfc2616#section-5.1.1>

13. Gajek S et al, "On the Insecurity of Microsoft's Identity Metasystem CardSpace", Ruhr University Bochum, April 2008, <http://demo.nds.rub.de/cardspace/>

14. Graham R, "More Side Jacking" Online posting, 14 January 2008, <http://erratasec.blogspot.com/2008/01/more-sidejacking.html>

15. Hoover L, "OpenID Gets Explained, Maligned, and Dropped", January 2007, <http://ostatic.com/blog/openid-gets-explained-maligned-and-dropped>

16. Hughes J et al, "Security Assertion Markup Language (SAML) 2.0 Technical Overview", Oasis Group, February 2007, www.oasis-open.org/committees/download.php/22553/sstc-saml-tech-overview-2%200-draft-13.pdf>

17. Identity Claims Schema, Microsoft Corporation, <http://schemas.xmlsoap.org/ws/2005/05/identity/claims.xsd>

18. Jones M, "Gone Phishing" Online Essay, 2008 <http://self-issued.info/?p=73>

19. Jones M, "Mobile Information Cards Value proposition", Microsoft Research, <http://research.microsoft.com/en-us/um/people/mbj/misc/mobile%20information%20cards%20value%20proposition.doc>

20. Kirkpatrick M, "Here come the geo-smart apps" Online Essay, March 2008, Read Write Web, <http://www.readwriteweb.com/archives/yahoo_geolocation_api.php>

21. Laurie B, "OpenID: Phishing Heaven" Online Essay, January 2007, <http://www.links.org/?p=187

22. Mclaren and Mishra, "Draft SSTC protocol discussion", Oasis Group, <http://www.oasis-open.org/committees/security/docs/draft-sstc-protocol-discussion-00.doc

23. McMurtry C et al, "Windows Communication Foundation Unleashed", March 2007, Indiana, US, SAMS Publishing

24. Mercuri M, "Beginning Information Cards and Cardspace", 2008, New York, Apress

25. Popescu A,"Geolocation API Specification", Draft Specification Proposal, May 2009, W3C, <http://dev.w3.org/geo/api/spec-source.html>

26. Schneier B, "Two-factor authentication: Too little", Too late" Online Essay, April 2005, <http://www.schneier.com/essay-083.html>

27. Serack G, "Me and my PPID: Can I rely on it?" Online Essay, January 2007, <http://www.fearthecowboy.com/2007/01/me-and-my-ppid-can-i-rely-on-it.html>

28. Serack G, "What happens when my laptop gets stolen?" Online Essay, <http://www.fearthecowboy.com/2007/01/what-happens-when-my-laptop-gets-stolen.html>

29. Stross R, "And the password is...fundamentally insecure", New York Times, August 2008 <http://www.nytimes.com/2008/08/11/technology/11iht-digi11.1.15135411.html?_r=1>