Internet Identity Workshop 10
Book of Proceedings

IIWX INTERNET IDENTITY WORKSHOP 10

A WORKING GROUP OF IDENTITY COMMONS

www.interentidentityworkshop.com

Version 2

Book of Proceedings is compiled by
Kas Neteler and Heidi Nobantu Saul

IIW Produced by Kaliya Hamlin, Phil Windley and Doc Searls

May 17, 18 & 19, 2010
Computer History Museum

Mountain View, CA

Table of Contents

Day One - Monday May 17, 2010 Sessions

Designing a Faceted Identity System (M1C)

URL: http://iiw.idcommons.net/Designing_Faceted_ID_System

Convener: Xianhang (Hang) Zhang
Notes-taker(s): Xianhang Zhang

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

- If we're going to define the basic standards of identities, we need to first make sure that we are approaching them from a sound theoretical basis. This means understanding what theorists from other fields (sociologists, psychologists & anthropologists) have to say about identity.
- Every mainstream identity system works on the basis of each person getting a single login through which they present a single presentation of self
- The work of social theorists have already proven we contain multiple presentations of self and that the ones which are manifest depend on audience & context
- One potential solution for modeling this is this is to provide people with "faceted identities"
- Each "facet" of an identity represents a different "voice" through which the author talks to a different audience and context.
- In the context of blogging, this would mean that one person should run multiple blogs, each with it's own theme & content
- However, each person should only have one blog management console through which all the different facets are managed. This allows a person to publish the same piece of content to multiple facets.
- The author's job is to control the voice and it's the audience's job to determine which facets they are interested in following. This is contrasted with the "groups" model of access control in which the author determines how to segment their personality and *also* determines who is able to access each group.

- Brian Holdsworth of Microsoft has data to show that people have around 7 facets. This is in concurrence with Xianhang Zhang's experience of around half a dozen.
- Randy Farmer talked about his experiences designing multiple identities at Yahoo 360. His findings were that each Yahoo property tried to use multiple identities in a different and often mutually incompatible manner and that the complexity was too much for the average user to handle.
- Facets are explicitly not a privacy mechanism. They don't prevent people from accessing certain content. However, they do provide the context to allow people to interpret the content they are reading. That being said, privacy can be overlaid on top of facets in potentially interesting ways. eg: facets that are invite only, facets that are hidden, facets that require a shared knowledge question to access etc.
- Facets need to be carefully distinguished from tags since the two are often confused. Although both can use the same technical backend infrastructure to implement, tags are about content and facets are about voices. We talk about a wide range of content under a single voice but we only contain a few, discrete voices. Trying to use facets as content tags quickly leads to them becoming unusable.

Find out more on Xianhang Zhang's Blog Post http://blog.bumblebeelabs.com/faceted-identities-presentation-at-internet-identities-workshop-x/

Nascar For Sharing and Personal Service Discovery - (M1D)

URL:http://iiw.idcommons.net/Nascar_for_Sharing

Convener: Jian Shen
Notes-taker(s): Charlie Reverte SK

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

will meyer - addthis
jim fenton - cisco
jian shen - meebo
michael hanson - mozilla
jari koivisto - cisco
jared hanson -

will: xauth as an XRD cache? need to switch it to be service vs. host oriented. need this to support long tail of sharing

jian: sharing tools want to be able to use any sharing service vs. a pre-defined set, not just oauth + xmpp

will: want protocol support for dynamic service discovery so users can add any site they want. this is a different use case from xauth "is a person logged in to X" use case. "services I use" is > services I'm logged in to, > services I have accounts at. how do you track services a user is interested in using. store in webfinger? local storage is browser-specific and not public. problem is how to record for a user, "what services do they care about?"

jim: problem is that there are too many services that people want to use. assumption was that there is a small set of services. what if a user has multiple personas or identity providers? not everything is a browser. knowing the identity provider might reveal information about a user.

michael: discovery is disclosure, discloses personal information
all: don't want to use nastyporn.com as their identity provider
mike: want lots of services, late binding
will: keep url sharing simple at first and then build on top of it

jian: what services make the most sense on given sites?

charlie: let the user specify

michael: MRU is frequently right. just killed css visited hack

will: standardize something below "what is the best service algorithms"

michael: explaining web powerbox. content-type-specific service discovery.
will: services are handling more and more types of content, more compelling to solve problem of finding out services people want to use rather than content-type based negotiation

michael: spec is trying to handle discovery for physical resources, very noun centric. can be extended to web? invoke flickr instead of camera? movement to move local computing resources into url domain will: good, now your list of services is even larger. who registers the handlers?

michael: still vague, multiple options

will: want to be able to define "I use flickr for images" and have it be a global handler, persists across machines and be publicly knowable so others can discover it.

micheal: opportunity for browser to discover services and persist them to webfinger via XRDP. xprefs in the browser

jian: would love for service prefs to be in the browser, won't have to communicate with each service

michael: we need to articulate the possibilities for storing the prefs

jared: options are converging, xauth can cache your webfinger with browser support. re xauth, doesn't like that services can query it without your consent. powerbox lets page declare services it's interested in

jian: good idea to notify the user but companies are pinging each other in the background anyway. propose spec so people do it more in the open and so it's more seamless

will: practical issues: way for site to allow user to add that site as a preferred service. this can work with xrdp with some changes. what's the latest status of xrdp?

jared: thinks google will implement xrdp on their webfinger

will: charlie has xrdp demo, would love to start hooking in

michael: thinks webfinger might be the weak link for the long tail. difficult to implement

charlie: long tail sites can delegate their xrdp/webfinger management to others

jared: where is mozilla on this?

michael: firefox could do this as an addon. looking to see if this can make a cross device impl with weave that can persist to webfinger. can do a lighthouse impl as a tech demo

jared: going to come down to who implements webfinger etc.

michael: has webfinger support with contacts plugin. <demos it>

will: all of the pieces are there to put this all together. <oexchange demo>. good use case if a site can add a "save me as a preferred service" button, can use for personalization and persist to webfinger.

michael: public, protected and private services are separate use cases, webfinger is good for public case. powerbox is a private impl because it's local.

jian: trying to do openid and trust exchange. xauth providers should be audited.

will: use case of "send content to my mom". can't discover services she uses unless they're public, webfingerable

jian: likes the idea of generic contacts, describing which protocols they use. need to associate ids across various networks.

will: long tail niche sites can drive much higher engagement

michael: facebook going for universaility, question whether it can handle all use cases
michael: propose host-meta relation for source code link "source-of"
--going to follow up with more convo in xrdp, oexchange

Using DNS and ENUM for Identity Management (M1E)

URL:http://iiw.idcommons.net/Using_DNS_ENUM

Convener: Esther Makaay (esther.makaay@sidn.nl)
Notes-taker(s): Leon Kuunders (leon@trusted-id.eu)

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

These are just random notes that need to be finished.

The mentioning of ENUM in the title triggered a specific response from some attenders. They were interested in what was going on with ENUM and a summary of the developments in the last two years.

However not everyone present had knowledge about the subject, so we started off with a description of Public User ENUM.

With Public User ENUM you can register your telephone number as a domain name. E.g +31 802233445 → 5.4.4.3.3.2.2.0.8.1.3.e164.arpa.

With this domainname, you can publish a plethoria of other contact options, e.g. an e-mailaddress, skype account, SIP account, IM, and many more. Telco's are generally not enthousiastic about this, because it changes their monopoly stronghold (you could circumvene PSTN if you know someones SIP-address).

The domain name isn't registered on a first-come-first-serve basis. Only the person or company using the telephone number is allowed to register the number. The registration is periodically validated against the number and its user.

In The Netherlands, we've seen some use cases emerge that were inspired by ENUM, but drift in a different, identity-related direction. The idea was that if you put contact or reachability data into the domain zone, you could also put other kinds of information in the zone. This could be additional information about the phone number (the domain name) or information about the user of that number.

You could point to a website-URL containing invoicing information or an employee record (with restricted access).

The next step was to think about different domain names. Because you don't per-se need an ENUM-domain, you can do this with any registered domain name. You could work with employeenumber.idm.company.org and only publish the records on your internal network (many companies work with internal DNS servers). You can run your own 'registry' this way.

You can publish information through the domain name, or point to a data source containing more information, like a database, website or server. Although all information in DNS is public, the data source can have restricted access.

Leon is working on a use case to give employees from different departments (physical and organizational) access to each others work environments by working with their employee numbers in a domain name. Since all departments use MS LDAP, it's easy to put that information into the internal DNS servers. The DNS network is already deployed and in use (big over stacked servers that now hardly see any load). Each department can maintain their own information and decide what to publish.

This, as Dave Crocker pointed out repeatedly, shouldn't be called ENUM anymore. ENUM refers to a set of IETF-protocols that are described in RFC 3761 and anything that deviates from this (especially if it deviates this far) simply isn't ENUM. The definition of ENUM should be very precise and there's already lots of discussion going on about the narrow definition (eg in the E2MD IETF wg). Semantics are important!

The conversation dispersed into a broad range of topics, most of them concerning the technology involved.

- Does a telephone number resolve to a person or a place?
- Use a particular reference mechanism from your records (concepts/schema's)
- Business case based on making your IDM implementations more flexible. Also inspired by Phill Windley's “Digital Identity” fourth level of IDM: integrated IDM, IDM is on the infrastructure level.
- Is this mapping to an IP-addres? DNS is based on a string of names. Traditionally it maps a domain name to an IP address, but a lot of its current usage has to do with pointers that do not (directly) resolve to an IP-adress.
- Why not use XRI (discovery protocol)? Doesn't that solve these issues already? But everything already uses DNS. What's the current penetration of XRI? The main advantage is to use the infrastructure that is already there.
- Is the way you get a result from your DNS server rich enough to uses this actually?
- Are domains and e-mailaddress sufficient as an identifier? Most people have multiple e-mail addresses. Why not use iNames as persistent identifiers?
- XRI, XRD, Webfinger → should ENUM be integrated with these discovery protocols?
- DNS calls on the weblayer is that possible? (Javascript sandbox)
- Does this relate to E2MD discussions? → The telephone carriers are talking about adding attributes as well. (Calling party name, number not in use, attributes needed for handling calls via IP on an infrastructure level.)
- What about security? → DNSSEC!

- What about privacy? This depends on your use case, but you should be aware of the public character of DNS and the possibilities to use internal/private networks (like with private ENUM).
- Telnic works with its own references, is this a standard to follow? Again, depends on the use case. Telnic works with TXT records for labels to go with the contact information (eg work phone, mobile phone), uses extra address and naming fields and works with encrypted records for restricted information (only friends can decrypt).
- How can you make sure the identifiers will be unique? DNS will only work when unicity is guaranteed? Domain names are unique on the internet.
- Not everyone has a domain name. Situations differ across different countries in the world. If you don't 'own' your domain name (or a delegation), then you have no guarantee of the availability of the name as an identifier. Has also to do with the maturity of the internet space (eg in the early days, all websites resided under the providers domain). If there is need and usage for owning your own domain, it will happen.
- How does somebody who does not have your phone number find you? people have telephone numbers, e-mailaddresses, domain names
- Laws about portability of mobile phone numbers. There is not such a thing for e-mail.
- Phone numbers are very public, how do you control access to this? You don't (DNS is public), but it's a voluntary registration. It's different from handing out business cards of course, but the DNS is not a database-lookup system. You cannot do “select * from .com where domain like thisname”. You can only look up records with a domain name, not the other way round.
- It would be possible to shield information by using proxies.
- Validation of regular domain names could be helpful for building trust. Validate the WHOIS credentials of the registrant of a domain name. Is this the same as the extended validating from certificate providers? No, those validations apply to SSL-certificates that are used for websites. Validation of a domain name extends to all use of that domain name (eg with e-mail).

The ideas around using DNS and ENUM are very interesting, but since there's so many technical aspects involved (discovery, identifiers, reference-schemes, pointers, usage), it easily gets over-complex and confusing.

In the end it was decided that Esther will (try to) describe the subject in a tight non-technical manner. It should help to simplify the subject if we leave the technology (however interesting) for a later stage.

Getting Started-Internet Identity-Understanding Key Technologies and Issues (M1F)

URL:

Convener: Gordon Clarke – Kaliya Hamlin

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

We went over the basics of user-centric identity
* OpenID
* Information Cards
* Federation

Can the Open Pile Become Beautiful Again (M1G)

URL: http://iiw.idcommons.net/Can_the_Open_Pile_Become_Beautiful_Again

Convener: Johannes Earnst
Notes-taker(s): Doc Searls, Johannes Ernst

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

> The premise of the session was that the "Open Stack" of identity technologies (identifiers, discovery, authentication, information exchange ...) keeps growing like weeds. It becomes increasingly difficult to keep track of its (decentralized) evolution. This creates multiple problems:
>
> 1. potential adopters are intimidated by the sheer size of what they are expected to know
>
> 2. implementors are unable to implement all parts; there seems to be no example in the wild for implementations that support most of the member technologies
>
> 3. the usage model for the user is not becoming simpler, to put it mildly.
>
> 4. interoperability becomes more rather than less elusive.

There was general consensus in the room that this description of the state of the art is accurate. However, participants were unable to come up with an approach for how to "cut 80%" and make it simple, beautiful and cost-effective again.

Small Business Software on the Open Web (M1H)

URL: http://iiw.idcommons.net/Small_Business_Software_on_the_Open_Web

Convener: Sunir Shah
Notes-taker(s): Sunir Shah

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Sunir from FreshBooks (sunir splat freshbooks dot com) reintroduced the Small Business Web, an organization of over 100 companies looking to build the market for small business software over the Open Web.

We talked specifically about use cases about how to use the Open Web stack in order to engineer a bigger market for small business software.

Use Case 1. Seamless integrations

Cross-sell related services from one app to the other. e.g. FreshBooks recommends Tick for time tracking.

1. FreshBooks creates a Tick account for the user (OpenID+OAuth)
2. Tick accepts log-in info from FreshBooks (OpenID)
3. FreshBooks grants API access to Tick (OAuth)
4. Tick discovers data! (Contact info, clients, projects) (Portable Contacts, ???)
5. Maybe Tick charges FreshBooks for the account (???)

Use Case 2. Marketplaces for SaaS apps

e.g. Google Apps Marketplace

Present
Google creates a FreshBooks account (OpenID+OAuth)
Google provides single sign-on to FreshBooks (OpenID)
Google grants FreshBooks access to data (Contacts) (OAuth, Portable Contacts)

Future
 FreshBooks charges Google for account (???)
 FreshBooks registers Invoice service w/ Google (XRD + registration protocol)
 BatchBlue discovers FreshBooks from Google for Invoices (XRD, Webfinger)

We discussed aggregating identity

* email provider and equipment OEMs are natural sources
* single sign-on
* single payment source
* when you hire Sally and fire Joe and promote Sue, it's easier to add/remove software if identity is managed centrally

We then talked about the issues:

1. negotiating contracts with each ISV partner. Can we automate this?
2. do we need to audit companies with a better business bureau?
3. We need a protocol to register services with the XRD profile
4. Business identities are owned by the business, not the human
5. Password recovery is hard when there are catastrophic staff losses

OAuth 2.0 & OpenID Connect (M1I)

URL: http://iiw.idcommons.net/OAuth_2.0_WTF

Convener: David, Joseph, Eran, Allen
Notes-taker(s): Judith

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

OAuth 1 was (is) influential, but not adapted widely. Crypto implementation is hard.

The discussion illustrated the tension between ease of implementation and security. If it's too hard, we know web developers will do something MUCH less secure.

See http://hueniverse.com for some good blog entries

Finally, parts of the spec are being locked down in the next month or two: only changes that will be accepted will be those to enable interop or security.
--==X==-- Introduction --==X==--

In draft form; implementations ruby, PHP, Cocoon, Perl . Servers are twitter & facebook. (Not sure but believe this is being developed with ITF standards.)

* Now simplified. Rely on SSL, skip signing.
* Now with more flows: web, desktop device, mobile, etc. (This simplifies over the attempt to have one unified flow.)
* Now a notion of refreshing tokens, so temporary access tokens.
* Now to provide a place to list scopes. Currently providers define own scope. Questions on how to reduce scope.

--==X==-- Q&A --==X==--

Security: webserver flow - type access code in URL.
A voiced dislike of having SSL. Signatures is an option to get away from SSL. Also short lived tokens like one use.
A voiced dislike of access code in URLs. What about server logs? Pass it using headers or POST (not sure I heard it right, with further discussion of PUT & GET).

Signatures are now more simple, only one algorithm. Gives developer an option to not use bearer tokens. Yahoo notes that the security needed for signature secret can be contained and by distributing bearer secrets limit risk exposure.

1.0 Access tokens lived forever, but could be revoked. 2.0 Refresh token lives forever, only used to get access token which is short lived.

Extensions: Aaron, editor of OAuth user guide. 1) Providing: additional flows in req type option with a registry. 2) Additional parameters like display parameter. Name space registry. 3) options with signatures.
-- Registry questions. IANA mentioned.

Twitpic is a 4th party usecase: open discussion.

--==X==-- NEXT --==X==--
OpenID 2.0 connect

Synthesis of OpenID & OAuth in the past: boat car.
* OAuth required relationships
* OpenID open relationships

Twitter built ID on top of OAuth, other providers have been hacking different solutions.

OAuth2 has scope: Call this one OpenID to get ID and some profile info (another scope email?).

Response is an identifier (Webfinger account URI or https URL). Time issued. &ccess token (just for the OpenId scope?)

Scope call with access key gets a collection of five or six reseved keys including names and profile URLs. Potential to express endpoints for requesting various resources like portable contacts.

Separate profile url from identifier, thus widely personalized profile sources (like OpenID) but rigorous standards for the implementation.

Discovery flow & technology: LLRD....

Eve pushes back that some information may be useful on initial claim. Much discussion about modular design goals, making it simple.

Demo implementations ontop of existing OAuth 2 is only a few more lines of code.

Simple to do the core easy stuff, but allowing some hooks to solve different & difficult problems.

Online Voter ID – Registration: How do we do that? (M1N/O)

URL:http://iiw.idcommons.net/Online_Voter_ID_How_do_we_do_that%3F

Convener: Wayne Burke
Notes-taker(s): Heather West, Andre Boysen

Tags for the session - technology discussed/ideas considered:

Voter, egov, online service delivery, proofing, Voter registration, identity, claims , verify claims before they are issued

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

NOTES FROM Heather West:

Online voter registration hinges on authenticatable online identity for the public
BC government is working on a system where the information isn't shared (avid the SSN issue)

Netherlands have a centralized municipal database of voters (thanks, Napoleon!).

Canada and the US do not have this kind of centralized databases at the provincial/state levels

Trust in hardware is very different than trust in machines and hardware
In person voting is the one way to guarantee a private, secret vote

Make a list of the possible uses for an online voter registration: constituent services and communications, rulemakings, verified stakeholders, health care for single payer systems,

Don't want an identifier, want to be able to see whether someone is entitled to vote (lives in the district, has not yet voted in this election, etc)

Keeping government communications and verifications to attributes rather than full information is helpful. How do you decide how to verify, though?

The real world system is bound to change because the change in the way that people interact and the speed at which we can change locations

The current proofing of voters often involves a utility bill - maybe get them to print some barcode on the bill to verify location.

No good way to prove residency - utility bills!

Canadian universal health care number is much like SSN in terms of being universal

NOTES FROM: Andre Boysen

- How to identify people.
- How come I can’t vote online?
- Holland last election tried to innovate, but rolling back for technology failures.
- How to govern the process for US voting and registration, without invading privacy.
- Omni handles are powerful but invasive.
- What is the information I need to decide if this person can vote?
- How do I answer that question – who can authoritatively answer this question?
- Risks around the different steps in the process.
- Risk around voter influence
- Risk around voter identification
- Risk around voter collection of intent
- Risk around correctly tabulating voter intent
- There is certainly the potential to increase voter participation if we could remove the friction
- Verified voter capability will be great for voting, but what are the positives and negatives this being used in other contexts.
- This would be very handy to allow online communities to meet online with all participants knowing that their peers are validated in the same way.
- Knowing that there is a ‘real’ person attached to online collaboration is going to be key to prevent manipulation.
- It comes back to identity proofing.
- User control of brokering of information between AP and RPs is key. The AP can not prevent (nor should they) the user from making this disclosure
- Multiplicity or duplicated entitlement prevention needs to be considered.
- Voter registration basically an affidavit, license, or vendor bill showing address.
- There are tradeoff between how much data is in the credential and how much is derived through the use of the card.
- How to prove address without actually inviting auditors over to the house. Documents, credentials, volume + consistency of evidence.

Mozilla Proposes: Account Management in the Browser (M2A)

URL: http://iiw.idcommons.net/Mozilla_Proposes

Convener: Dan, Mike, Ragavan
Notes-taker(s): Dan Mills

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Lots of questions about how this is different from just a password manager?

* it is an extension of password management, and an abstraction of ui so that different auto mechanisms can be used.

* password managers try to guess how a site works, account manager aims to be completely deterministic in it's behavior.

* account manager includes ways for the site to advertise to the browser multiple ways of interacting (e.g. Passwords vs http aith vs openid), as well as current state (signed in, not), which would otherwise need to be scraped.

How will this work with other password management extensions?

* Apis need to be included so that extensions can make use of account manager.

How is this better than something 401 related for intranet sites?

* allows for disclosure of endpoints and status without a 401
* goal is to interop with http auth, not force it on everyone.

Are you introducing new http headers?

* yes, we are using the Link header, and defining a new one for the website to advertise to the browser the current user signed-in status.

How do you behave if you get a 401 with an unknown authentication scheme?
401 vs. 200 discussion?

Where will this code live? Will it be part of platform? Firefox? How will this affect derivative works?

* not clear 100% how much will be part of the embeddable gecko vs in firefox.
* the protocol will of course be open, and the firefox I pleme tat ion will be open source.

Why did you restrict to just username/secret?

* I'd/secret negotiation is for the username-password-form profile only; other profiles could do something different (but would need ui implemented)

Why didn't you use html markup for registration?

* still a possibility, feedback welcome. We felt like it would be more error-prone.

If I don't support registration, can I just put a URL in the AMCD?

Can the site specify a password policy? Sites seem to truncate long auto-generated passwords?

* yes, feedback welcome on whether it is sufficient.

Preventing against malicious attackers flooding the registration flow.

* need to think about this one, open problem.

How does this fit with OpenID Connect?
* hope is that the protocol pieces of opened connect can be managed using the account manager ui.

Can you to a well known location to get a 401 that also incorporates the TLS client cert piece?
* interesting idea, needs more thought.

Companies holding a lot of identities need to get off username/password to access tokens because of phishing attacks. They like OAuth, because it is based on the WWW-Authenticate header.

Account manager meet-up at the Mozilla offices in downtown Mountain
View this Friday afternoon, see the Account Manager page for details:
http://www.mozilla.com/firefox/accountmanager

Digital Heritage - What Our Info Says About Us (M2C)

URL: http://iiw.idcommons.net/Digital_Heritage

Convener: Stacey Pitsillides
Notes-taker(s): Stacey Pitsillides

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

- What happens to your data when you die?
- Is there an info-structure to support this, if not could there be one and how would we begin to go about developing one
- What does it mean from a legal dimension? Who owns your data, do you even have a right to own your data once it is online? If someone takes a picture of you, whose property is it? Yours (as it's OF you) or the person who took it?
- What do people generally want to happen to their data? Are people even thinking about these questions?
- What does it mean to disseminate this topic to the public and how does one go about it?
- Is the system we have at the moment (which ignores death) creating harmful social repercussions
- What does it mean for people or communities to engage in Digital Archaeology?
- What place will the digital information we create today have in future generation ability to reflect on this period in time?
- How is this to be curated or edited, does it need to be? Should all information be there and free for all to access? Does this include personal or sentimental information?
- How does one go about bequeathing their information to a relative?
- Is there such a thing as a digital asset?
- What does it mean to be awash in Digital information, how do we stop our getting lost in an ever expanding Digital footprint?
- How does 'perfect memory' ie digital memory change our cultural identity?

Recovering A Lost Identity - Can We Do Better Than Email (M2D)

URL: http://iiw.idcommons.net/Recovering_a_Lost_Identity

Convener: Michael Sprague
Notes-taker(s): Michael Sprague

Tags for the session - technology discussed/ideas considered:

multi-factor authentication, strong authentication, recovery, legal

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Lively discussion… a dozen or so participants… many open questions…

As the value of services on the web grows, i.e. SaaS, banking, etc., the value of the identity used to access those services grows. Also, as identity providers emerge to assert identity across multiple relying parties, the compromise of a single identity grows in importance.

The two systems used today for identity recovery are email and secret questions. Both are easily compromised. Secret questions (i.e. what high-school did you attend) were particularly ridiculed. Most answers can be easily found through a web search.

These systems, however, developed out of a need to keep costs down. Having help desk personnel engage a customer to recover a password for a free email service is cost-prohibitive.

Perhaps recovery of a lost ID could be a chargeable transaction. Some reacted to this notion as if it could be exploited as a form of blackmail. On the other hand, a user could pre-establish a method of recovery and commit to such a charge when opening an account. This is more palatable. If one forewent this option the identity could be unrecoverable.

Recovery is a back-door to authentication and thus should be commensurate with the strength of the original authentication. If my authentication is level 2 my recovery procedure should similarly be at level 2.

Within an organization the policy is usually, go talk to your admin, who likely can verify your identity and re-issue access. On the open Internet this is not an available procedure.

Of course a way to establish recovery is to link authentications. When more than one form of authentication can access an account then secondary access can be used if primary access is lost. Barring this what is the legal process? Can it be standardized? Many examples were explored. An admin with the only access to critical company data in a cloud service gets hit by a bus. The CFO will go through legal channels to gain access to this account. Perhaps this is something that can develop into standard and accepted procedures ...essentially an out-of-band counterpart to a technical authentication mechanism.

Voluntary OBLIVIOUS Compliance (M2E)

URL:

Convener: Alan Karp
Notes-taker(s): http://iiw.idcommons.net/Voluntary_Oblivious_Compliance

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

THERE IS A SLIDE-DECK (PowerPoint) FOR THIS SESSION

P2P Networks Version VEGA (M2F)

URL: http://iiw.idcommons.net/P2P_Network_Version_Vega

Convener: Markus
Notes-taker(s): dsearis

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

There are three architectures.

Centralized... FB, Twitter, passport

Decentralized, Server-based, somewhat distributed... PDX, OpenID

Distributed. Mesh-able. No servers. Computer to computer. Like bittorrent. This is VersionVega.

Look up peeople by i-names. Yes, i-names need a centralized registry, but it's possible to store these in the network itself. People have done it. I haven't done it yet. For example: CSpace.

Demo with two machines, one Windows, one Mac.

Windows machine has a background system that exposes the api and services, he can access the api and background. Needs TCP and UDP.

Two apps, customlight and Restarbot. Launches Customlight, which is the main browser app. Distrib net with five proxies. There is the concept of a node runlevel. Choice to start a new network or connect to an existing one.

Andy: It's not an XRI lookup, but a local lookup of the = name.

At the high level there is an XDI store. Strong certification. Based on RDF triples. Can store XDI statements in a distributed way.

The one weak point of the story is that i-Names are centralized.

The point of using i-names is so everybody can use a private/public key pair. Once they start communicating the session uses noting from i-names. A piece of data may end up being stored on multiple nodes by replication.

Andy: what if the network goes down, where is it?

Markus: gone. Think of the data as in the network,

Drummond: Lead access with link contracts?

Markus: no.

Storage is one of two major functions that the network provides. The other is simple messaging,. I can send messages between individual nodes.

Built into the network is the concept of groups or multitasking. Members can send messages to any or al of those in a given group.

This one is build on XULrunner. So running this is very similar to running a Firefox connection. The extension manager looks like Firefox’s.

Showed a voting mechanism.

Mary Rundle: what about Malware?

Using a library called Freepastry, which has a lot of protection and intelligence.

Open Source? Haven't thought about it.

Is It Time for a New “Liberty” from Single-Vendor Dominance Alliance (M2G)

URL: http://iiw.idcommons.net/A_New_Liberty%3F_to_prevent_single_vendor_dominance

Convener: Johannes Ernst
Notes-taker(s): Doc Searls

Notes:

Johannes: Things have actually gotten worse. There are too many acronyms. (Consider Kaliya's talk.) If I want to implement discovery on my server, I don't know what to do. Yadis started simple, and now there are N proposals, no agreement.

Stuff might or might not be in an HTML header.

Many versions of the hammer stack.

unless we have something smaller and better defined, we can't implement it.

The focus on user centricity is being lost.

"User control and consent" Remember that? OpenID Connect ignores that. It is now considered naive to be user centric. People don't want it, supposedly. Kim Cameron's Laws of Identity have fallen far behind

Identity infrastructure needs to be distributed and simple. Nothing any more centralized than DNS. XOauth?

Panzer: Xoauth does not have to be centralized. All data lives on the client. Centralized server required by Javascript.

Andy: In IIW #1, we had a common ideal. Agreed on basic premises.

John Panzer: Xoath may start with cantral server at Google, then Google will implement it in Chrome if the Chrome people will do it.

Andy: There is a cynical take here, involving third parties using client-side data.

John: want to reduce the NASCAR buttons required when checking in.

Johannes: Idea behind LID was implementing in an afternoon. Don't like the trajectory now. Have no UI because there are too many technologies involved.

Hank Mauldin: Now have VRM, PDS, Data Portability. Way too many things being proposed. Confusing. Not much on identity.

Doc: Here is some history. DIDW, Identity Gang, IIW... all personal.

Johannes: Now we have deployment, where before we had an open space. Still, openID usage is not very high.

Jaap: Play time is over. It's time for the corporates.

Johannes: True maybe for telcos. But not for the rest of the world.

John Panzer: Google has its stuff. Whenever you're figuring out what to deploy... there is a make or use choice: make it yourself or use what others have made already (open source). For using other guys' stuff, Well, OpenID has all these versions...

Hank: One good trend is OIX. Frameworks are attempting to be built, and this is a good thing. From a biz perspective, OIX is one of the better things that has come down the pike and is at least a step in the right direction. Huge step.

Johannes: Somebody has to say "we'll take this one, and let the others go... would be nice if OpenID or OIX had a narrower spec. It now has to please many factions, killing adoption.

Andy: Toolkit and use cases.

John: consider the users who put their Facebook login in the URL bar or the Google search bar. That's what we're dealing with.

Johannes: Want to start with blog comments. Or health care. Need a decentralized org or tech focusing on a single market Or people. Neither has happened.

Andy: That's what I'm doing. ooTao tried to solve problem that didn't exist. Now in industry. After 1.5 years at OCLC I can see the use cases. If you have a business where value is flowing, they'll pay for something.

Johannes: Kim Cameron liked common ceremonies. I now don't know what to evangelize any more. Maybe the best avenue now is for facebook to take over the Net in general for personalized experiences until the rest of us get it together.

We were trying to standardize, but all this wild development makes it harder, not easier. Easiest adoption is using the big evil company's SSO.

The UI should be invisible.

From the consultant's perspective, it's "Implement Facebook and get it over with."

(somebody) Can't find one open solution. Facebook looks attractive.

Hank: If all I have is one login screen, and I can't do it, where are we?

John P: We need implementations that are simple to make happen. The user experience has to not suck.

Jaap: If I only have to produce an iPhone and click on this. T-Systems.

?: You'll have to rely on some big company.

Omidyar is using simple SSO for paypal. His new take on journalism news site. Hawaii Civil Beat. $20/month.

Open ID Connect – WTF? (M2I)

URL:

Convener: David, Erin, Joseph

See: http://openidconnect.com/

Magic Signatures & Salmon (M3A)

Convener: John Panzer

See : Salmon Pixie Dust Session Notes
http://iiw.idcommons.net/Salmon_Pixie_Dust
See John Panzer's post on Magic Signatures
http://www.abstractioneer.org/2010/01/magic-signatures-for-salmon.html

E-ID Business Ecosystem (M3B)

URL:http://iiw.idcommons.net/Cet_Competing_e-ID_providers_creating_a_Market

Convener: Douwe Lycklama
Notes-taker(s): Chiel Liezenberg

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

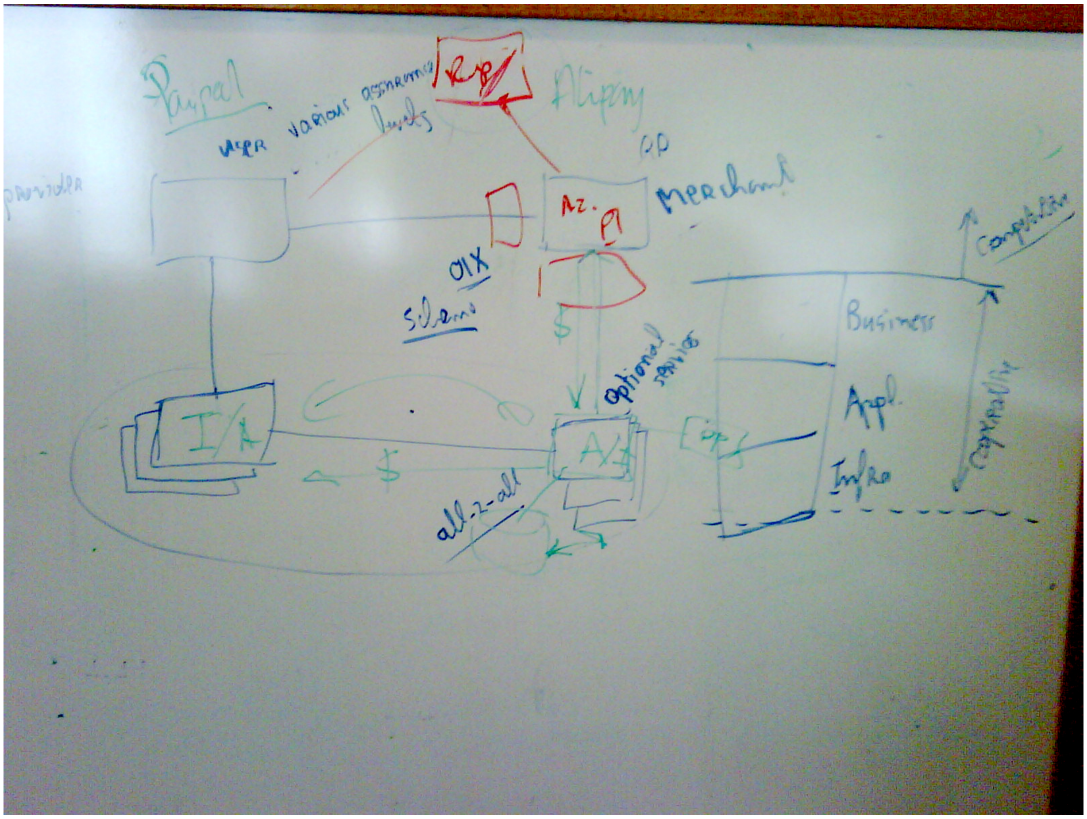
E-id business ecosystem:
Competing EID providers work together in creating an EID market

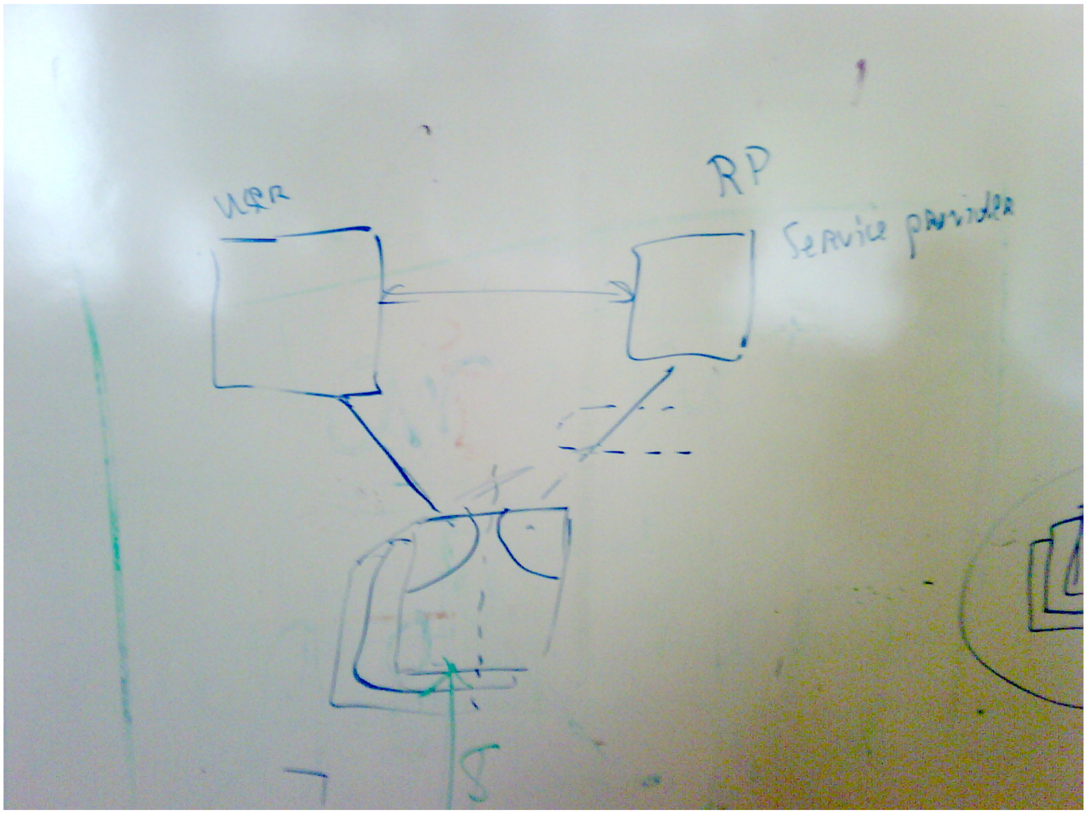
The problem - No single identity provider will ever get 100% market share, leaving users and relying parties with fragmentation in solutions. They have a reciprocal interest: users want to use their credentials in as many as possible places and relying parties want to be able to accept as many users as possible. A world where every identity provider is a 'walled garden' of end users and relying parties will not lead to massive adoption and usage. Also it is unclear what the business model behind identity will be. Everything seems 'free' and dealt with in barter arrangements.

The solution - EID providers need to cooperate, leading to interoperability. This interoperability is not only on technology but especially on functionality and business aspects. Lessons can be learned from the payments world where a scalable ecosystem has developed (e.g. Visa, Mastercard, credit transfers). Such an ecosystem is build up around a 'scheme' where all the agreements for business interoperability are made. Providers define their 'cooperative space' on top of which all can build their competitive propositions. End-users and relying parties each have their own service providers who interoperate. Service providers cooperate because they feel that by working together a bigger market will be realized.

The case - In The Netherlands recently a scheme for EID has been developed by market parties, where the government is a launching customers for identifying businesses represented by their employees. The scheme also supports various assurance levels for the EID providers. The scheme has oversight and the EID providers must comply with scheme rules.

Action - Have a look at OIX, who are addressing similar issues, but more from the trust side and less from the business model angle.


Paypal
user
various assurance levels
RP
Alipay
provider
RP
Merchant
Az.
Pl
OIX
Silemo
$
Optional service
I/A
A/A
$
all-2-all
Business
Appl.
Infra
Competition
Cooperation


user
RP
Service provider
$

Ome Social Web - aka xnpp & Social Web (M3C)

Convener: Laurent Eschenauer @eschmou

See: http://onesocialweb.org/

What do “regular” web developers need to know about identity? (M3D)

URL: http://iiw.idcommons.net/What_do_regular_web_devs_need_to_know_about_ID

Convener: Laurel Fan
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

This was a small session -- not a lot of solutions, mostly statement of the problem (and some free association...)

Most regular web devs don't care about identity. They care about what identity enables, such as:

- putting a user's action/content into buckets
- acting as the user on another service (post to facebook wall)
- getting information about a user (name, photo, friends list)
  - to personalize content
  - so the user can give you this information without filling in another form

Single sign on isn't that attractive by itself. It's easy to maintain your own username/password, email verification etc. There's a library for that. It's hard to depend on other sites, and explain this to the user (Do we need an I forgot my OpenID button?)

If there's a library, people will use it. Better if it's built into PHP, Rails, etc. (Oauth seems to have learned this)

User Managed Access - UMA (M3E)

URL: http://iiw.idcommons.net/User_Managed_Access_-_UMA

Convener: Eve Malek
Notes-taker(s): Tom Holodnik

USER MANAGED ACCESS

For complete details, please See:
http://kantarainitiative.org/confluence/display/uma/Home

The protocol flow is described here:
http://kantarainitiative.org/confluence/display/uma/UMA+1.0+Core+Protocol

Here's a friendly overview:
http://kantarainitiative.org/confluence/display/uma/UMA+Explained

History
Protect -Serve evolved into UMA
Last IIW, WRAP was presented that overturned some OAuth dependencies that UMA had had.

UMA:

Influences:
policy-decision making
privacy
information self-determination
data portability
"an open stack"
volunteered personal information personal data stores

outcomes:
a dashboard that allows you to control access
engaged data sharing

a protocol headed toward IETF applications area
a set of draft specs free for anyone
multiple implementations under way
simple, OAuth-based, identifier agnostic, REST-ful, modular, generative (can be used to build more things) and developed rapdily
targeting delivery as a spec (to IETF) in the August time frame.

The players:
Authorizing user - a web user who config's the AM with policies to control how to

make access control decisions)
Host (protected resource server) - enforces access to the protected resources it hosts
Authorization Manager (AM) - carries out an authorizing users policies
Requester - an entity that wants to access the AU's resources

Compare OAuth and UMA models:

the UMA model is different from the OAuth model is subtle ways; it establishes a contract for access management
the UMA AM is also offering IDP and discovery

participants
there is one resource owner and consumer in OAUth; the AU may be granting access to an autonomous party
resource server respects tokens from its authz server; the host outsources authz jobs to an authz manager chosen by the user
the authz server issues tokens based on the client's ability to authN; the authZ manager ussues tokens based on user policy and clienams coneryned byt he requester

provisioning
cleint and server must meet outside the oauth context to provision trust; the requester can walk up to a protected reseource and attempt to get access without registering first

dynamic trust
the resource server meets its authz server ahead of time and is coupled with it; the authz user can mediate the introduction of each of the hosts to the authz manager we wants to use
the resource server validates tokens in an unspecified manner, assumed locally; the host has the option to ask the authZ manager to validate tokens in real time

protocol:
oath: get a token use a token; uma: intro, get token use token
user delegation flows and automous flosed; uma: protfiles fo of oauth flows

relationship with oauth: based on oauth 2.

UMA Protocol Details:
(reference the links at the top of the notes)

Establishing trust; passing a handle to the protected resources
- could establish trust on first use (TOFU);

Policies
unilateral - allow access for a week
claims-requiring - "allow anyone access who agrees to my licensing terms" or allow

access to someone who can prove themselves to to bob@mailco.com, or allow access to anyone 18 years old or more.

Claims 2.0 are JSON based claims that establish attributes about a user; they don't have to be issued by the AM, but they could be issued by an IDP associated with the AM.

DEMOS and Implementations in Progress:

SMART at Newcastle University: This illustrates how to issue and manage simple kinds of claims:
http://kantarainitiative.org/confluence/download/attachments/38371737/SMARTOverview.pdf
http://kantarainitiative.org/confluence/display/uma/SMART+project+user+experience

Christian Scholz: This illustrates how we might create policies and provision access to resources we want to protect with an UMA AM:
Prototype: http://bitbucket.org/mrtopf/uma
Demo: http://host.clprojects.net/

if the token does not contain information about the resource (and to whom it was issued), it's vulnerable to confused deputy

claims confirmation could be as simple as "confirm that you are over 18" or "confirm that you will abide by the terms of Creative Commons..." - enforceable legally, or could be supported by claims issued through CardSpace/InfoCards,   could be a URL of a BBB statement, or a URL pointing to other indepedent assertion of claims.

Permission vs. Consent (M3F)

URL: http://iiw.idcommons.net/Permission_vs_Consent

Convener: Kevin Marks
Notes-taker(s): Stacey Pitsillides

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

- User name/ password model
- More comprehensible to people
- You were doing what with my stuff???
- Meaningful choice about what I'm providing
- How many users they've got to sign up/ not how many users understand what they've signed up for.
- What am I giving permission for this thing to do?
- Is this app gonna do something bad to me, do people I know and trust, trust it
- App invites – neg half of it
- The app is able to deduce who my friends are?
- Holding the app – personal context
- Quality, copy, disappointment
- Permissions you give to that app
- Disclosure- this app demands these things and you cant switch them off
- Binary yes or no!
- Info to help you make your choice
- "we need your e-mail" wait till you have the reason to answer the question - gets them more e-mail addys that you cannot use
- trade- want that- prove your value
- know when your gonna use it
- bring the user to you ?? design system so
- consent for sharing – abstract idea of groups
- delegating identity back to you – empathy
- statement of purpose
- the cake is a lie – Randy Farmer
- specific statements of claims – say the following things on my app-
- when you make an assertion…. Separation between app provider
- classify apps – different ?
- im a level 3 guaranteed app
- set of purposes mapped to a set of capabilities
- sensitivity of information – defaults – exception – purpose statements
- compared to categories
- redemption ? new app 0 is better then -5 ask forgiveness button /
- reputation decay ?

- post on my behalf
- this spams me
- promises, drag the canneries in – initial promise
- how you develop trust?
- Abuse mitigation tool
- Curated computing - social curation within an app store
- How do you scale trust without a legal framework ??

eCitizen Open ID National Architecture (M3G)

URL:http://iiw.idcommons.net/ECitizen_OpenID_National_Architecture

Convener: Dazza
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

OpenID Connect: Under the Hood (M3I)

URL: http://iiw.idcommons.net/OpenID_Connect:_Under_the_Hood

Convener: David, Joseph, Eran
Notes-taker(s): William Mills

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

- Signatures
  - ID Format
  - Delegation
  - OAuth Enhancements
    - immediate mode
    - userid param
    - display
  - scoping/audience compartmentalizations.
  - get user/?? API
  - (Doscovery)
  - Session lifecycle (cookies, etc.)
  - which OAuth 2.0 flows?
  - Browser discovery
  - Client discovery
  - extension model

Discussion points outside the above"

How do I trust a site to assert form domains other than the main domain? This is a significant discovery question. This is verification, not discovery. The flexibility of allowign an ID provider to assert for other domains is good, but the cost is that we have a harder discovery problem for all the sub or delegated domains. This is also the white label problem for white label products.

Are we willing to restrict this? Domains can only assert IDs in their own domain?

One of the nice things baout the URL format of an OpenID is that it's easy to get back to the sourcing domain. (David R.'s rant) This seqgues into "why isn't user@domain sufficient, since the URI is just a transform of that?

"So, what does OpenID Connect do?"

You need to end up with a domain and a scheme, form that you go do LRDD discovery. It gets back an OpenID token endpoint.

Delegation today is really aliasing, but this is "utterly useless" and "everyone gets it wrong". As yet there hasn't been a lot of demand for this.

"Is the stability of names a problem? What happens when a domain goes away?"

In the end the user needs to have multiple identities. The "rel" links allow you to connect between them. This needs to solve this, if one ID goes away you need to have a backup.

Scoping....

User impersonation is a problem. Scoping is needed so that a token granted to one party is not valid when used by soemone else. The way it's solved here is that the 3rd party is stored as part of the scope.

Does this support delegation?

What is supported is going from your preferred ID to an ID that is asserted by the provider.

------
http://openidconnect.com has the current stuff.
http://specs.openid.net has specs.

~~Using~~ Trying to use PubSubHubbub (M4B)
URL:

Convener: Mnica Kelen
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Private User Centric Claims (M4C)

URL: http://iiw.idcommons.net/Privacy_Enhancing_Approach

Convener: Peter Watkins
Notes-taker(s): Andre Boyser

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

It is important that the citizen knows, that the gov knows, who you are. For trust, efficiency, to commit the to process

How do they know who you are? Registration process, in process, deep check.
Govs have an interesting advantage from a longevity point of view (life time of events)
Process ends with the issuance of a card. But don't want a wallet full of cards. So…
Don't want universal handles (like drivers license)
But every program has a unique handle already – keep this
How do we bind credentials to existing unique handles?

How do we use a single credential, allow for cross program use, without having cross program handles?

INSERT SLIDES HERE ?

One issue may be card replacement --- how to handle legitimate card change without mucking up all the anonymized handles.

Contextual Identity (M4D)

URL:http://iiw.idcommons.net/Contextual_Identity

Convener: Sam Curren
Notes-taker(s): Sam Curren

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Expression and communication of Context must be encoded in a way common to the domain to allow for it's use by another party.

Context is the cross section of attributes and data points of several entities relevant to a particular situation.

Agents / Context Brokers can act upon context to provide contextually relevant options or actions. In a non automated world, this agent is ourselves, or our conscience.

Context includes attributes from and is affected by Social situations, legal considerations, and conventions.

Identity Lifecycle -Getting the Genie Back in the Bottle (M4E)

URL: http://iiw.idcommons.net/Identity_Lifecycle

Convener: Jeff Stollman
Notes-taker(s): William Mills

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

We have several classes of data:

Uncontrollably exposed
  Here we are throwing up our hands.
Controllably exposed (for example by contract)
retrievable/withdraw able/erasable
  Vanish is an example of a product here.

Is there another level here which is "Assured destruction"
  Disappearing Ink was a company in this space.

Data security becomes very much like a DRM system.

How do we apply this to Identity Information?

Another taxonomy:

- Permanent data that we might want to withdraw.
- Transient data that should have a limited life.

What controls do we have here?

- Contract
- Legal obligation

Can a trusted 3rd party help ensure te dustruction/privacy/revocation.

A key question, "what is the incentive for people to be good actors?" or what's the penalty for not obeying privacy restrictions?

Another interestion question of information lifecycle: when the data changes context you may really care. It can be a big problem. The privacy boundaries here are significant and the user needs to be in control.

Verified Attribute Scheme (M4F)

URL: http://iiw.idcommons.net/Verified_Attribute_Schema

Convener: Kick Willemse
Notes-taker(s): Chris Obdam

Tags for the session - technology discussed/ideas considered:

Attribute validation, AX 1.0 (1.1), Defining standard methods/levels of attribute verification, leaving the identity validation to the RP's. OIX.

AX - OpenID Attribute Exchange Validate Mode - draft van Google van 24 nov 2009 - http://step2.googlecode.com/svn/spec/attribute_exchange_validate/trunk/openid-attribute-exchange-validate-mode.html

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Methods of Validation

1. Self Assertion
2. Proof of possesion
    a. Challenge Response Token
        i. Email
        ii. Bankaccount
        iii. Mobile (SMS)
        iv. Postal Adress
3. Authentic Register
4. Official Statement
    a. Face-to-Face
    b. Passport
    c. Claim

Can a attribute also be validated by a organization that did not issue the information e.g. can Stanford confirm that I am a Berkeley student?

There is need for 2 things: 1. a addition to AX for the validation information: validator, validation date and validation method/level.
A way to check if the validation method is executed in the right way (OIX?)

How do you handle the liability for the correctness of the information.

Follow Up Questions:

- Will AX 1.1 support attribute verification ?
- What Attribute schemes will be used?
    - X500

  - HCARD
  - Soap/XML
  - -AX-Sreg
  - Other?
- What are suitable attribute verification methods?
- Open Identity Exchange OIX <> Open Attribute Exchange?

Personal Data Stores – PDS (M4O)

URL: http://iiw.idcommons.net/Personal_Data_Stores

Convener: Paul Trevithick
Notes-taker(s): Drummond Reed, Stacey Pitsillides

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

NOTES FROM: Drummond Reed

They are not necessarily the place that all of a user's data is stored, but more like a virtual directory and dashboard - a single point of control for sharing personal information. The actual PDS can actually get/set (or read/write) data from multiple data sources (e.g., health records from doctors/hospitals, financial records from banks/insurance companies, home records from tax authorities, etc.)

Paul explained that the Higgins Project is implementing PDS. This idea seems to be catching on quickly now, and some of the key challenges are now ready to be discussed.

LOCATION: PDS can run on any device or be accessed from any device - they are not necessarily based in the cloud, though they may be.

ENCRYPTION: PDS can stored data/metadata in the clear, or "blinded" so that it is encrypted. This raises many questions, e.g., how does data from legacy apps get into a blinded PDS? Does the PDS handle synchronization?

PERSONAL DATA MINING: This is much easier (or in some cases only feasible) if the data is not blinded in the PDS. New forms of aggregation are possible in which the aggregation is happening on the user side. This is also someplace where zero-knowledge-proof technology (ZKP) can be useful.

PERSONAL DATA TERMS/LINK CONTRACTS - This "turns on its head" normal terms-of-service (TOS) that sites offer to users today, but which users really have no choice to accept. A PDS ecology is potentially a way that users can, acting as a "class", publish terms for personal data sharing that sites will accept.

PERSONAL DATA BANKING: Once a user has the ability to store and provide access to their data, they can begin to "bank" it or make it available to "exchanges" so that it can realize its latent value, similar to the way a bank earns interest in your money. Exchanges can provide a service to extract data (with permission from the user) and then aggregate it and provide value to multiple parties (e.g., companies,

governments, non-profits, and the user) that is either very difficult or impossible to unlock today.
RELATIONSHIP TO INFORMATION CARDS AND RELATIONSHIP CARDS: PDS architecture is a close fit with Information Card architecture when either the cards, or data accessed by personal cards, is stored in the PDS. This is the Higgins architecture. PDS is an even closer fit with relationship cards (r-cards) where the data relationship can be dynamic, and thus can set up a persistent feed of data.

SOCIAL ADDRESS BOOK: This is a classic example of a PDS application -- users can share their own address records with each other. The result is a "Plaxo without the Plaxo", i.e., p2p address book record sharing without any company in the middle.

THE DATA MODEL PROBLEM: One big challenge with PDS is how to do the data model. The schema for all personal data is so large: how can it be modeled? And how can it be mapped to all the various places from which the user is going to want to publish and subscribe data? The Higgins approach is to maintain a mapping at the PDS (that can be shared by a large population of PDS). Paul shared a figure that from one industry participant that a set of 15,000 form-fill mappings for websites had a breakage rate of 125 per day. That means that a team of two developers could maintain the mappings for a significant percentage of high-usage websites.

CORRELATION MANAGEMENT: Another advantage of PDS is that it gives a user a place to manage correlation between identifiers.

SEMANTICS: Yet another advantage is that a PDS (or a PDS ecosystem) can help develop and standardize the semantics used so that the services using PDS can be much smarter dealing with "things" rather than "strings".

PDS APPS: Many apps can be built on top of a PDS. One example demonstrated by Azigo VP Engineering Mike McIntosh was a password manager where all the usernames and passwords are saved in your PDS. This same app can be available on multiple devices, including mobile phones, to automatically authenticate you using your PDS as the "sync".

NOTES FROM: Stacey Pitsillides

- data sources
- users/service providers - everything about you – shift it to you??
- Dashboard – centralizing control not necessary geography
- Ok to give '…' access to these photos - relying party
- Concept of personal data stores
- Businesses made possible because of it.. – stock market of data
- Personal data banking – take it back?? Financial markets (trust to be your bank) - ie google - business aspects – build it?? Diaspora group If they don't interoperate?

- Multiple streams – local vs something that follows you same digital user accounts?
- Cloud service talking on your behalf – where does it live? Master copy?
- Synchronization service, blinded data store, employees have no access to your data. Your data is in your devices and in the cloud they can't read it! Encrypted link (blinding) – co-operation?? How do you share if all the data is encrypted? Copy which is encrypted, which then unencrypted … theory
- Q: protocols?? Speak
- Store data there it's fine
- Challenge the assumption that the data needs to be encrypted… prevent data mining – aggregated data , layer above it
- Consensual where my data goes! Just happens … provide a 0 knowledge proof
- Analogy to the financial system
- User consensually say 'yes'
  Turn Facebooks terms of service upside down You can have access to my data but you can't cache it
- Digital signatures on a legal policy
- One way functions (privacy – secrecy?) - no audit, you don't know what happened to your data, trust frameworks. Protected by an agreement on what happens to data, audit logs, what we did with our data - prove!
- Natural element of an identity management system. Directory service
- Person – set of correlations – lets me say that this character is that character and that account – meta persona
- Narrow context – persona – multiple set of interactions – trust (non profit) incentives – different - store data in order to leverage it
- Stored value system – same store – hold the key
- Everything goes to my store first - Get a copy into my store
- Could your identity to be stored in a myriad of places- from client to store to world – one data flow ? (key) pair ? public/private - information card in the cloud ? problem- it's on one machine, the card itself is a piece of data, card store, active client – relationship card – gesture is here's some stuff – one time push – include one more claim – the pointer to the stuff – if I trust you build a rel with what I give to you, when you hand somebody, one shot copy of v- card data pointer back to… delegated – social address book (plaxo, done right??)
- How are we gonna figure out the uniqueness of this data storm? How does it work ? transport problem … fragmented and splinted , central dashboard

- Let me consol the policies - common data schema (fail) simple, have wide distribution, doesn’t do much ?? Higgins data model – persona data model – mappings, because the world is how it is build mapping tables and mapping rules. Common data model, bi-directional mapping rules, hides a multitude of sins – master schema – ‘it can’t be done‘ ? concluded you don’t have to convince the world, massive energy for massive consensus
- It can maintain its own mappings, at a cost of having to maintain Open source – mapping is propriety – next company builds their rules – collaborate
- Internationalized domain names and strings – include
- Not indeed unique – redesign – how unique the interaction you engage with ?
- If you live on the e-commerce personalize the msg for each one is insane
- Can’t care who you are .. intermediates
- Meta data – i-phone “from strings to things” – meta data along the data string
- A person at a place, at a time
- XTI – synchronize info – client libraries – password manager, Higgins pd store - your data is never written down? Issue of duty
- What if you lose the key????? What happens to your data ? Google pass phrase recovery, password reset no/ coz they don’t know it!!
- Demo: save values into personal data store, same account for different devices, it will be synchronized – icons that show its recognized – 1st time visited – saved an account there – recognized that it s a user password field .. remember – simplest app that we could think of – simple schema

Voice Biometrics for Anonymous Identity Proofing & Authentication (M5D)

URL:

Convener: Dan Miller
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

VRM Parts and Holes (M5E)

URL: http://iiw.idcommons.net/VRM_Parts_%26_Whole

Convener: =DOC
Notes-taker(s): Markus Sabadello

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

ProjectVRM grew out of IIW
- Individuals should be able to control relationships under their own terms
- Concerned with interaction with commerce and government

User = Point of Integration

Change the marketplace where we as customers have more to contribute

CRM: 12$ billion business.
"Customers are creatures you hunt, seek, lock in, acquire, manage"
= language of slavery used by the sellers of the world

CRM Magazine: "Customer does not belong to you."
Change happening in CRM business!

CRM is not gonna like VRM? VRM = reciprocal?
No! Venn diagram, they are part of the same thing.
Buyer and Seller both gain from it. Power to all parties.
Advice to CRM people: Learn from it, take advantage!

Projects: MyDex, Mine, Switchbook, Azigo, Kynetx, Banyan, etc..

EmanciPay: For circumstances where there is not already a price, is there a way where we can say "Here is what I would like to pay" ?
--> Web site owners can put RDFa code on websites that points to where you can be paid
Mechanism to determine prices on the fly
Empowers the consumer to say "I want to pay this for that"

Legal structure in place for VRM?
No, but there is a lot of attention, e.g. in Kantara DataSharing group.

End of phase 1 of e-commerce! (Cookies, absurdly long terms of service, giving all the rights to the seller).
Can we have a casual relationship with vendors online? Like simply walking into a shoe store, without having to become a member etc.

European privacy laws more friendly than American ones

Foursquare TOS: If we're sold, your data will be one of our assets and also sold.

William explains situation in UK: Issues around ID-Cards, centralized health records, data theft.
UK government interested to move towards something like VRM. Times are right to propose a working community prototype where government agencies act as relying parties.

Iain gives example: "Tell us once" program by government. Works, but is run by a government site, not in a user centric way.

2 important trends for VRM:
- self tracking
- personal informatics

Powers are becoming more equal, abilities are increasing.

Need code to move forward.
Need for figuring out the legal side.

VRM = extra money for sellers - potential additional customers that want to pay on their own terms.

Linking Data Across Social Nets – API's (M5F)

URL: http://iiw.idcommons.net/Linking_Data_Across_Social_Networks_APIs

Convener: Rohit Khare
Notes-taker(s): Laurel Fan

Tags for the session - technology discussed/ideas considered:

Tags: APIs, interoperability, implementation

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Rohit gave a brief description of his company's product: knx.to. The short description: it uses the APIs of various social networks the user is part of, and aggregates the user's social graph over all of them.

Next a round of introductions. Some projects represented:

Dreamboat - Mozilla
Familysearch - geneology
Opensocial foundation
IBM services
Amazon
Games
BT - social network aware telephony
43things - community site

Questions:

1. is there business value from linking data?
2. how? how to make it open and standard so anyone can do it?
3. combine graphs of people (the 'address book' metaphor is out of date -- relationships mean more than contact info)
4. as service providers, how to correlate identities using the partial data available (ie. is "Rohit K" the same as "Rohit Khare")
5. is there a policy layer -- "copy rights" for portability, reversibility

The list of APIs (some used by knx.to, some introduced in the session):

facebook
linkedin
twitter
myspace

gmail
yahoo
flickr
msn/windows live
hyves

see also the chart that Rohit should be emailing

Issues/problems when integrating with specific APIs:

facebook
- privacy settings are not computable (you can't calculate whether you are allowed to share information you got from facebook -- you have to make another API call to ask them)
- you can't get email/phone numbers unless you are special

linkedin
- it's hard to look up by email
- the data retention policy is unknown -- they can audit you but the policy is not well specified
- heavily throttled

yahoo
- short lived sessions: 30-60 minutes
- user gets an email notification every time you get a new session

myspace
- friends list is first name only

One of the ways to implement the "which of my friends is on this site" is by searching by plaintext email. However, using plaintext email is overkill, since you don't need to reveal that information. Can use hashed email instead.

Webfinger is another way to interoperate between social networks using the email address as the key.

The problems to solve are:
- technical
- business
- legal

Six Degrees of Sharing (M5G)

URL:

Convener: Alan Karp
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

OAUTH 2 for SASL (M5I)

URL: http://iiw.idcommons.net/OAuth_2

Convener: Bill Mills, Allen Tom, Joseph Smarr
Notes-taker(s): Bill Mills

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Extended discussion will continue athttp://tech.groups.yahoo.com/group/sasl_oauth/

The proposed spec is at http://docs.google.com/View?id=dhjg77m3_0gsn2psdq

We walked through various parts of the proposed spec. There were questions about how it fits in and how SASL fits in to other protocols. The general feeling was it's better to use SASL than to implement OAuth separately in each protocol.

It was pointed out that the IETF drafts:

1. draft-lear-ieft-sasl-openid-01.txt
2. draft-wierenga-sasl-saml-00.txt

Both will have useful as they are very similar problems.

Interesting questions asked:

How do we sort out what OAuth 2.0 auth flows are supported? Much of this may shake out in the OAuth 2.0 discovery information stuff. If not, then the SASL spec should patch this up until OAuth 2.0 gets there?

Is there anyone out there that want's to do OAuth 1.0a over SASL or will you just go to 2.0 to get SASL? Consensus was that everyone will go to 2.0.

There was a lot of discussion around the question of taking an OAuth token and authenticating a session with it, and the implications of that. Major points of concern still to be hashed out are:

What if you us OAuth an XMPP session that allows password change?
Should sessions be limited to token life/expiration?
How does does a desktop client get a client_id and secret when desktop clients can't really keep a secret?

Quite a bit of good feedback.

ORCID Open Research and Contributor ID (M5K)

URL:

Convener: Geoffrey Bilder
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Day Two - Tuesday May 18th 2010 Sessions

What About OpenID for Organizations That NEED Strong Authentication? (T1C)

URL: http://iiw.idcommons.net/Strong_Auth_and_OpenID_getting_Comfie

Convener: Eric Skinner, Entrust
Notes-taker(s): Eric Skinner, Entrust

Tags for the session - technology discussed/ideas considered:

Strong auth

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

The issue: banks, governments and other parties operate sites where high-value transactions take place. These organizations have been resisting OpenID, due to concerns that OpenID is not sufficient for high-value transactions. This session sought to clarify the reasons why.

Philosophically OpenID is viewed by some as a framework that empowers users to choose whatever OpenID Provider (OP) they want. Since OP's vary widely in terms of enrollment and authentication policies, clearly sites that require strong authentication for high-value transactions will seek out specific OP's that they trust.

While the "dynamic trust model" (use whatever OP you want) works well at low assurance levels it's not well-suited to high assurance. But OP's can be vetted (directly by the RP or via a trusted third party) then RP's can decide to trust them to deliver on particular authentication strengths.

Core question: should banks allow OpenID for high-value transactions, or are their (frequently raised) concerns valid? People in the session quickly said "the concerns are valid."

There are four key questions:

1. How strong is the OpenID set of protocols? The experts feel that the current protocol is not quite strong enough, but is getting improved in the next rev. At the last IIW, Paypal did a presentation on some of the issues here. The US Govt has said that given these challenges the current OpenID is not well suited for transactions that need auth strength beyond assurance level

2. How well did the OP vet the identity? This needs to either be audited externally by a trusted party, or relying parties can establish a contractual relationship or other trust relationship with the IdP that makes them comfortable.

3. How does the OP perform individual authentication? As per point 2 above.

4. And how secure is the OP over time? As per point 2 above.

The OpenID provider site is vulnerable to spoofing; some strong auth techniques such as Out of Band confirmations can be used to combat this.

Why would a bank want to rely on OpenID anyway? Well, the bank has an ongoing relationship with the customer, who uses that credential frequently, so they likely won't see customers wanting to use another credential at the bank. On the flipside, a bank credential would be useful for logging in at a government website

Banks might not want to be an OP (How do they get paid? Or otherwise get value for having offered that service?)

Why do people want to adopt OpenID anyway, if it doesn't quite meet their needs? Shouldn't they just use something else? Well, some governments for example want to get out of the identity management business, and specifically the expensive processes related to registration of users. So OpenID holds out some promise, even if it doesn't meet the higher-level security requirements yet. Also, governments suffer from infrequent interaction with citizens, so would benefit from relying on a more frequently-used credential.

Many of today's authentication and communication mechanisms have weaknesses but people accept those out of familiarity. For example, we trust many things done over email (e.g. an emailed PDF contract). But when something new such as OpenID is introduced, everyone naturally stops to consider the implications.

In some ways the OpenID model can provide better security than if RP's try to do their own authentication. For example, the OP can spot patterns of activity across multiple sites that help identify an attack more quickly. And of course, the OP can be a security specialist while many RP's are not.

Information Cards and Government IDs (T1D)

URL: http://iiw.idcommons.net/Information_Cards_and_Gov_Cards

Convener: Drummond Reed
Notes-taker(s): Marc Licciardi

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

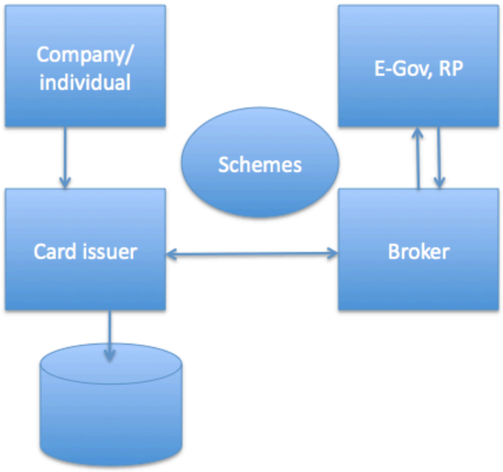
2 weeks ago at the European ID conference. The German government is rolling out the Government ID card for all citizens for 100% of the citizens. It will be an electronic ID card that works on line. The European standard is that the government issued very strong ID and they have been working on how they can use these ID's for online ID.

German privacy laws presented a strong conflict. But the EU is working hard on how government ID cards can be used for online ID.

Clarification: They are issuing physical cards, but what is the link to online ID. There are different solutions. Some have an online reader that can read your car, but others have an electronic version. The German government is promoting industry development and propagation of readers.

Netherlands and BC government have programs they are working on and they will explain what they are doing and then open up to the group.

The Netherlands are doing something slightly different. They will not issue a card. They say if you have an e-government service that uses a broker that can connect to card issuers, which can issue cards to companies and individuals. One person could have multiple credentials. So there is freedom of choice for multiple accredited providers, but when the individual interacts with the government they need to have some ID from accredited service.



Companies can then use power of attorney to get cards on behalf of their employees. They are opening it up to B2G, B2B and B2C and C2B.

They are deploying a citizen to government record access to health records. There is a unique scheme for accessing the health records. The main reason for their thinking was that the limited use ID card was not well adopted. So they are identifying more useful cards.

What Types of businesses can get credentials: Lawyers, Banks, Telecom providers, specialist providers, some will produce one level of assurance only and others will produce higher LOA. The fee between the RP and Broker is based on the level of assurance.

They are pushing for a centralized fee setting. Part of the scheme is that there is some recourse back for failed identification. This is described as a contractual model.

Norway has a system that uses the Bank ID model that is restricted to the banks, which capture 99.7% of their citizens. In Norway sitting next to those banks systems they have additional systems.

In the Netherlands the government will accept any of the accredited agency.

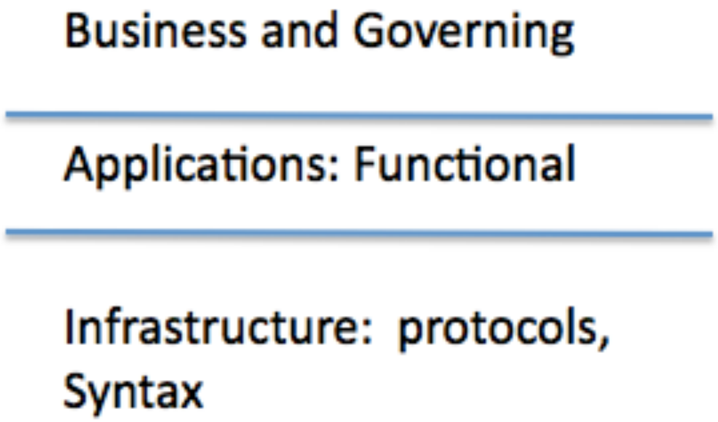
In tax filing the accountant is given a power of attorney to act on behalf of the filer, so there is a credential entity interacting with the government.

From a usability perspective, a user goes to a government part and logs in, and then relying party validates the through the broker to the credential issuer to validate the person and that validation is returned to the broker back to the relying party.

The Dutch government is not going to issue ID cards because it is relying on a variety of already issued ID cards. So there is a business incentive for issuing parties to be part of the credentialing process.

How many schemes are there in the NL, there is just one scheme, any credit issuer that takes part in the scheme can answer.   There are 17 market players to develop the market rules for the schemes.

A large like Shell can apply and enter into an issuing role.   They are accredited in both a legal, technical and business details.   So there are technical profiles and policies.



Business and Governing

Applications: Functional

Infrastructure: protocols, Syntax

The term scheme in the NL is layered in three layers, which are infrastructure and protocol agreements, Functional agreements and then Business and government agreements.   This three-layered set of agreements is called a scheme.   The government's role is in a collaborative development stage, which will be transferred to a scheme government, which is controlled by the participating companies, which will be audited externally.   Above these three-layered agreements are the competitive propositions in the market place.

There is also a legal impulse in Europe to mandate that the top two layers have to be able to cooperate on the top two levels.

Large companies can fill the roles in the card issuer and broker stages as well as long as they can meet the assurance levels, but they are allowed to get credentialed.

When you start to make it a business network the layers become very interdependent. All of this is being built on top of existing legislation as the foundation.   There are some discussions on passing specific legislation because some of the interaction is on the edge of the expectation of the laws.

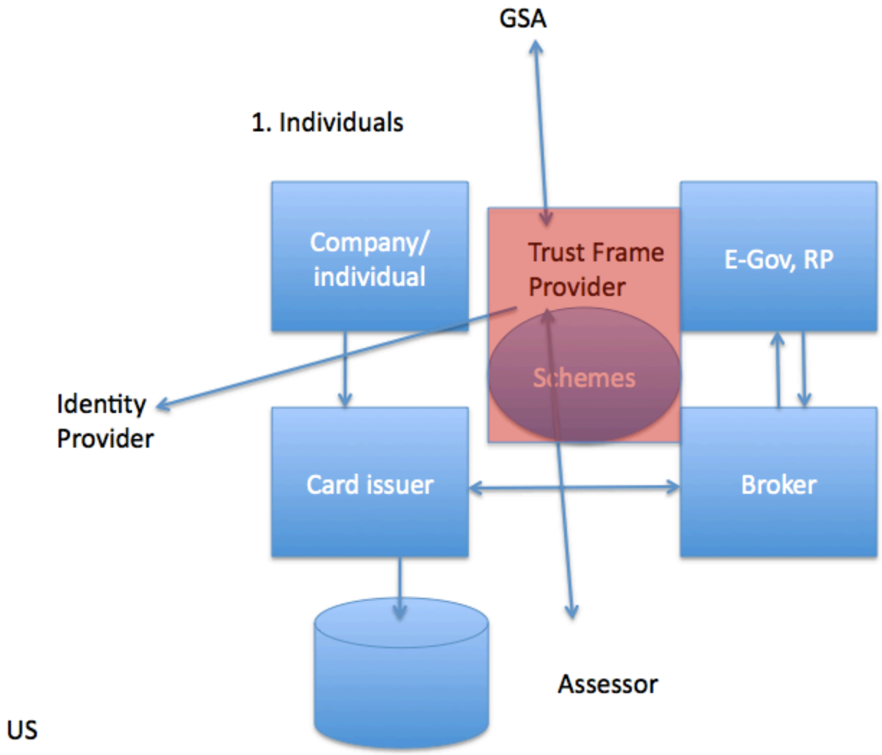
What's happening in the US for open government in the same roles.   In the US,

In OIX terminology.   The three layers are called the Trust Factor

Point 1.   In what the US government is trying to solve

In this program the Trust Frameworks is broken into Technical Profiles, of which they did two, one for Open ID 2.0, which is right now approved for LOA1, and for IMI approved for 1,2 and 3; There is also a SAML profile available for 1,2,3, and 4.

Trust Frame providers, OIX and Kantara.



The GSA anticipates that there will be deals struck between the IDP and the GSA for the higher assurance levels.

In the NL the cost of issuing the credential can drive the assurance level requirement so that the high cost forces the consideration.

In the US government there is a risk assessment done on RP that says you must have this assurance level.

In the NL model the relationship is between the IDP and the Broker and the broker has the liability.

Right now OIX has not necessarily been the hub to provide the higher assurance level.

In the NL the interchange fees can be negotiated between the broker and the IDP.

The RP can go to the broker and get the fees for the varying credit issuers. The Canadian government is looking to set up a similar relationship. But the US government is looking to make deals directly with the IDP.

For Inter-federation in the EU as large company you can get accredited to participate in one of the various roles or use an existing accredited entities.

The relationships in the NL are based on two party trust at each of the legs. They can agree on the security, there are some rules.

There is an attempt to make the SAML more of a feature in the trust framework by ICANN. It will utilize SAML metadata.

De-Confusing: High Level Overview (T1E)

URL: http://iiw.idcommons.net/De-Confusion_Big_Picture

Convener: Kaliya Hamlin
Notes-taker(s): Aaron Bronzan

Tags for the session - technology discussed/ideas considered:

Overview of Identity, Standards Organizations, Acronyms

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

"De-confusing" Identity (5/18 session 1)

"On the Internet, nobody knows you're a dog" (IIW logo)

- Anonymity is important
- But people need the set of tools to be able to represent who they are (at varying levels of granularity/disclosure)

Communities in attendance

- Business
  - Enterprise Customer
  - Enterprise Identity Management Product
  - WebPortals (e.g. Google, Yahoo, MSN, LinkedIn)
  - Regular websites
- Government
  - Europe, BC, DC
- Standards Development Community
  - OASIS (InfoCards, SAML, XRI/XDI)
  - IETF and Internet Society (SMTP)
  - W3C (HTML)
  - ITU-T (phone) and ISO
  - "Floaters"
    - XMPP - Jabber
    - OpenID
- Sysadmins
- Web Developers
- Etc. Etc. Etc.

- Enterprise identity management: Where it all sort of started
  - Provisioning/issuing credentials for use of internal enterprise systems
  - e.g. username, password, auth token, etc.

- SAML (Security Assertion Markup Language): Directory of employees with specific privileges
  - Authorization, or AuthZ (What you're allowed to do)
  - Authentication, or AuthN (The identifier – the username you use, etc.)
  - Verification
  - Enrollment into system (new users)
  - Termination from system (ex-users)

- SAML Federation
  - Business to Business sharing (e.g. American Airlines + Boeing)
  - Trusting each other's credentials
  - Doesn't scale well

OpenID = outsourcing username and password (same "username" or i-name)

- Problem is phishing: Fake forms for OpenID providers
- Therefore, OpenID is designed for low-security transactions

NASCAR problem: Addresses challenge of usability with OpenID (logos instead of having to remember your OpenID URL)

Info Cards

- IDP issues card, or you make your own card
- User selects cards
- Open Source InfoCard Selector repository: Higgins Project
- Send various attributes only, customize the amount of information sent

OpenID + Information Cards = Open Identity Exchange

XRD is Discovery: A protocol for understanding and discovering services

We then went over a bunch of the organizations and how they relate to each other. See Kaliya's flowchart slides for an overview.

Past, Present, and Future of Genealogy Software (T1F)

URL:http://iiw.idcommons.net/Open_Geneology

Convener: Gordon Clarke
Notes-taker(s): Judith Bush

Tags for the session - technology discussed/ideas considered:

Genealogy, death, identity, interoperable citations, remixing, OAuth, OpenID, OpenAPI, GEDcom, digital death, digital heritage

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Discussion began with an overview of the FamilySearch.org role in the genealogy space, in providing a definitive ID for persons. After the overview, there were recommendations for the web "consumability" of the FamilySearch IDs.

Gather & Share: Artifacts to boxes
Records (census, county), trees (the relationship data), artifacts
Share: copiers, email, now websites &blogs

Digitize: Boxes to Computer

Share: Computer to Web

View All (mashing) Web to Computer

QUESTIONS
Are there deduping methods? Online and offline software offerings. GEDcom parsing and standardizing libraries in open source from familysearch.
Goal is to be best definitive source with n identifier that can be used broadly.

Users of 30+ popular products can collaborate with FamilySearch
Data from PAF and Gedcom can migrate in.

Synchronization with Family Search: search for matching people from 3rd part client, assciate person on desktop with person in FamilySearch, Add new people to FamilySearch, Uploiad changed person to FS, download FS person info to desktop. MANUAL merge because individuals have different requirements.

The community can become, collaboratively, authoritative.

Richer Shareable Family History:

- ○ Attachment of artifacts
- ○ Mobile convenience (work incrementally)
- ○ Social Networking (SharingTime, Kynetix)
- ○ Private group collaboration (Public information and private; now able to share with a limited group)
- ○ Community Links and Web resources

REMIXING

- ○ Group trees
- ○ Group Geo-mapping (Names in Stone)
- ○ Group Media Solutions (PhotoLoom, AppleTree, OurFamilyology)

The decorator pattern – digital heirloom project with Microsoft
First class objects on persons, events, places

Webresources: photos, records, DNA, Geo, Audio-video

If webresources can search trees and save with FamilySearch IDs, certified with FamilySearch.

What about place and events? FamilySearch would like this sort of things.

[Guided conversation ends now and moves to open discussion]

First class object: given the identifier, you can go get interesting stuff.

Searching led in past, linking growing in importance (in the genealogy domain). Family search has person identifiers (the Family Search ID): good leverage place. People will want to interlink to established research.

Remixing with a match type a person ID and some other ID gives a URL parameter. Matching on a locality? Yes.

Other companies want to create place and time information.

Given microformats things can be spidered.

FamilySearch is invisible to google. Interoperable citation will be bibliographic metadata.

Concern that FamilySearch has saved search URL (not Restful), need permalink URLs. Answer: genIDs?

Discussion of the language mapping between the identity groups and the genealogy group:

Person and persona, claims vs conclusions with evidence.

Reputation based recording of conclusion/evidence to personas/persons.

Challenging that the reputation system should resolve into the best to meet the Read Only use case.
Optimization is for read-write, curator

FSData.familysearch.org is the demo back of coming interface

Migrated to OAuth. Need to straddle between the access restrictions of legacy data with older familysearch proprietary data. Question of binding between existing system user IDs and OpenIDs. “They know mother’s maiden names.” Implementing Aliasing.

XRD Provisioning (T1G)

URL:http://iiw.idcommons.net/XRD_Provisioning

Convener: Jared Hanson
Notes-taker(s): Jared Hanson

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Links:
xrdprovisioning.net

Topics:
How to identify the link?
-- use the xml:id attribute or the href:type:rel tuple
-- href:type:rel should be good enough but xml:id is the purist solution
-- consensus to use the xml:id to identify the link rather than matching the href:type:rel tuple
-- the POST of the <Link> can request a particular xml:id but the service can override the xml:id and return it to the caller

Ownership of who is allowed to update which links
-- Use OAuth to protect the REST APIs
-- proposal to add an extension element "dc:owner" to the actual link element

Is there a need to identify what the protection mechanism is?
-- maybe a separate doc to map to HTTP Basic or OAuth
-- leverage the WWW-Authenticate header to identify how the

Need to make sure that an attacker CAN NOT update someone else's <Link>
-- this is a critical security requirement

Request to support a form-encoding mode for simple addition of links
-- only support for limited <Link> elements

JRD should be out of scope for now
-- eventually make it an optional encoding

Define a rel type to represent a visual editor for the XRD
-- defines a relationship between the user and their user management page

Building Mitre ID (T1H)

URL:

Convener: Justin Richer
Notes-taker(s): Justin Richer

Tags for the session - technology discussed/ideas considered:

openID, enterprise, corporate identity, recycling, pseudonymity

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

MITRE is deploying as an experimental prototype an OpenID server for corporate usage. This service will be available both to internal systems that can't or won't be a part of the official SSO yet still want to externalize authentication as well as external systems that are not under MITRE control. Only MITRE employees will ever be provisioned identities on this service, and usage of it will be tied very tightly to existing MITRE credentialing systems. We're building it as a Java servlet and hope to release our OpenID server implementation as a reference implementation for other organizations to use. Perhaps as a kind of "Apache server for OpenID". One of the great benefits of OpenID is that the developer of an RP need not register with the administrator of the OP in order to use the authentication system. This is especially important for rapid prototyping of new systems internally as well as external systems with which the company has no prior engagement.

What MITRE is doing is surprisingly unique, but we're not sure why that's so. Today companies issue employees an email address and a telephone number as a matter of course, and these are definite tools for engaging with the outside world. We want to explore the idea of also issuing all employees distributed identities such as OpenIDs that they can use at their discretion with outside systems and services. The policies for use of such systems should probably reflect the policies surrounding usage of corporate email accounts in the outside world.

Sun had previously done OpenID@work, which was a very similar program. However, in Sun's case, the system was opt-in and employees could choose their own identifiers. MITRE is building this in a way that employees are simply issued OpenID credentials that they can access immediately. The government of Estonia issues its citizens OpenIDs that are strongly tied to a national identity card system, and many government applications in that country use it. Google is using OpenID internally for some things by nature of using the Google App Engine internally.

There are open questions on where to tie to existing identity systems in the enterprise. Sun tied to existing systems at credential issue time only, while MITRE is tying in both the credentialing and authentication steps. There are real issues surrounding recycling of identifiers and the usability of opaque identifiers, though webfinger starts to address these.

The US Government is working with the ICAM initiative to determine trust profiles for different protocols and implementations. OpenID has LOA1 profile already in place, and MITRE is interested in pursuing higher levels of certification for MITRE's instatiation. OpenID is susceptible to the same DNS poisoning attacks that most internet-facing systems are, even if these other systems (like SAML) don't want to admit it. In addition to tying certain attributes (a unique identifier and an email address) to a user, the LOA profiles can give rise to the idea of "how real is this person".

Whitelisting of OPs and RPs is an important capability of a system like this, and distributed whitelisting and the OIX model can help this to scale. MITRE is planning on allowing all RPs but whitelisting the RPs of trusted partners in its implementation. MITRE is also looking at what it would take to accept the OpenID credentials of other companies in its external-facing systems.

The notion of "Trust on First Use" (or TOFU) was brought up as a very usable pattern that is supported by the OpenID trust flow. In this method, users decide to trust an endpoint on its first usage and simply verify continuously after that first use. Chris Palmer has given a talk recently about how to fix HTTPS that goes into detail on this topic.

OAUTH 2.0 and SASL (T1I)

URL:

Convener: Bill Mills
Notes-taker(s): Bills Mills

Notes:

Good discussion about whether this is actually needed given the OpenID/SASL proposal.

- There seem to be different use cases that make both useful.
  - A significant difference is the durability of tokens.
  - Another is that in the OpenID case delegation is easy, admin@myblog.wrdpress.com being delegated to any domain for authentication for example.
  - OpenID really issues one time tokens.
- Discussion of both and what the characteristics of each are.
- Talked through the use cases for each in the context of a Mail server, and found that we really think there are use cases for both.

Infogrid Graph Database (T2A)

URL:http://iiw.idcommons.net/Info_Grid_Graphic_Database

Convener: Johannes Ernst
Notes-taker(s): Johannes Ernst

Notes:

Johannes gave an overview of graph databases and InfoGrid.org on the white board, similar to http://www.slideshare.net/infogrid/info-grid-core-ideas

Lots of Q&A and discussion on nodes, edges, dynamic types, models, and graph scaling via XPRISO.

More info: http://infogrid.org/

UMA and the Law (T2B)

URL: http://iiw.idcommons.net/Legal_Issues_Underpinning_of_UMA

Convener: Jeff Stollman
Notes-taker(s): Eve Maler

NOtes:

UMA main site: http://kantarainitiative.org/confluence/display/uma/Home
UMA unfinished Legal Considerations document: http://kantarainitiative.org/confluence/display/uma/Legal+Considerations+in+UMA+Authorization

Attending: Jeff Stollman (leader), Heather West, Iain Henderson, Eve Maler, Mason Lee, Judith Bush, Alex Smolen, Stacy Pitsillides, Brian ...worth (? - didn't catch), Mark Lizar

Some participants in the UMA Work Group at the Kantara Initiatives have been focusing specifically on legal considerations. One concern is scalability. Even if we constrain UMA initially to something that will work simply, we want it to scale much bigger eventually.

UMA holds out the prospect of an individually negotiated contract between an authorizing user and a requesting party, which starts out with the user's wishes being the initial terms. The user's wishes can be carried out by an "authorization manager" that decides whether a requester application deserves to get an access token (a la OAuth) for accessing some host.

Liability is the place where all federated identity seems to fall apart! As Tom Smedinghoff has pointed out to us, we need to figure out what legal theory of liability should be in play. E.g., there's contract, tort, negligence... Heather explains that contract law doesn't see the "I Agree" clicking process as an example of a contract. It's just terms of service (clickwrap). The OIX approach is going in the direction of contracts, which is much stronger. The Computer Fraud and Abuse Act is what has been used most often to prosecute TOS violations, same as for prosecuting hackers -- and TOS is simply not very strong.

However, if UMA were used to ask for positively asserted claims vs. just a user interface that asks for "click to agree", or if UMA were deployed within a trust framework that is contract-based, it's possible to apply a contract theory of liability.

Iain's Information Sharing work at Kantara, and his work at Mydex.org, focuses in part on "volunteered information". His lawyers have said that volunteered information with an advertisement of the terms would use contract law. Europe has the eight principles of privacy protection, and if these principles are part of the offered contract terms, it becomes quite strong.

Contract theory is the only one that scales internationally. Various boundaries are relevant to this question: domestic, treaty-member nations vs. non-treaty nations, etc. Again, certified parties to a trust framework are a strong way to get some level of contract protection.

Eve sketches two areas of UMA that seem like they would have impact on the liability theory used.

One is the particular user experience on the requesting-party end. E.g., if it's Bob (person-to-person or Alice-to-Bob sharing), we don't necessarily want to make him agree to the same privacy policies, to the same "strength", as if it's a company (person-to-service sharing where the service is run by a company acting on its own behalf). And you might have an "I Agree" button for Bob, but not the company.

The other is the particular method that Alice uses to provision the requesting party with knowledge of the resource. The Data Dominatrix method has Alice pasting (or whatever) a URL in the recipient's interface and the latter discovers the constraints on trying to get the information. The Hey Sailor method has Alice advertising something like a personal RFP, and Iain notes that if the advertisement also includes the offered terms, that would use standard contract law.

Brian points out that the UMA proposition is a lot like DRM for personal information. Alex observes that the power imbalance between people and companies seems to make that okay. :-)

If you want to have a contract between the authorizing and requesting parties, but the intermediary parties only have pairwise TOS's, the TOS's can weaken the contract at the ends.

The UMA protocol uses the technical notion (not the legal notion) of "claims" for finding out more about the requesting party to figure out if they qualify to get access. What if the authorization manager demanded a claim saying (in a verifiable way) that the requesting party is a certified member of a trust framework? This could mitigate the risk of having any part of the ecosystem using TOS's versus a contract.

Heather points out the example of the Veteran's Administration, which is incredibly successful with its e-health records because the entire ecosystem is under very strong contractual privacy protections, which makes it easy for ordinary humans to understand and consistently applicable by other parties in the ecosystem. Could it be the case that more stringent but more consistent privacy controls could be good for the growth of the commercial market! However, recognizing that "privacy" and "security" generally aren't good selling points, it's hard to use this rationale for business benefits.

What would incentive hosts to accept an authorization manager's policy decisions, and all the liability implications thereof? The theory of the UMA folks is that most websites offer terrible (or no) selective sharing options, and if they could offer it as a value-add simply by adding an "UMAnizing" module, that would be attractive.

How would the changes in various contracts in the ecosystem be handled? UMA could help an authorizing user decide to revoke access, but what if other entities in the system change their policies? If there were a standard for giving notice of changes (maybe using Atom feeds?), that could be used.

Contacts In The Browser (T2E)

URL:

Convener: Dan, Mike, Ravavan
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Migrating From HTTP to HTTPS Open ID (T2F)

URL:http://iiw.idcommons.net/Migrating_from_HTTP_to_HTTPS_OpenID

Convener: George Fletcher
Notes-taker(s): George Fletcher

Tags for the session - technology discussed/ideas considered:

OpenID, HTTP OpenIDs, HTTPS OpenIDs, Upgrading

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

XRDS: https://api.screenname.aol.com/auth/openid/xrds
XRD: http://www.aol.com/.well-known/host-meta

Type URI:
   http://specs.openid.net/auth/2.0/httpMapping

When validating an OpenID via discovery and XRDS

```
  <XRD>
    <Service xmlns="xri://$xrd*($v*2.0)">
      <Type>http://specs.openid.net/auth/2.0/httpsUpgrade</Type>
      <URI>http://openid.aol.com/gffletch</URI>
   </Service>
  </XRD>
```

Example mapping for XRD

```
<?xml version='1.0' encoding='UTF-8'?>
<XRD xmlns='http://docs.oasis-open.org/ns/xri/xrd-1.0'>
    <Subject>https://openid.aol.com/gffletch</Subject>
    <Link rel=openid' href='https://openid.aol.com/gffletch'>
        <Property type=’http://specs.openid.net/auth/2.0/httpMapping’>http://
opened.aol.com</Property>
    </Link>
    <Link rel='describedby' href='http://profiles.aim.com/gffletch' type='text/html' />
</XRD>
```

OP defines <Type> URI in XRDS/XRD specifying that it implements the HTTP->HTTPS upgrade path.
   http://specs.openid.net/auth/2.0/httpMapping

Processing rules for the OP
1. When the discovery request is made over HTTPS, then return the alias type and associated http OpenID.
2. If the initial request for the OpenID flow come in over HTTP, then 301 to the HTTPS identifier

Processing rules for the RP
1. When the OpenID assertion contains an HTTPS OpenID the RP first looks to see if this OpenID is known. If found, then done.
2. If the HTTPS identifier is NOT found, look in the discovery document for an "alias" type in the discovery document
3. If the Alias is found, the RP looks for that OpenID in their store
4. If found, then update the user id in the RP data store from the HTTP version to the HTTPS version
5. If not found, then this is a new user. Add them with their HTTPS OpenID

Roll out/deployment
1. OP updates discovery documents to provide alias in documents retrieved over SSL but still returns the HTTP OpenID
  -- this allows RP's to run "batch" jobs to upgrade the user
2. After some period of time, the OP stops returning HTTP identifiers and only returns HTTPS identifiers

Questions:
1. How to protect against from rogue OP returning a mapping for a different OP?

Alternate proposal (John Bradley)
* RP's assume that an HTTPS identifier is the same as the HTTP identifier and do an "autoupgrade" if the identifiers are exactly the same except for scheme.

1. OP MUST move to HTTPS OpenIDs permanently
2. OP MUST 301 HTTP requests to HTTPS
3. If the RP has an existing HTTPS identifier for the user, it MUST NOT consider the HTTP version to be the same
4. Upgrading the HTTP OpenID to HTTPS MUST only occur for users that do NOT have an existing HTTPS OpenID identifier

Identity Business Models (T2G)

URL: http://iiw.idcommons.net/Identity_Business_Models

Convener: Lars
Notes-taker(s): Christie Grabyan

Tags for the session - technology discussed/ideas considered:

- Business Propositions for Identity Providers
- Liabilities
- Customer Value Propositions

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

1. **Business Propositions for Identity Providers**

- Discussion around is there a business model for identity by itself (i.e. identity as a service)?
- For a business model to succeed, you don't have to generate a profit, but you do have to generate revenue in order to be sustainable
- How do you monetize the value of providing identity?
- Identity as a service does not seem to provide inherent value. It is identity + some other service that creates the value. What is going to incent the customer/consumer to want to go through the process of allowing increased identity data to be shared, for example? Equifax sells identity to consumers but it is because they provide a value proposition to consumers in the first place.
- It is agreed amongst the group that a single or universal identity is not an option

The following ideas for business propositions for identity providers were discussed with some evaluation on the level of value or the willingness of an entity to pay for that service:

Business Proposition (Finding business opportunities in the world of identity)	**Value** (is of monetary/ business value to a 3rd party)	**Pay** (Would an entity pay for it?)
Access to Customer Base	High	Low / ?
Identity Verification/Trust	High	Possible
OpenID Provider Certification	High	High
Data Sharing (leveraging data associated with one's identity, i.e. shipping address)	High	?
Abandonment	High	
Trustable Assertions	High	High
Payments – Avoid PCI	High	High

Liability Insurance		?
Reputation (for both customer and merchant)		
Convenience – standard UI		
Consulting Services (around providing identity)		
Neutral 3rd party/Broker for identity (i.e. non-profit)		
Cross Promotions		
Privo – Permission-based Marketing		
Traffic Patterns		
Authentication as a Service (Multi-Levels)		
Trustable Attributes		

It was reiterated and discussed that most of these business propositions combine identity *plus* something else (i.e. Payments, Data Sharing). Many of these propositions do not generate direct revenue from identity alone, but they are associated revenue streams that rely on identity. Additionally, some of these propositions are not necessarily identity issues, but they are things to overcome in order to be an identity provider.

Identity providers were discussed and ICAM (Identity Credential Access Management) was mentioned. The US government as a replying party doesn't pay for it, but IdPs do pay for the ability to be an IDP.

One opinion is that credit card brands and e-commerce sites (VISA/MC/Amex, Amazon, Ebay) are IDPs already in existence that cover the majority of issues/people. The question was proposed: How do you solve the naming issue between these identities already in existence? Others conflicted with this view to say that naming standards are not the issue, and that identity is more than just authentication.

Also, what is the business value for Amazon/Facebook, etc to share identities? They already have such strong brands. For example, if a customer wants to buy a book from an independent bookseller who also has a presence on Amazon, they are more likely to purchase through Amazon itself than the seller's own site because of trust in the Amazon brand and the level of assurance it provides consumers.

2. **Liability insurance**

Questions: What happens if your account gets compromised? Who is responsible for that? (i.e. Amazon eats the cost of a merchant or customer account being compromised; they take on the liability)

Discussed Liabilities:

- Compromise
- Brand compromise
- Loss of control

- Cost of compliance (HIPAA, SOX, PCI)
- Identity theft
- Account portability

3. **Customer Value Propositions**

- Fewer Passwords / Ease of Use
- Data Portability
- Trust (buying from Amazon vs a small player) / Trust Broker
- Aggregation of data (i.e. Mint.dom, or aggregate purchase history)
- Social Reputation
- Auto-fill forms
- Social Reputation (i.e. Facebook)
- Financial Reputation (i.e. use Equifax to prove your financial stability to a bank or landlord)
- People Discovery (i.e. LinkedIn)
- Service Discovery
- Personalized Recommendations
- Access Control (to use the employee example of de-provisioning)
- Authentication Convergence
- Personal Ownership of Data
- Pseudonymity
- Outsourcing relationship management (See a supporting paper for this by Bob Blakley at Burton Group at this blog address: http://identityblog.burtongroup.com/bgidps/2009/02/relationship-paper-now-freely-available.html or this download: http://www.burtongroup.com/Guest/Idps/RelationshipLayerWeb.aspx)
- A-to-Z guarantee

Final discussion was a debate over OIX and whether it can/will solve the IdP problems in the industry or not.

Patents, People and Development Pools (T2H)

URL:

Convener: Marc Davis
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Enterprise Signing In OAUTH2 (T2I)

URL: http://iiw.idcommons.net/Enterprise_Signing_in_OAuth

Convener: Brian Eaton
Notes-taker(s): William Mills

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Brian: "I'm really happy with Oaut2, except for the signing... Can we take section 5.3 of the spec and set it on fire?"

Problems seen:

- HMAC covers the situation where people don't want to do SSL, but requires the whole ugly key management thing. Brina proposes (with a long list of others interested) that we need public key signing.

- The signing is not extensible, you can't add additional fields to the signature. This seems to be a problem in the current spec.

- New protocol substrates.

- One signing choice doesn't seem enough. Extensibility seems key here, can we do this with some form of discovery?

Proposed fixes:

- Create a JSON blob we want signed.
  - an example... does not require reconstruction of the string to be signed
  - BUT how do you verify the signed stuff is part of the request?
  - ALSO not great: duplication of data.
- We need key versioning
- Needs key discovery
  - Can use something liek https://<app>.appspot.com/.well-known/oauth or some such.
    - This might not work for folks that are white label, because there isn't a separate URL for all the entities that need to be discovered.
  - There is a google group for the OAuth Key Discovery spec. See Brian's copy of the slides.
  - It's worthwhile to join the OAuth2 mailing list to comment on this.
  - This is a big discussion topic, I don't type that fast.
  - XKMS is again, good reference reading for this.

Discussion:

- How does it work with firewalls -- in and out...
- XMLDSIG is very similar to this, it would be worthwhile to learn from that. Also CMS (Crypto Message Sig).
- Does have the whole PKI problem.
- Can we solve this with SSL? SSL with client certs? Maybe...
    - Much discussion here about how key excahnge and key management should work.
- Comment -- If you want to sign arbitrary parts of an HTTP request then use SAML. you don't really want to duplicate that here.
- Need to make sure we get the key exchange right, if you try to put it in here people will get it wrong.
- KeyID/key discovery.
    - comment: see XKMS for related reading.
    - keyID probably belongs in the envelope and not the payload (from his modified example.

Simple Reputation Feed (T3A)

URL: http://iiw.idcommons.net/Simple_Reputation_Feed

Convener: Tatsuki Sakushima
Notes-taker(s): Tatsuki Sakushima

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Simple "Reputation" Feed
Tatsuki Sakushima
NRI/OASIS OpenReputation TC

Agenda
1. Introducing "Simple Reputation Feed"
2. What is your use case? What do you want the data to be?
3. Issues in the current design?

What is "Reputation"
Reputation:

- is a metric (a score, a rank, a state, a multi-dimensional profile, etc.) associated to an entity (a person, a business, a digital identity, a website, a system, a device, a category of devices, a computing resource, etc.) or to a tuple [entity, attribute(s)] (e.g. [person,skill]) in a particular domain and at a particular moment in time. (Data)

- is a subjective evaluation of the assertion about a subject being true based on factual and/or subjective data about it, and is used as one of the factors for establishing trust on that subject for a specific purpose. (Function)

- helps you make sound judgments in absence of any better information. (Purpose)

Examples of Reputation Data

- FICO credit score
- Nielsen ratings for television programs
- Page views for web sites
- Online movie reviews and ratings
- Dow Jones Industry average
- Amazon Book reviews
- eBay User feedback score
- PageRank, Buzz, Digg, Facebook "Like", etc.

These are all aggregated reputation data examples.
Reputation data takes other forms:

- Assertion: "Harvard is a good law school."
- Prizes and Awards: Nobel Peace Prize.
- Actions: Volunteer for Earth day.

The Goal of Simple Reputation Feed

Why has never Reputation Feed taken off?

- No standard data format for distribution.
- No motivation to share?
    - eBay declined Amazon in late 90's.
    - No Web20. Web may not be so social back then.

Our goal would be:

- To provide a data format in order to open up existing reputation data and to share it across domains.
- To provide a data format like "Atom" for reputation data.

If we can share existing reputation data out there in a standardized way and let developers freely aggregate them, the potential and impact is tremendous.
Feed helps de-compose data and re-compose something new.

Approach

- Reputation is polysemic and polymorphic.
- We need a reasonable scope to describe reputation data.
- Reputation data must be machine-friendly and aggregatable.
- How about Subject, Context, and Score?

Subject:

The evaluated entity(reputee). The entity could be a person, a business, a digital identity, a website, a system, a device, a category of devices, a computing resource, and a tuple [entity, attribute(s)] (e.g. [person,skill]) in a particular domain and at a particular moment in time.

Context:

The definition of reputation data. Context defines who/what(Subject) is evaluated in what context and how(Score).

Score:

The result of evaluation in this context. The score should be somewhat numeric value like a score, a rank, a grade, a rating, a digg/likeit/thumbup. It excludes opinions and reviews(text) because it is hard to aggregate as data.

MODEL

? – unable to copy image from PDF

Examples

Example 1. Simple ORMS Example

```
<Reputation>
<Context>http://bookstore.co.jp/reputation/bookreview</Context>
<Subject>http://bookstore.co.jp/literature/auther/hmurakami/title/1Q84</Subject>
<Score type="http://bookstore.co.jp/reputation/score/products/books/fivestar">4.3</Score>
<Score type="http://bookstore.co.jp/reputation/score/products/books/rank">1</Score>
<Date type="http://bookstore.co.jp/reputation/date/lastupdated">1970-01-01T00:00:00Z</
Date>
<Date type="http://bookstore.co.jp/reputation/date/created">1970-01-01T00:00:00Z</Date>
</Reputation>
```

Example 2. ORMS ReputationBundle Example

```
<ReputationBundle>
<Reputation id="http://bookstore.co.jp/reputation/document/MIICyjCCAjOgAwIBAg"
type="parent">
<Context>http://bookstore.co.jp/reputation/score/products/books</Context>
<Subject>http://bookstore.co.jp/literature/auther/hmurakami/title/1Q84</Subject>
<Score type="http://bookstore.co.jp/reputation/score/products/books/fivestar">4.3</Score>
<Date type="http://bookstore.co.jp/reputation/date/lastupdated">1970-01-01T00:00:00Z</
Date>
<Date type="http://bookstore.co.jp/reputation/date/created">1970-01-01T00:00:00Z</Date>
</Reputation>
<Reputation id="http://bookstore.co.jp/reputation/document/MRIwEAYDVQQIEwlXa"
type="child">
<Subject>http://bookstore.co.jp/literature/auther/hmurakami/title/1Q84</Subject>
---whatever sub reputation data here---
</Reputation>
<Reputation id="http://bookstore.co.jp/reputation/document/F1VuaXZlcnNpdHkgbX"
type="child"/>
---only link to sub reputation document---
</ReputationBundle>
```

Design Principles

1. It only defines a "framework". ("Relationship" among elements.)
    - The data provider must define the context and detail of what the data means.
    - It is a markup language(XML) for writing reputation data.
    - No standardized score format, normalization and representation. (The data provider should define those in a "profile".)

2. The transport layer is out of scope.
    - However, it assumes REST API(HTTP GET) to be used for retrieving this document (data).

3. It keeps the structure simple.
    - No more than 2 layers in hierarchy. (3 layers with <ReputationBundle>)
    - Only 6 elements are defined. (excluding ds:Signature)
      <Reputation>, <Context>, <Subject>, <Score>, <ReputationBundle>, <Date>
    - It avoids defining a group element if possible. (To make the structure flat.)
    - It avoids defining unnecessary schema (elements, attributes).
    - It avoids defining unnecessary data types. (e.g. id values since URI is identifier.)

4. It supports and covers 80% of use cases.
    - It avoids adding function for specific usage. (Can the usage apply to other use cases?)
    - It avoids supporting all needs out there.

5. People intuit how to use it.
    - The semantics of elements and attributes make sense to everyone.

Action Items for ORMS TC

1. To consider element naming alignment with terminology in
"Building Web Reputation Systems".

2. To consider "JSON" representation from the beginning (for
expecting more adoption.)

3. To consider non-numeric reputation data to be presented.
(e.g. text for book review)

Lawyers & Accountants – How To (T3C)

URL: http://iiw.idcommons.net/Lawyers_and_Accountants

Convener: Stu Leudan
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

The Right Question – Making Privacy Policies User Centric vs Data Centric (T3D)

URL: http://iiw.idcommons.net/The_Right_Question

Convener: Aaron Titus
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

OIX (T3E)

URL: http://iiw.idcommons.net/OIX

Convener: Scott David, Drummond Reed, Don Thibeau
Notes-taker(s): Christie Grabyan

Tags for the session - technology discussed/ideas considered:

> Catch up on what has happened in the last 6 months, plus review of what OIX is today.

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Defined: OiX is a community-developed solution to the problem of how open identity credential (Open ID, Info Cards) can be trusted online.

Up-to-date:

6 months ago the discussion was primarily around terminology and getting everyone on the same page.

Recent developments include:

(March 2010)

- OIX launched at RSA
- Approved as US ICAM TFP
- First 3 ICAM IdPs certified
- Open Identity Trust Framework Model paper published

(May 2010)

- Working groups commence
- Expanded membership doc approved

Review of the OITFP Model: Under the OITF Model, the Trust Framework Provider (TFP) communicates with the Identity Service Provider (IdP), the Relying Party (RP) and the Assessor. The IdPs and the RPs interact directly with the Users. The IdP provide levels of assurance to users and assessors, and the RPs provide levels of protection to users and assessors.

The feedback from the industry and the priority was to make OIX: simple, lightweight and extensible. Deliberately designed for global scalability, with enough room for policymakers and other trust framework providers to enter.

Data protection notions are already well-defined, therefore it is anticipated that there will be objectively testable levels of protection that could be defined, tested, and assessed. For example, NIST levels can be leveraged as a framework.

The role of ICAM has bridged both technical and policy requirements. ICAM is Identity Credential Access Management: a committee of committees in the US government with a co-chair from the DoD (Dept of Defense).

In the UK, there are also notions of registration authority, credential providers, and identity providers that all fit into OIX's sense of "Identity Service Provider"

Question was posed as to who accredits the assessor? Accreditation could be provided by the TFP. Pr, a role called a Special Assessor could be designated by the TFP (maybe it will become a government agency, or some of the Big5 firms, etc).

In the health sector, there is a problem with the identity side of the equation, but there are also problems with the RP side. The challenge is does that RP qualify to offer services in the health sector (or other sector)?

Question around Liability: Does the TFP provide direct indemnification? Are they are rating agency or a guarantor? Answer: The liability issue is being explored right now, including where are the balance of duties, what kind of contractual elements need to be put in place, if legislation is required, etc. There is a desire to have an industry-lead discussion around liability, rather than wait for the government to tackle it. The intention of lightweight assessor responsibilities in the first phase is a placeholder to allow for working groups and other trust frameworks to chime in and provide more context so that decisions can be made to further define the responsibility of assessors.

Questions: Who accredits TFPs? Answer: The policymakers themselves.

TFPs will multiply. It is not designed to produce a TFP monopoly.

Other trust frameworks are coming…..

- Line Information Database (LIDB) - To safeguard access to telco subscriber data
- PBS Public Media – To connect public TV stations, users, and sites
- XAuth – To simplify movement between social sites
- PDX (Personal Data Exchange) – To support individual data on their terms

Question: What is the sustainable model for these trust providers, particularly for OIX which is solely in the trust business, and doesn't have other revenue streams to rely on? Answer: With OIX, the business model is based on membership fees (by assessors, RPs, etc). Longer term, the goal is to support the cost of maintaining the listing cost.

The listing service (meta-federation) needs to be designed before further decisions can be made about how to operate and maintain it.

Credit card operating rules model is already established in this space.

There will be a session on Wednesday to discuss the PCI trust model and terminology.

There are OIX TF Working Groups to join to further the discussion.

THERE IS ALSO A PP Slide Deck FOR THIS SESSION

UX of no Logout in Single Sign On (T3F)

URL: http://iiw.idcommons.net/UX_w/no_logout...single_sign_out

Convener: Judith Bush
Notes-taker(s): Judith Bush

Tags for the session - technology discussed/ideas considered:

SSO, Single sign out, logout, close the browser

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Ideally a user will close their browser to securely terminate all SSO sessions. Single Logout has many UX issues that make actually implementing problematic. However, users who may be using public or shared computers need to securely terminate but may not know that closing the browser is the best way: they expect a logout.

Stanford has gone over ten years in their heterogeneous application environment and their Kerberos/Shib SSO environment. Student will come out of these environments trained to close the browser (not just the tab ot the window)> How to train others?

Offer a "logout" button that redirects the user back to a specific page of their IDP. SAML 2.0 (or shib) may have a page for this use specified.

Alan Karp suggests that close tab as well as "logout" would send user back, continuing the education action, and suggests that unguessable URLs be used for personal machines.

Steve Williams thinks single sign on (one entrance of credentials for a day) is a bug.

Sharing URLs with the Open Stack (T3G)

URL: http://iiw.idcommons.net/URL-Sharing_Using_the_OExchange_Protocol_Stack

Convener: Will Meyer
Notes-taker(s): Charlie Reverte

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

trying to enable sharing to long tail services from content publishers. few goals, standardize way to send urls (exchange) to any site on the web, discover new services and allow users to personalize services that they see. services won't always be known at design time, use a late binding so new services can be discovered.

exchange: the service has an "offer" endpoint, standardize the params for passing an url, title, etc.

discovery: xrd document that describes the service, the name, "share" verb (send to, tweet this, translate). the service xrd is linked to from the site's host meta and via link tags in the page head (similar to rss).

personalization: publisher can offer you sharing options to the set of sites that you actually use including long tail sites they wouldn't normally link to. niche communities often have higher engagement and drive more traffic than general purpose social networks.

multiple options for persisting a user's service preferences, xauth, cookie, browser local storage, webfinger. webfinger is preferable as it persists across machines and allows others to discover services you use (and interact with you on those networks).

goals are to first codify how sharing is done today, later try to add options for new flows like popups, headless sharing, etc.

questions about splitting the personalization, personal discovery part into a separate spec; the exchange and service discovery parts are ready to go but webfinger etc aren't deployed yet.

there are roles for the browser here, to help discover new services and store preferences for services that you don't want to be public

Secure Web Auth: Against Phishing, IETF Draft, Implem., Next-gen ideas, Demo! (T3L)

URL: http://iiw.idcommons.net/Secure_Web_Auth

Convener: Yutaka OIWA
Notes-taker(s): Yutaka OIWA, Tatsuya HAYASHI

Tags for the session - technology discussed/ideas considered:

Secure Web Auth

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Presentation is available at:
https://staff.aist.go.jp/y.oiwa/publications/2010-IIW10-MutualAuth-P.pdf

Questions from the floor: (identities of questioners wanted!)

- Q(___) How about the header format?
  - A. The protocol uses a format based on RFC 2617, compatible with existing protocols.
- Q(___) Scalability issues
  - A. The protocol supports a domain-based single-sign-on (e.g. *.yahoo.com). Cross-domain authentication might be useful with integration to existing authentication mechanisms (e.g. SAML, OpenID etc.)
- Q(___) Compatibility with existing applications and their migration
  - A. It requires a small change to existing applications.
    It includes several extensions to existing HTTP auth mechanisms, which enables migration of current (form-authentication-based) applications to our scheme without changing the whole design of website (current Basic/Digest auth has difficulty on user-experience compatibility). For example, it includes support for guest-user support (optional authentication), server-initiated forced logout, redirection of unauthenticated users to dedicated log-in pages, and others.
- Q(___) Compatibility with existing browsers
  - A. Browsers must also be extended. We already implemented it on Mozilla codebase and see how much modification needed.
- Q(___) How to migrate from or co-exist with existing auth, such as Form auth or Basic?
  - A. Application frameworks can support parallel support with Form-based auth (because existing browsers simply ignore our WWW-Authenticate headers). Parallel support with Basic auth may need some additional functionality for negotiations (HTTP spec supports two or more WWW-Authenticate headers at the same time, but it does not work well on existing code).

- Q(___) Standardization issues and schedules
  - A. We are working on IETF to making this a WG issue. Originally handled in HTTPBIS WG, now under OAuth WG temporarily. We maybe need a new WG for this and related issues. We want it within around 1 year.
- Q(___) Real-world experiences, deployment and field-tests.
  - A. We've done a field test on "Yahoo! Japan Auction Trial" website. We've got a feedback on deployability and compatibility with existing web applications. For scalability and user-experience we may need more testing in a near future.

The Design of and Case for KRL (T3O)

URL:

Convener: Phil Windley
Notes-taker(s): Phil Windley

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

The location metaphor that is often used to describe the Web lead to siloed Web sites. But people what to achieve a purpose, not go somewhere when they are online.

Building Internet apps that respond to complex event scenarios and use multiple APIs with diverse authorization schemes places an unscalable burden on developers.

Architectures and APIs help, but do not go far enough. When computer science has faced similar problems we have built notations--programming languages. Programming languages can wrap the architecture and APIs in a syntactic and semantic layer that provides significant mental leverage for the developer via the abstraction.

KRL is a rule-based language for describing reactive systems. At KRL's core is a complex event expression language. KRL has a full featured expression language that includes primitives for interacting with OAuth authenticated data sources. Rule actions depend on the target domain. For the web actions can modify the DOM.

Using KRL, developers can easily create apps that combine and remix data from multiple Web sites and superimpose the results onto any other.

Research Report on Information Sharing (T4A)

URL:

Convener: Ian Henderson
Notes-taker(s): Ian Henderson

Tags for the session - technology discussed/ideas considered:

User Research, VRM, Volunteered Personal Information

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Several pieces of research have been done around the issues of personal information management and use.

Materials will be published and progressed via Information Sharing workshop (will email details when uploaded)

OAuth for Native Apps (T4B)

URL: http://iiw.idcommons.net/OAuth_2_for_Native_Apps

Convener: Marcus Scurtescu
Notes-taker(s): Eric Sachs

Notes:

OAUTH2FLOWS
-web server
-useragent
-device
-username & password

NATIVE APP TYPES THAT CAN EMBED OR LAUNCH A BROWSER
- GUI app
- Command Line app
- Phone app

LIMITS OF OAUTH2 USERAGENT FLOW
- it works okay with an embedded browser
- but does not work well if the browser is launched by the app
- user agent does not get refresh token, so app's access to API expires
- WebServer OAuth2 flow is closer for native app needs, however it requires registration and that doesn't make sense for native apps that can't keep secrets
-Also no callbackURL for nativeapps, so may need the "oob" value back from OAuth1

TECHNIQUES
- copy&paste
  fallback, but would be nice to work better

- embedded browser
  depends on how embedded browser handles cookies and the user experience
  if the service provider has a two-factor auth process when cookies are not present, like a bank, then it really hurts user experience

- custom scheme
 OS dependent, works somewhat on some phones, but hard on Windows especially when there are multiple browsers the user might use

- local web server
 Takes more resources on the machine
 Firewall software can cause problems

- monitor cookies

Requires using hacker techniques to peek into cookie jar

- monitor title
 Some OS variance, but works well on Windows
 More variance in ability for app to bring itself back to the foreground

- browser extension
 Too much variance

- use a web-service to request the token
 but still requires launching a browser, and still have same problem for app to know when to bring itself to the foreground

- app can keep polling authorization server to see if token is valid, but creates a lot of load and potential DOS alerts on provider

IMPLENTATION OPTIONS
- library
- service
 Preferred option like Android Account Manager, but this does not exist on other platforms
- command line tool
- Android use a registered custom scheme, but can't auto-close browser

UMA and Claims (T4C)

URL: http://iiw.idcommons.net/User_Managed_Access

Convener: Tom Holodnik
Notes-taker(s): Eve Maler, Tom Holodnik

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Notes from Eve Maler:

See Tom's slides: http://kantarainitiative.org/confluence/download/attachments/37751312/UMA+Claims+-+IIWX.pdf

UMA has two really great ideas in it: dynamicism and claims. Dynamicism is by its nature inclusive, and claims are by their nature exclusive.

Claims are going to have to support both self-asserted data and third-party-asserted data. And there are even ways of authenticating a "walk-up" requesting party that are so lightweight that they feel like self-asserted data. E.g., if in the process of engaging with a future requesting party in person (your dentist) you give them a tear-off paper with a unique temporary password that they need to present when seeking calendar access, you've authenticated them pretty strongly and only need to correlate "the same party" in future.

If you want to share dental records on a more formally authenticated basis, other things might have to happen.

UMA needs to have a unified way of "being" a requester endpoint, even if it has different flows for how they are interacted with. We think we have that now.

Should a specific person at a company be given access to Alice's stuff, or should a role at the company (or just generically "the company") be given access? The former is brittle.

What if you want to grant any plumber who has a good certification rating access to my plumbing service record? There will be company and certifier assertions involved.

Claim format definitions clearly need to have a generic/horizontal core set, and likely we would need domain-specific plugins for specialized policies and claims.

+++++++

Notes from Tom Holodnik:

Summary: So far, only simple forms of UMA claims are currently defined. Under our current understanding, we can either have support for broad dynamically configured access management circles of trust with relatively low requirements for trust, or we can have strong assurances of identity and legal certainty but with rigidly defined circles of trust (e.g., JSONifying SAML). That is, we can either have broad and shallow circles of trust, or narrow and deep, but nothing in between.

Currently defined claims models are defined here: http://kantarainitiative.org/confluence/display/uma/Claims+2.0

and here: http://kantarainitiative.org/confluence/display/uma/Simple+Access+Authorization+Claims

The kinds of claims we support now are self-asserted claims such as "Are you older than 18?" and "Will you comply to these licensing terms?" Any claims that demand that someone assert an identity with any level of assurance are yet to be defined.

We got bogged down in a conversation about ontologies and grammar for claims. In the writers opinion, this missed the point of the session- this was to understand the goals for claims and less about how claims are expressed.

Regardless of the way it's expressed, it needs to be decided whether we're simply translating SAML assertions into JSON within a closed Circle of Trust, or whether we're attempting to build an internet-scale system for provisioning and processing claims of identity and other attributes.

That said, the user who establishes a relationship with an UMA Access Manager shouldn't have the expectation that the AM will serve as a directory of service providers. The idea is that the AM has only enough infrastructure to support validating claims- it doesn't promote only those requesters that it has a relationship with. An UMA AM is not a VRM.

Client-Side Opt-In Cross-Site Data Sharing: Can We Build XPrefs? (T4E)

URL:

Convener: Dan, Mike Rgavan
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

9195993141 or @ajbraun Who Am I? Telco vs the Net (T4F)

URL:

Convener: ?
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

WebBIZcard: Using URL Syntax For Naming & Data Exchange (T4G)

URL

Convener: chen.WebBIScard.com
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

SAML Profiles for OAuth (T4H)

URL: http://iiw.idcommons.net/SAML_Profiles_for_OAuth

Convener: Chuck Mortimore
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Overview

At IIW 2010a, a small group gathered to discuss use-cases we've been seeing for using OAuth 2 together with SAML. The following attempts to capture the essence of that discussion, as well as additional conversations I've had with people throughout the industry. In short, a number of us are seeing real use-cases, and real customer demand; there is even some prior art to learn from in early WRAP deployments. Hopefully this can act as a catalyst for formalized profiling within the community as to avoid fragmentation.

Exchanging a SAML Assertion for an OAuth Token

While OAuth 2 provides a basic Assertion Flow, it's real intent is to provide a pattern for message exchange; the flow itself is somewhat useless without further profiling, or at least a private understanding between the various parties in the OAuth exchange. It's clear that a profile of the assertion flow for use with SAML is required if we are to hope for interop and vendor software to emerge. First a use-case.

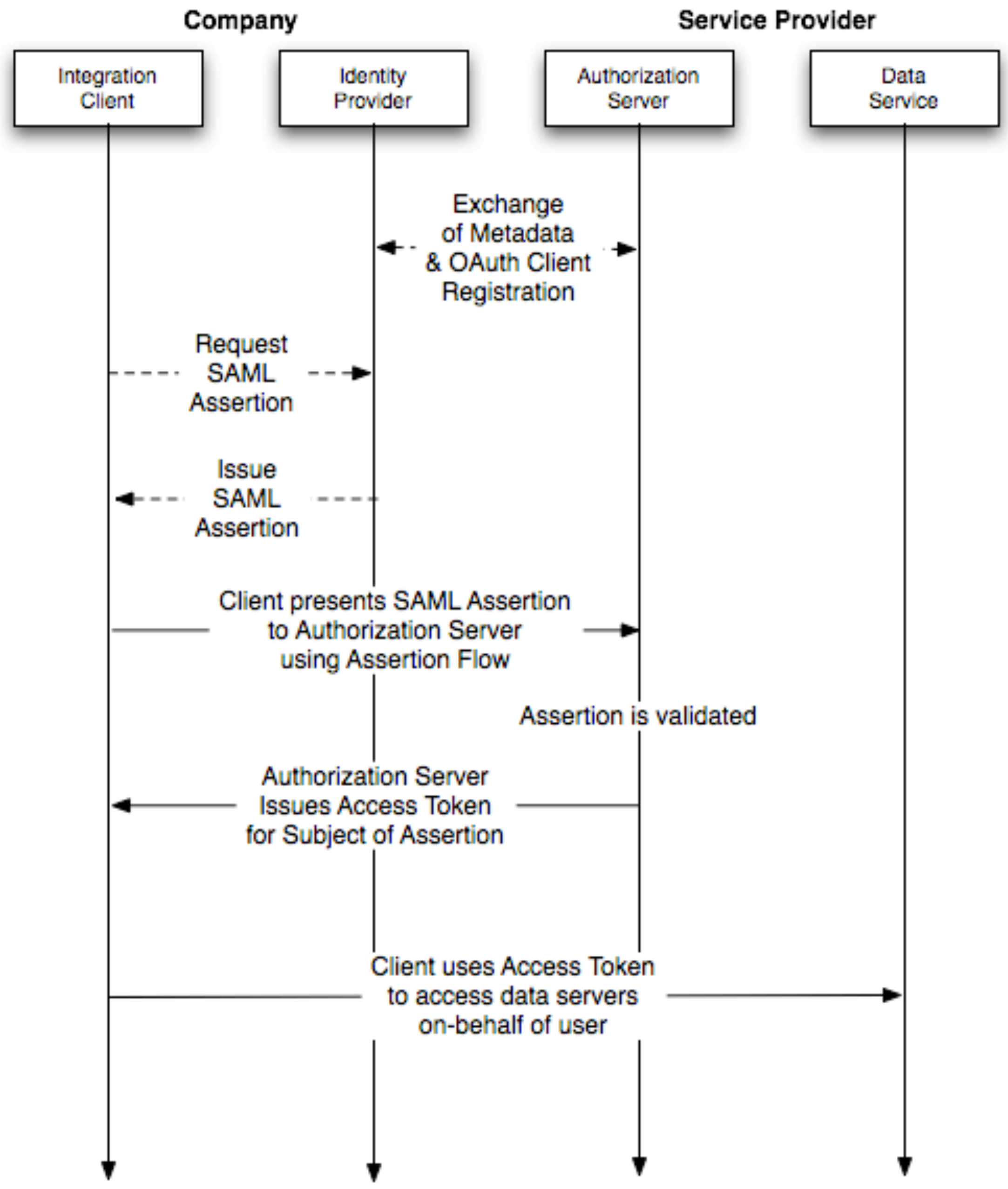
Use-Case:

A company has an existing relationship with cloud service provider. In order to provide improved control over credentials used to access the service, as well as a simplified user experience, the company and service provider have established web single sign-on using the Web SSO Profile of SAML 2. With the company acting as an IDP, their users are easily provided access to the service without exposing their corporate password, or managing new credentials.
The company now wishes to establish integration with data services that act on the user's behalf; for instance syncing calendar and contact information. While a traditional 3-legged flow of OAuth 2 could be used to establish individual access or refresh tokens, the company wishes to perform this synchronization as a back-office process on a nightly basis. Given an existing

trust relationship has been established by the company it is neither required, nor desired to have all of their users perform the OAuth dance to establish these tokens; rather the company would like to use a signed SAML assertion to interact with the API. (Note this is conceptually similar to 2-legged OAuth: [1]) In turn, the service provider would prefer the simplicity and consistency of simple OAuth 2 bearer tokens be used for access to all their services.

Using the SAML profile of the OAuth 2 Assertion flow, the company is able to rapidly write a integration client. This client fetches a SAML assertion from their IDP software, and presents it to the service providers Authorization Server. Having previously establish a trust relationship through the exchange of SAML metadata, the Authorization server validates the signature and contents of the assertion, and issues a access_token on behalf of the subject of the assertion. The company is than able to impersonate the user with the service providers data services, acting on their behalf. The company syncs the user's calendar and contacts, and disposes of the access token. Subsequent access uses a new SAML assertion, rather an refresh_tokens as to maintain the existing trust relationship (refresh tokens would essentially bootstrapo a new trust relationship)

Presented as swim lanes, the exchange might look like this:


Company
Service Provider
Integration Client
Identity Provider
Authorization Server
Data Service
Exchange of Metadata & OAuth Client Registration
Request SAML Assertion
Issue SAML Assertion
Client presents SAML Assertion to Authorization Server using Assertion Flow
Assertion is validated
Authorization Server Issues Access Token for Subject of Assertion
Client uses Access Token to access data servers on-behalf of user

Design Goals for profiling the Assertion Flow

1) Attempt to match the simplicity of OAuth 2: While SAML can be complex when you view the specification as a whole, focus must be placed on profiling a thin and simple path. Initial emphasis is placed on bearer-tokens.

2) Piggy back on the relative success of Web SSO: The Web SSO Profiles are the most broadly deployed and well understood aspect of SAML, and of these, the HTTP POST Binding is arguably the most popular, as well as the closest in nature to the assertion flow. Modeling the

profile after these proven deployments likely provides the shortest path to working code, leverage of existing software, and existing methods of trust establishment. While this last subject requires improvement, this is a broader issue that this profile would benefit from.

Proposal

The SAML Assertion Flow Profile builds upon the Assertion Flow by specifying the exact format and assertion values to be used with SAML 2.0. Specifically, it is intended to provide interoperability with the SAML 2.0 Web Browser SSO Profile [urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser] and the SAML HTTP POST Binding [urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST] by allowing the authorization server to process SAML assertions with the same format and characteristics as those used in SAML Web SSO.

format: “urn:oasis:names:tc:SAML:2.0:profiles:oauth2:assertion”

assertion: The value of the assertion parameter MUST contain a valid SAML Assertion. The SAML Assertion MAY be enclosed in a valid SAML Response. If enclosed in a SAML Response, the Assertion MAY inherit it’s signature from the Response per SAML Core 5.3. The value of the parameter MUST be form-encoded by applying the base-64 encoding rules to the XML representation of the message. The base64-encoded value MAY be line-wrapped at a reasonable length in accordance with common practice.

As with SAML web SSO, the assertions used in this profile are one time use assertions. The mechanism by which the client obtains the assertion is out of scope for this profile.

All protocol, assertion format, and processing rules that are defined by the SAML 2.0 Web Browser SSO Profile, and scoped by the SAML HTTP POST Binding MUST be adhered to in this profile, with the following clarifications and exceptions:

- The authorization server MUST accept unsolicited SAML <Response> messages. AuthnRequest messages MAY be used, but their usage is left unspecified by this profile
- RelayState data is omitted from this Profile
- The form-encoding specified by the POST binding is superseded by the assertion_request message defined in section 3.6.2

For example, the client makes the following HTTPS request (line breaks are for display purposes only):

```
    POST /authorize HTTP/1.1
    Host: server.example.com

type=assertion_request&client_id=s6BhdRkqt3&format=uurn:oasis:names:tc
```

```
:SAML:2.0:profiles:oauth2:assertion&assertion=PHNhbWxwOl...[ommited
for brevity]...ZT4%3D
```

The response is a standard OAuth 2 response

Protocol for Clients Requesting SAML Assertions

The previous section covered how a client would use a SAML assertion over the OAuth 2 assertion flow, and the related processing rules. However, it did not specify protocol for obtaining an assertion. There are a variety of mechanisms by which an assertion may be obtained by the client

Self Issued: The client may be intelligent enough to issue it's own SAML assertions directly. It should be noted that this is only appropriate for clients where the private key may be appropriately secured.

Active Clients: This client is intelligent, and knows how to interact with it's IDP in order to authenticate and obtain an assertion for the subject using the client.

- WS-Trust - an enterprise may want to use this internally
- OAuth2 - an OAuth 2 flow could be used between the client and the IDP - basically the same as WS-Trust but loosely "restful"
- The company IdP server acts as a intermediary (of sorts) between the client and the cloud authorization server. Client requests access token from IdP SAML Gateway and Authorization Server using something like Username and Password Flow or Client Credentials Flow. IdP validates the request and generates a SAML assertion that it sends to the cloud Authorization Server using an Assertion Flow. The access token is passed back all the way to the client.
- Some profile of SAML similar to ECP?

NEED SUGGESTIONS IN THIS SECTION

Passing OAuth Responses over SAML

In addition to the classic STS pattern that the assertion flow defines, there is desire to provide optimized exchanges of OAuth tokens, piggybacking on SAML as a transport and pre-established trust mechanism. This is conceptually similar to the OpenID + OAuth flow defined here: [http://step2.googlecode.com/svn/spec/openid_oauth_extension/latest/openid_oauth_extension.html]

Use Case: "IDP knows the SP is going to want a token"

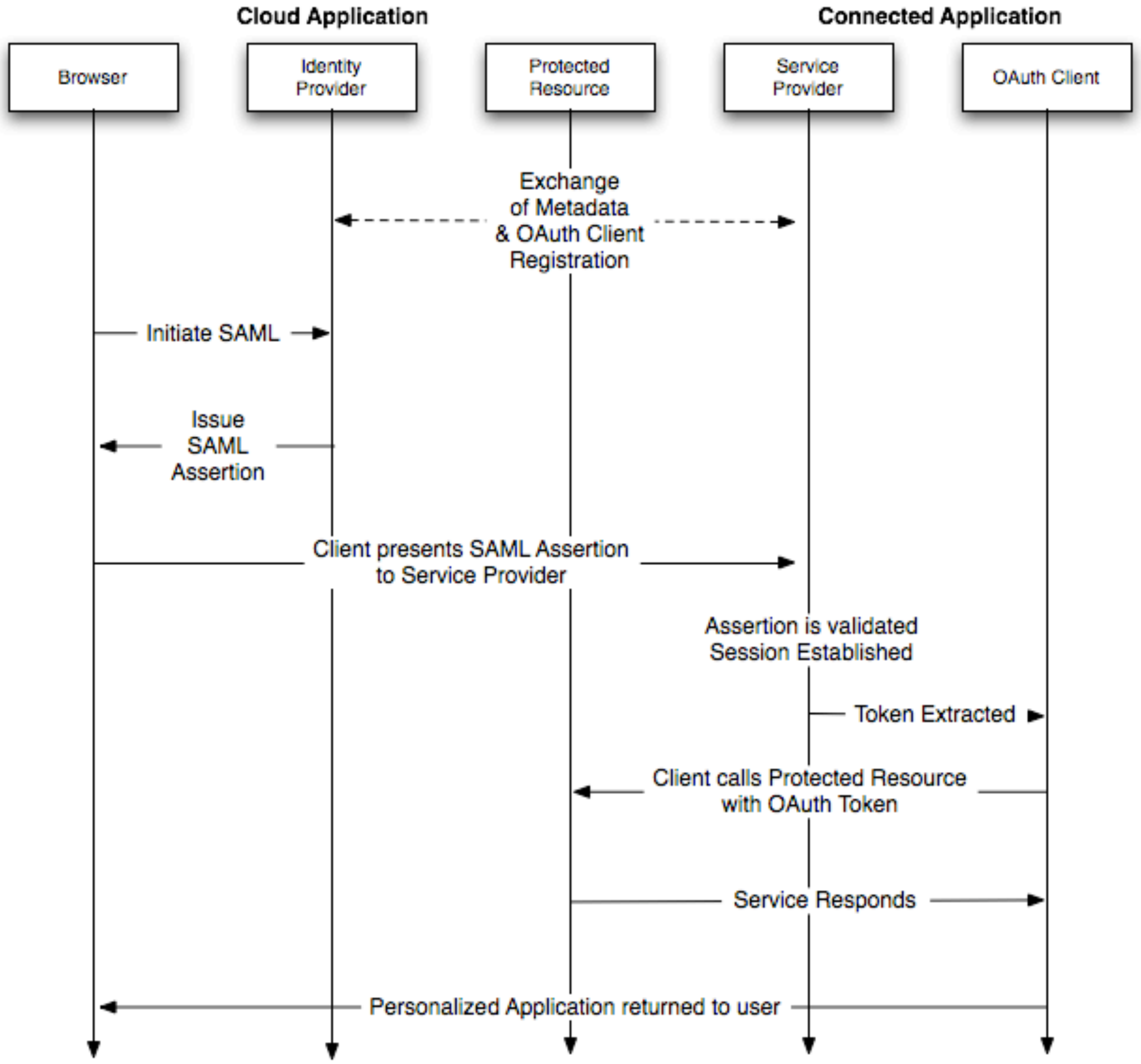
A Cloud Application is assembling a application market place, where related business applications (Connected Application) will be able to integrate with the main application through single sign-on and data services. (Note existing examples of this include Google Apps Marketplace, Intuit Marketplace, Salesforce AppExchange, etc) As part of this marketplace, the Cloud Application will be acting as an SAML IDP to the Connected Application acting as a SAML Service Provider, and the connected application will be accessing data services at the IDP using OAuth. Technically you have an IDP (SAML) which is also a Authorization Server(OAuth2), and a Service Provider (SAML) which is also a Client (OAuth2)

When a developer launches a new Connected Application in the marketplace, they've established the appropriate SAML metadata to allow for single sign-on between the IDP and the SP. When an admin installs the Connected Application into the their tenant in the multi-tenant Cloud Application, they explicitly grant permission for the Connected Application to access protected resources. This delegates authority for all users to the application.

When a user logs into the Cloud Application, they see a list of Connected Applications they have access to. The user clicks on a link from this list, which kicks of a SAML exchange between the two parties. When a SAML Response is sent to the Connected Application, it includes a OAuth Response as part of an Attribute Assertion. This response includes an access token (and optional response token) is valid for the subject of the assertion, and for the resources which the admin granted permission for.

The Connected Application receives the SAML Assertion, processes it, establishes session for the user, and uses the oauth token to call protected resources in order to personalize the experience for the user. No subsequent user level delegation or oauth dance is required.

Presented as swim lanes, the exchange might look like this:


Cloud Application
Connected Application
Browser
Identity Provider
Protected Resource
Service Provider
OAuth Client
Exchange of Metadata & OAuth Client Registration
Initiate SAML
Issue SAML Assertion
Client presents SAML Assertion to Service Provider
Assertion is validated Session Established
Token Extracted
Client calls Protected Resource with OAuth Token
Service Responds
Personalized Application returned to user

Proposal

It is known by the IDP/Authorization Server that a SP/client requires an access_token in order to act on-behalf of a user. In addition the IDP may choose to issue a refresh_token. The manner by which this is known, and the trust between the two is out of scope.

When the IDP generates the SAML assertion, it includes an Attribute statement, with an attribute conforming to the xml format of the OAuth 2 specification.

Option 1:

```
          <saml:AttributeStatement>
               <saml:Attribute Name="access_token" xmlns:oauth="TBD
based on oauth spec">
                    <saml:AttributeValue
xsi:type="xs:string">SlAV32hkKG</saml:AttributeValue>
```

```
                 </saml:Attribute>
                 <saml:Attribute Name="refresh_token" xmlns:oauth="TBD
based on oauth spec">
                      <saml:AttributeValue
xsi:type="xs:string">8xLOxBtZp8</saml:AttributeValue>
                 </saml:Attribute>
           </saml:AttributeStatement>
```

Option 2:

```
           <saml:AttributeStatement>
                 <saml:Attribute Name="oauth_response" xmlns:oauth="TBD
based on oauth spec">
                 <saml:AttributeValue>
                      <oauth:OAuth>
                           <oauth:access_token>SlAV32hkKG</
access_token>
                           <oauth:refresh_token>8xLOxBtZp8</
refresh_token>
                      </oauth:OAuth>
                 </saml:AttributeValue>
                 </saml:Attribute>
           </saml:AttributeStatement>
```

Requesting OAuth Tokens via SAML

There is also some need for the reverse of the previous use case - the IDP is an OAuth Client, and needs to access services on the user's behalf.

Use Case: “I want to log the user in, and access protected resources at the Service Provider, without requiring their password”

NOTE: This use-case is essentially the same as the first use-case, but rather than using the Assertion Flow, it is simply an optimization of the existing 3 legged OAuth Flow.

A company has an existing relationship with cloud service provider. In order to provide improved control over credentials used to access the service, as well as a simplified user experience, the company and service provider have established web single sign-on using the Web SSO Profile of SAML 2. With the company acting as an IDP, their users are easily provided access to the service without exposing their corporate password, or managing new credentials.

The company now wishes to establish integration with data services that act on the user's behalf; for instance syncing calendar and contact information. In order to do this, they will negotiate access_tokens and refresh_tokens individually for each user that they send to the service provider. Since the user's login to the service provider via SAML, the flow to both authenticate the user and issue OAuth tokens is quite complex - if the OAuth redirect is done first, the user must be authenticated, so they are sent back to the IDP. If the user authenticates first, the SP needs to know to send the user back to the IDP to start the OAuth dance. The company and service provider would like an optimized flow for both logging the user into the service provider and accessing the OAuth token.

Proposal

In order to do this:

- A SAML Response is sent over Web SSO Profile to the Service Provider. It includes a Relay State of an oauth authorization request
- The SAML is processed, session is established, and the client is redirected to authorization URL included in the relay state.
- Since the Session is already established, the user only needs to grant access to the service
- User is returned to IDP in a regular OAuth flow
- The IDP gets it's access_token/refresh_token and sends the user on their way to the service

```
          <form method="post" action="https://sp.example.com/SAML2/
SSO/POST" ...>
                <input type="hidden" name="SAMLResponse" value="Base64
encoded response" />
                <input type="hidden" name="RelayState" value="https://
sp.example.com/authorize?
type=web_server&client_id=s6BhdRkqt3&redirect_uri=https%3A%2F%2Fidp
%2Ecom%2Fcb" />
                <input type="submit" value="Submit" />
          </form>
```

This basically works already with existing software

What's next

Who should be involved? If you've made it this far, than perhaps you.

What are the right profiles? I've seen use-cases for these. What others exist?

Where is the right place for this work? IETF? OASIS?

Separating: Authenticate Credential Management, Attribute Management and ID Management (T4I)

URL: http://iiw.idcommons.net/Separating:_ID,_Credential,_and_Attribute_Management

Convener: Jim Fenton, Cisco Systems
Notes-taker(s): Jari Koivisto, Cisco Systems

Tags for the session - technology discussed/ideas considered:

Identity system, attribute, relying party, identity provider, authentication, assertion

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Short presentation and discussion

Introduction

Terminology

A basic identity system

- e-commerce site
- IdP
- User

Elements of identity management

- Authentication (establishing who the Subject is)
- Credential managemen (prove to relying parties who the Subject is)
- Attribute management (provide information about the subject)

User trust

- User trust in their IdP is fundamental
- An ecosystem of IdP is required

Authentication

Authentication methods

- Methods useful for user authentication are situation-specific
- Problem: Many existing ID systems are bound tightly to IdP

Authentication Strength

- Relying party knows how good the authentication should be

Authentication endpoint

Security opportunities

- Users that authenticate frequently at a given service are more likely to detect anomalies
- IdP providers can detect anomalous user behavior
- Similar to detection of fraudulent credit card transactions

Credential management

Credential management: functions

- Act as a "key cabinet" for the user
- Support directed ID
- Enforce secure use of credentials

Directed identity

Security and availability issues

- Security
- High value target
- Availability
- Failure of an ID manager may have severe impact on its Subjects

Attribute management

Distributed attributes

- Self-asserted attributes have limited utility
- Authoritative sources for different attributes come from different places
- ID system has a role in locating trustable sources of attributes
- Attributes delivered as signed assertions

Attribute distribution: Example

- Example of Alice buying wine online
- Alice
- Wine merchant
- IdP
- DMV or Healthcare provider

Attribute trust

- Federation: Prearranged trust relationsships
- Accreditation: indirect federation
- Financial institutions, schools

Identity provider trust

- IdP has fiduciary responsibility

- To the Subject:
- Must use credentials only for the proper Subject
- To RPs:
- Must associate attribute request and responses reliably
- IdP may coincidentally funtion as an attribute provider

Observations

- Scaling is critical
- etc.

Discussion:

- RPs have different views what is identity
- Identity vs. Identity information
- Different IdPs give different sets of attributes
- What is the question that RP needs to ask (over 18/21 vs. birth day)
- Attributes of the Subject, e.g. Subjects attibute is: Subject is Cisco employee
- User -> RP -> User -> IdP -> User -> RP
- Yes User is >18
- Discussion about: Is it IdP who user trusts to get the attributes, or should User be the one who makes the decision
- Discussion about the assertions, e.g. State assertion does not mean that it is necessary better than self assertion. Address as an example.
- Federated trust frameworks, level of assurance (1, 2, 3, 4)

Story Cubing and Synergies: A Participatory Workshop About Labeling and Crating Group Identity (T4J)

URL:

Convener: Stacey Pitsillides
Notes-taker(s): Stacey Pitsillides

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Story Cubes and Synergies was a participatory workshop looking at how people label themselves and their practices (in a group) this workshop focused on the connections between peoples, the labeling (identity(s) they give themselves and their practices with the aim of 'literally' building a real-time alternating group identity model.

As it was a participatory workshop, there were no notes, however you can watch the workshop in action at: http://IDcoach.blip.tv/file/3662338

OpenID/AB: For Mobile etc. 2 Attribute Sharing on OAuth 2.0 (T4O)

URL: http://iiw.idcommons.net/OpenID-Artifact_Binding

Convener: =Nat, Breno de Medeiros, John Bradley, Allen Tom
Notes-taker(s):

Notes:

* AB designs for scalable and stateless. It works with mobile phones.
* With AB, OpenID can support up to NIST SP800-63(rev1) L2 - L4 because the assertions are sent in the direct communication channel between OP and RP.
* Asymmetric key signing and encryption will protect the threat defined in L3 - L4.
* RP can choose 2 types of the request mode:

1. Push: Encoded request messsage sent to OP (POST)
2. Pull: Prepare RPF(JSON) msg and let know OP only the URL to the msg
* The Assertion is also in JSON instead of key-value form encoding in 2.0.
* OP implementation in PHP is now around 400 lines of code! RP is 200 including even HTML part.
* For digital signing, "Magic Signature" is used. (to get LoA 2 - 3).
* Encryption:

1. Symmetric key encryption for encrypting "Artifact".
2. Asymmetric key encryption for encrypting "Assertion".
* URL for RPF can be published in XRDS.
* RPF can be cached in OP until updated.
* The "Holder of Key" parameter in the assertion for storing user's cert used for PKI based authentication. (In order to meet LoA4)
* The "Pull" mode is required for mobile phone not capable for JavaScript.

Making it Happen! A Concrete Win-Win Business Model for a Distributed Social Web (T5A)

URL: http://iiw.idcommons.net/Biz_Model_on_Distributed_Social_Web

Convener: Rolf von Behrens @rolfrb

Notes:

Directory Federation: XRI Naming and Discovery for LDAP (T5C)

URL: http://iiw.idcommons.net/Directory_Federation

Convener: Michael Schwartz, Founder Gluu
Notes-taker(s): Michael Schwartz

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

WHY

Enable organizations to share identity information in bulk, or to allow users to query information from more than just their home organization.

LDAP was for internal organizational use

Its so annoying having to do a useradd for each host
However
Inter-domain : LDAP servers cant talk to each other
Different schemas
Different namespace (dc=blah, o=blah)
ACIs based on BIND DN
Cant BIND a user
No way to do discovery
Host / Port / SSL

XRI LDAP Discovery

@gluu/(+ldap)
@gluu/(+ldaps)
Information in XRD:
 port
 host
 baseDN
 Schema
 Namespace (what ous are present)

i-number XRIs uniquely identify leaf entries
----

inum=${i-number}
Examples:
inum=!gluu.d6f2.6fcd.8399.326d,ou=people,dc=gluu
inum=!custa.1e5d.52c4.ea30.ef39,ou=groups,dc=custa
inum=!custb.713f.375a.1f01.cb33,ou=devices,dc=custb

i-name XRIs optional attribute value
iname: =nynymike

Sample XRD

```
<Service priority="10">
     <Path select="true">(+ldaps)</Path>
     <ldap:host>ldap.company.net</ldap:host>
     <ldap:port>389</ldap:port>
 <ldap:schema type=string desc=>givenName<ldap:schema>
    .
    .
    .
</Service>
```

New Functionality Needed For Servers:
Servers can reference entries in other directory services for ACIs

aci: allowREAD: @gluu*mike
aci: membeOf:@custa.PayrollAdministrators

Sample Applications

Communities or Virtual Organizations that could enable a way to publish information about people from diffenent organizations under one virtual LDAP tree.

Honey-Roasted Death-Camp Salad (T5D)

URL:

Convener: Xianhang Zhang
Notes-taker(s): Geoffrey Bilder

Tags for the session - technology discussed/ideas considered:

buzzwords, centralization, accountability, anti-patterns

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

The session started with the observation that in the food industry there are certain words/phrases than are often prepended or appended to make otherwise unpalatable/unhealthy items sound delicious/nutritious. Examples of these words/ phrases include “honey roasted” and “salad”.

“Salmon” isn’t normally a big seller. “Honey roasted Salmon” is a major feature of most family restaurants. And, of course, we are all familiar with “potato salad”, “pasta salad”, “roast beef salad”, “taco salad” and (ugh) “marshmallow salad”.

Technology seems to have similar words. Some of them include “open”, “distributed”, “user-centric”, “lightweight”, “framework”, “enterprise”, “federated”, etc.

So, for example, we probably all consider “DRM” to be misguided, intractable problem, yet one could imagine somebody making DRM sound plausible by introducing “open DRM” and a “distributed user-centric open DRM framework”. Which, on the face of it, would be nonsense, but from a marketing perspective might gain some traction.

We are not so sure that these words can make “death camp” sound palatable.

In a bit of a leap, we wondered if, in some cases, our predilection for jumping to use phrases like “distributed”, “open” and “user-centric” are a knee-jerk reaction that might, in fact, blind us to alternative ways of addressing the problems that proponents of “distributed”, “open”, etc. are trying to address.

For example, in most cases, when one advocates a “distributed” or “open” technology, what we are *really* doing is trying to come up with a technological safeguard against a centralized, proprietary (and, by implication “commercial”) service becoming too dominant and un-sympathetic to user needs and concerns.

But there seems to be a potential flaw to this approach.

For one thing, we observed that

“distributed begets centralized.”

That is, it seems that for every “distributed” system that we have created, we have then had to create a centralized system to make it useable again. Think of ICANN, DNS, Pirate Bay, Google, or even Kaliya Hamlin!. Counterexamples are frighteningly hard to come by. We may have even identified an anti-pattern.

The implication of this is that, even if we do succeed in creating a truly distributed, user-centric identity infrastructure, there is the very real possibility that a centralized service will come out that aggregates this information in order to make it usable again. And what happens if this entity turns “evil”?

This made us think that perhaps we are misapplying our energies. Maybe, instead of trying to create a distributed technical infrastructure that might still be co-opted by a centralized service, maybe we should focus some of our efforts on anticipating the need for a centralized service and making sure that *we* create this service and that it is unequivocally accountable to the communities it serves.

This approach, of course, would require us to refocus our efforts on hacking institutions and organization structures instead of hacking technology. Understandably, we might all be reluctant to do this because it isn’t in our core skill sets.

Open ID v. Next Discovery (T5E)

URL: http://iiw.idcommons.net/OpenIDvNext_Discovery

Convener: Allen Tom & Mike Jones
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

*Implications of User Owned/Controlled Data as Official Government Policy *The Mydex UK Community Prototype (T5G)*

URL: http://iiw.idcommons.net/Implications_of_User_Owned_Controlled_Data_as_Official_Government_Policy

Convener: William Heath
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Google As An OpenID RP (T5I)

URL:http://iiw.idcommons.net/Google_as_an_OpenID_RP

Convener: Ilan Caron, Eric Sachs, Yaniv Shuba
Notes-taker(s): Jacky Wang , Yaniv Shuba

Tags for the session - technology discussed/ideas considered:

Technology discussed / demo / google practice

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

1. Currently, Google accepts OpenID login for Blogger, Moderator, FriendConnect, Appengine, FreeMusic (in China)

2. GAIA - integrate OpenID into Google account management

3. Create Google account: "easy verification" - half of the Google accounts are created using the yahoo/aol(?) email addresses. Therefore, we'd like to verify whether the user is the guy they claim to be.

4. Hybrid onboarding - oauth plugged-in.

5. Support multiple ID provider protocols, like OpenID, Windows Live ID, and Chinese local ID providers (Renren.com, etc.)

[Demo1: sync email validation]...

- What kind of email addresses are considered to be trusted?

Only the email provided by the same IDP. e.g.: abc@yahoo.com from Yahoo!, which is an IDP.

[Demo2: federated login demo - share a Google doc to the Yahoo user]

- How could user move their email, say, from yahoo to aol?

The scenario is pretty complicated - it includes moving from federated domain to un-federated domain and vice versa, and federated domain to federated domain. It's an on-going effort.

- What's the checklist that an IDP need to go through before Google trust them?

Eric will start a new session on Wed to discuss it.

NOTES BY: Yaniv Shuba

Google's live OpenId RP features:

- Blogger,Moderator App Engine and China Free Music already function as an RP.

- Friend Connect - an easy way to make your site an RP.

Work in progress:

- Integrate OpenId support into Google's login page.

- Easy verification - allow e-mail address verification during account creation using OpenId, instead of the regular procedure where a verification e-mail is sent to the user.

- Hybrid onboarding - authorize Google to get your Yahoo contact information while you sign-in.

- Google is compiling a list of requirements for IDPs to qualify for being IDPs to Google.

- The Google login page will have to change to reflect the different use-cases introduced by federated login.

[Demo1: sync email validation]...

- What kind of email addresses are considered to be trusted?

Only the email provided by the same IDP. e.g.: abc@yahoo.com from Yahoo!, which is an IDP.

[Demo2: federated login demo - share a Google doc to the Yahoo user]

- How could user move their email, say, from yahoo to aol?

The scenario is pretty complicated - it includes moving from federated domain to un-federated domain and vice versa, and federated domain to federated domain. It's an on-going effort.

Day Three Wednesday May 19, 2010 Sessions

Personal Data Store Ecosystem Design (W1A)

URL:

Convener: Kaliya Hamlin
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

(in)Coherent Web (in)Security Policy Framework (W1E)

URL:

Convener: Jeff H
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Bootstrapping OAuth 2.0 Ecosystems (W1F)

URL:

Convener: Justin Smith, Brian Eaton
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Stateless Distributed Membership: An Inquiry (W2E)

URL: http://iiw.idcommons.net/Stateless_Distributed_Membership_an_Inquiry
Convener: Judi Clark
Note-taker(s): Judi Clark

Tags for the session - technology discussed/ideas considered:

openID, identity, personal data store, user-managed access, access control, social expectations, how things work, multiple identities, experiment

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

We explored the possibility of creating a membership site that does not have a traditional membership database (user names, passwords) but instead uses OpenID (or similar) and personal data stores to contribute to the site. A lot of the underlying tools/technology exists already.

- Eve M has spoken about ID data statelessness and this concept is related.

Access to the site (example): OpenID, location to personal data store, and default designation of sharing policy. Sharing policy might include, for example:
 * Sharing: A. Sharing; B. Others in this thread; C. Specific others
 * Storage: 1. cache & call to update, 2. no cache; 3. X days; 4. Permanent until revoked

- Related concept: cache and update
- Recommended reading: Future of Reputation (Solove)

First example: Forum/Conversation

My server stores a unique transaction record key, the openID & policy statement, and other pointers relevant to the specific interaction. For example: if visitors are contributing to a forum or ongoing conversation, my server may have a time/date/conversation ID stamp (each contribution is stored on the visitor's own personal data store (PDS); my server assembles the conversation according to stated policies and availability of visitor PDSs.

- Example of distributed conversations: blog posts and trackbacks
- important underlying concept: Operational Transformation (wikipedia)
- Might try installing a version of Google Wave/Jupiter to start

Second example: Personal RFPs

Similar to a job board or public wish list, my server might offer a commons area which points to Personal Requests for Proposal (RFP) for something that someone wants or offers. A common template for an RFP might include a title, description, price, and way to reach requestor, stored on the requestor's PDS. My server tracks the pointer to the PDS that holds the RFP, a community caution flag, the REL button status, and an expiration date.

General consensus that this was an interesting problem from social angle as well as having many tools that might be applicable. Many of the social norms and expectations have to be discovered or developed & discussed.

Thanks for the very constructive questions, suggestions and observations at this session!

EmanicPay: VRM+CRM Browser Plug-in & Personal Data Store Framework (W2F)

Convener: Doc
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Breaking Up With Atom Activity Streams JSON (W2G)

URL: http://iiw.idcommons.net/Breaking_up_with_Atom_Activity_Streams

Convener: Martin Atkins and Monica Keller
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

DNSSEC Explained (W3A)

URL: http://iiw.idcommons.net/DNSSEC

Convener: Esther Makaay
Notes-taker(s): Esther Makaay

Tags for the session - technology discussed/ideas considered:

DNS / DNSSEC

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

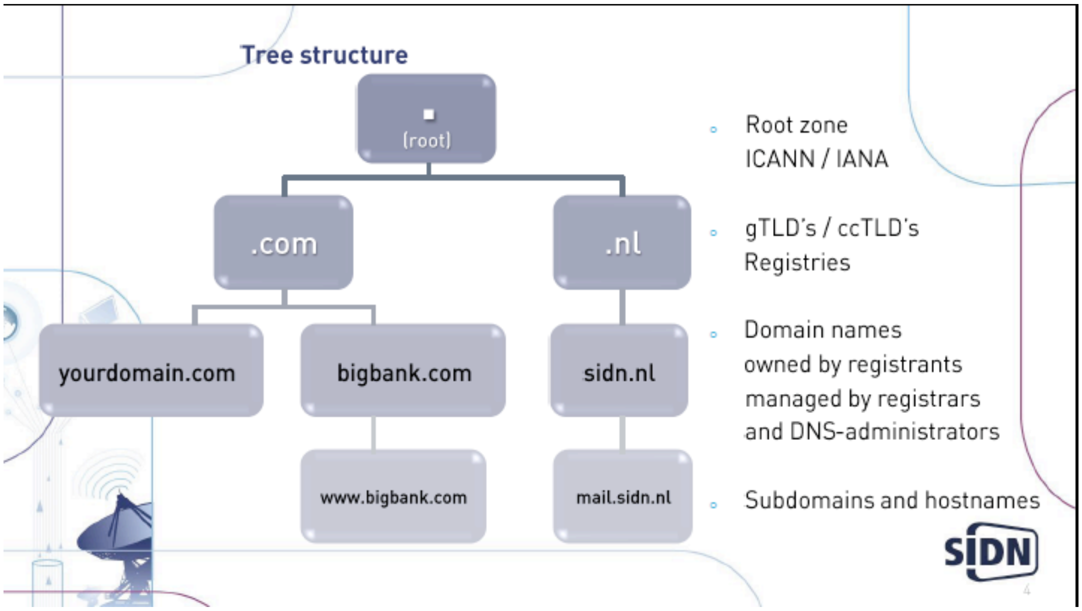
Presentation about DNS vulnerabilities and DNSSEC as a solution.

Excerpt of a presentation given at Infosecurity Brussels. The more elaborate slideshow on this topic can be found at:
http://www.slideshare.net/esthermakaay/dnssec-towards-enhanced-internet-security

DNS & vulnerabilities

DNS (domain name system)

- Road signs
- Hierarchical and distributed. Highly scalable and robust.
- Scalable, distributed data, delegated (tree-structure)
- 'DNS = the internet' (no, but it is to the vast majority of the end-users)
- One of the oldest protocols out there



Tree structure

. (root)

.com

.nl

yourdomain.com

bigbank.com

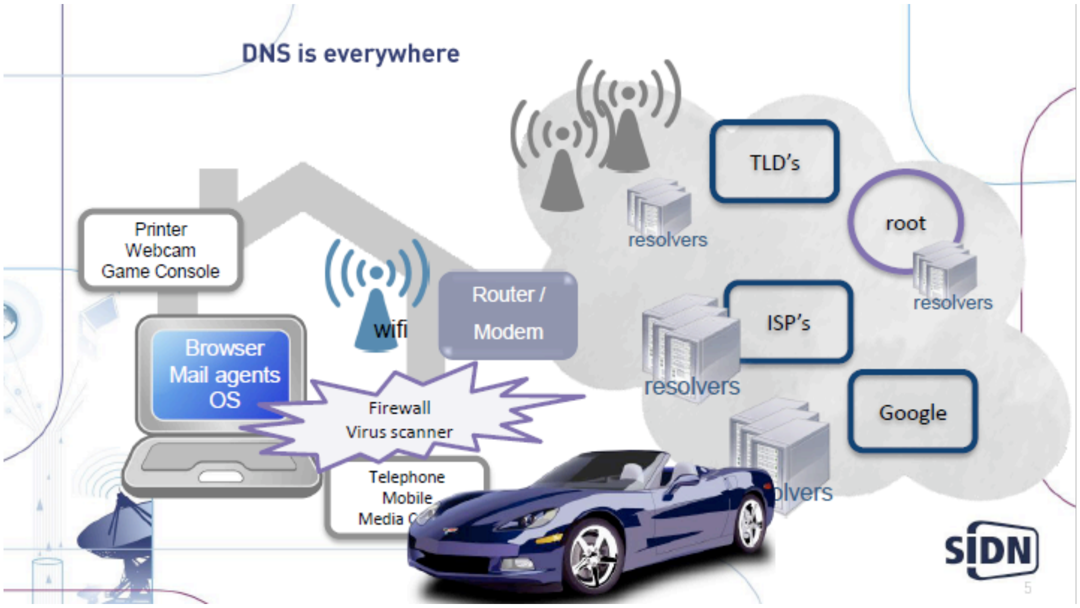
sidn.nl

www.bigbank.com

mail.sidn.nl

- Root zone
  ICANN / IANA
- gTLD's / ccTLD's
  Registries
- Domain names
  owned by registrants
  managed by registrars
  and DNS-administrators
- Subdomains and hostnames

SIDN



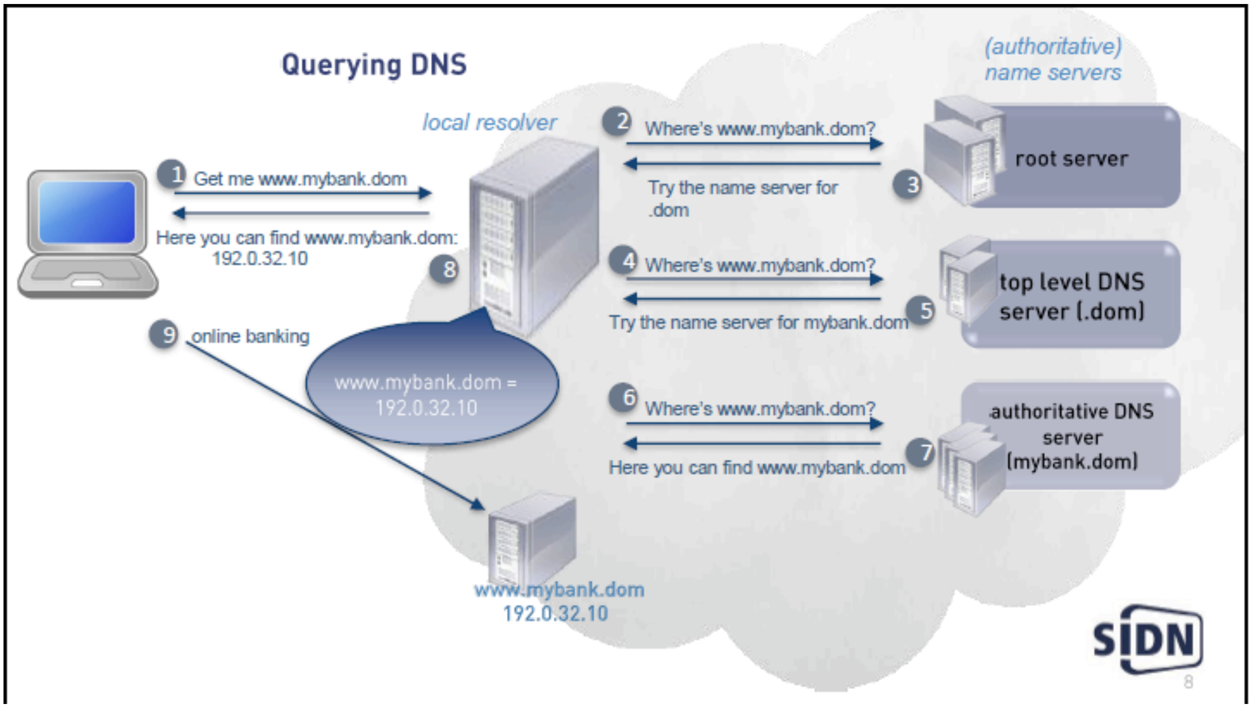
DNS is vulnerable

- Never designed for trust
- Nothing really changed since 1983
- Usage has broadened (functionality and quantity)
- Everybody wants (to handle) your DNS

Risks:

- Availability (no DNS – no internet)
- Integrity (wrong DNS – wrong internet)

DNS integrity



Querying DNS

(authoritative) name servers

local resolver

1 Get me www.mybank.dom

Here you can find www.mybank.dom: 192.0.32.10

2 Where's www.mybank.dom?

Try the name server for .dom 3

root server

4 Where's www.mybank.dom?

Try the name server for mybank.dom 5

top level DNS server (.dom)

6 Where's www.mybank.dom?

Here you can find www.mybank.dom 7

authoritative DNS server (mybank.dom)

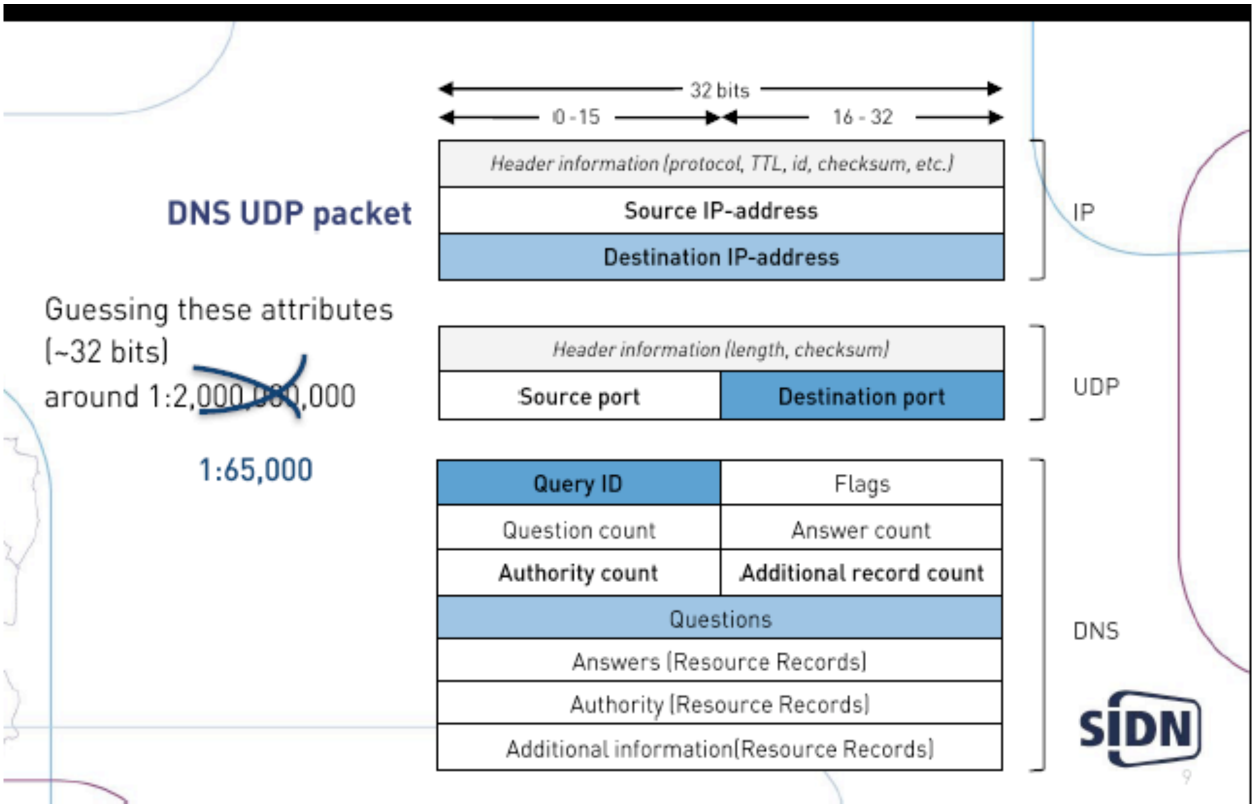
8

www.mybank.dom = 192.0.32.10

9 online banking

www.mybank.dom 192.0.32.10

SIDN



DNS UDP packet

Guessing these attributes (~32 bits)

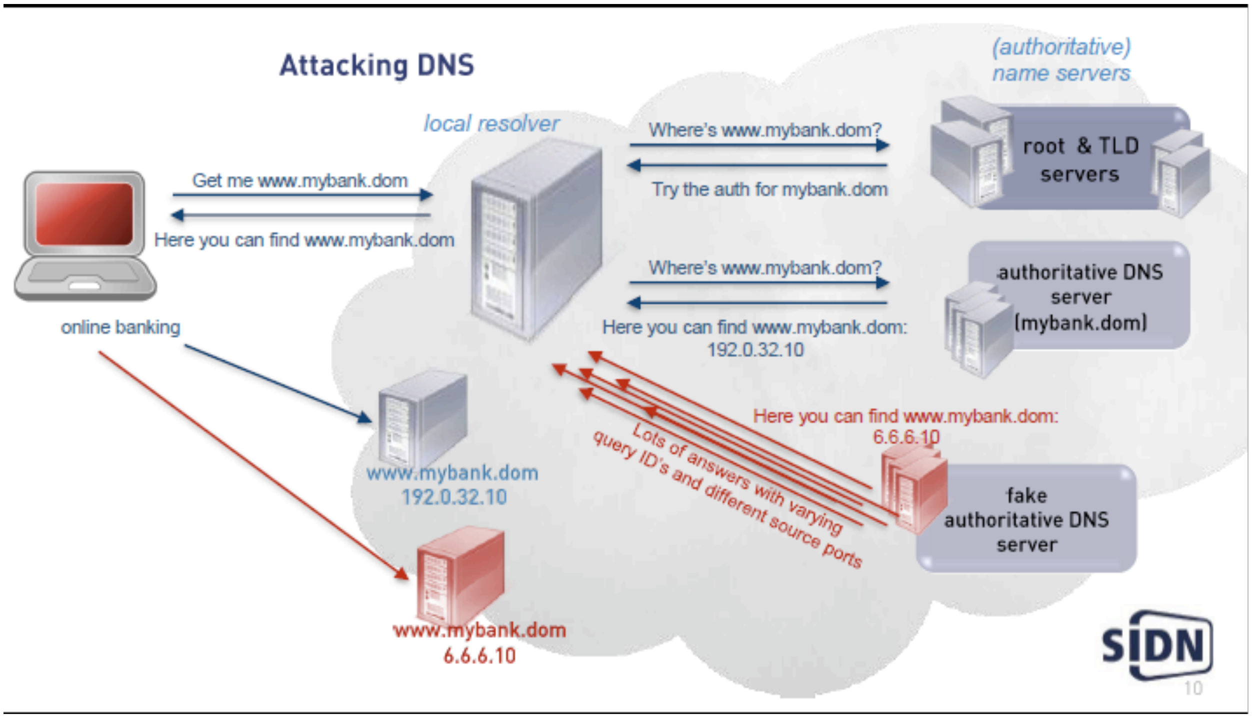
around 1:2,~~000,000~~,000

1:65,000

32 bits

0 - 15	16 - 32	
Header information (protocol, TTL, id, checksum, etc.)		IP
Source IP-address		
Destination IP-address		
Header information (length, checksum)		UDP
Source port	**Destination port**	
Query ID	Flags	DNS
Question count	Answer count	
Authority count	**Additional record count**	
Questions		
Answers (Resource Records)		
Authority (Resource Records)		
Additional information(Resource Records)		

SIDN



Attacking DNS

(authoritative) name servers

local resolver

Get me www.mybank.dom

Here you can find www.mybank.dom

Where's www.mybank.dom?

Try the auth for mybank.dom

root & TLD servers

Where's www.mybank.dom?

Here you can find www.mybank.dom: 192.0.32.10

authoritative DNS server (mybank.dom)

online banking

www.mybank.dom 192.0.32.10

Here you can find www.mybank.dom: 6.6.6.10

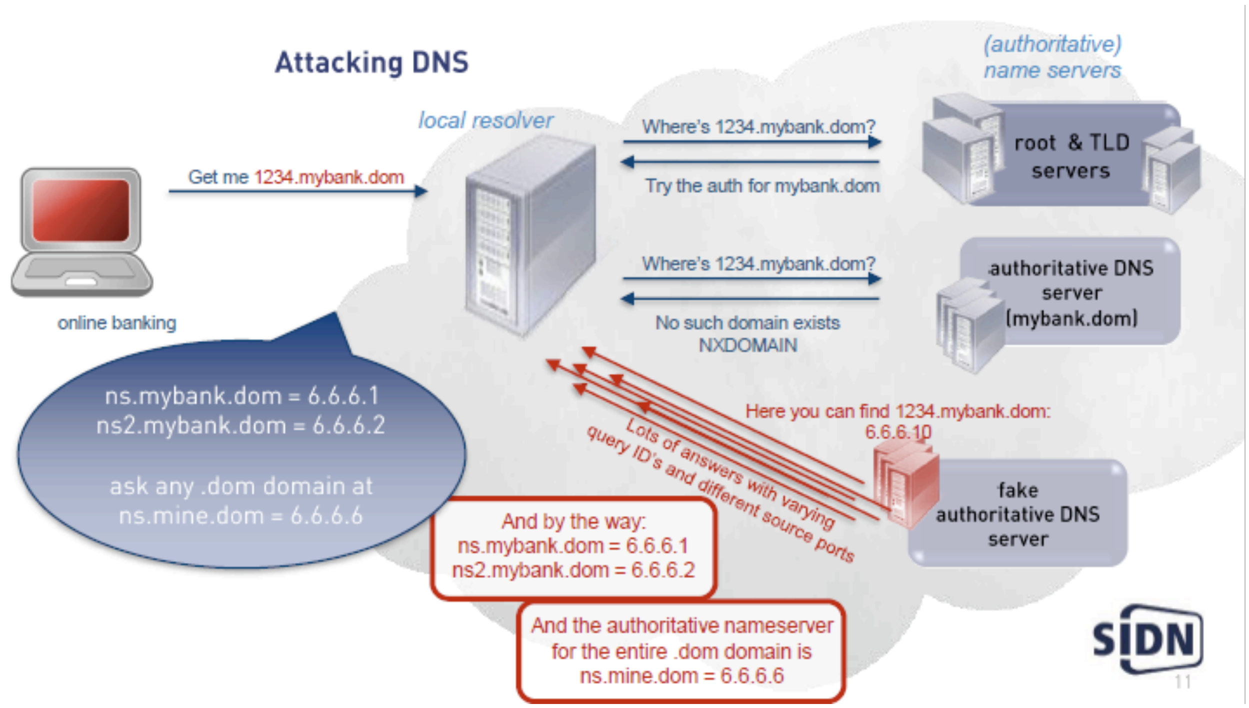
Lots of answers with varying query ID's and different source ports

fake authoritative DNS server

www.mybank.dom 6.6.6.10

SIDN

10



Attacking DNS

(authoritative) name servers

local resolver

Get me 1234.mybank.dom

Where's 1234.mybank.dom?

Try the auth for mybank.dom

root & TLD servers

Where's 1234.mybank.dom?

No such domain exists NXDOMAIN

authoritative DNS server (mybank.dom)

online banking

ns.mybank.dom = 6.6.6.1
ns2.mybank.dom = 6.6.6.2

ask any .dom domain at
ns.mine.dom = 6.6.6.6

Here you can find 1234.mybank.dom: 6.6.6.10

Lots of answers with varying query ID's and different source ports

fake authoritative DNS server

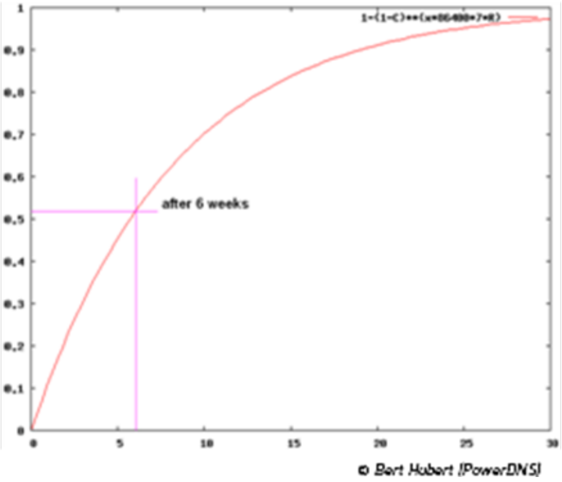
And by the way:
ns.mybank.dom = 6.6.6.1
ns2.mybank.dom = 6.6.6.2

And the authoritative nameserver for the entire .dom domain is ns.mine.dom = 6.6.6.6

SIDN

11

Chances to spoof a resolver (Based on research by Bert Hubert (PowerDNS))
In theory (50,000 queries/second): static source port – 10 seconds, random source port – 36 hours
In practice: slow attack, 100 queries/second– 30 weeks, 50% success after only 6 weeks

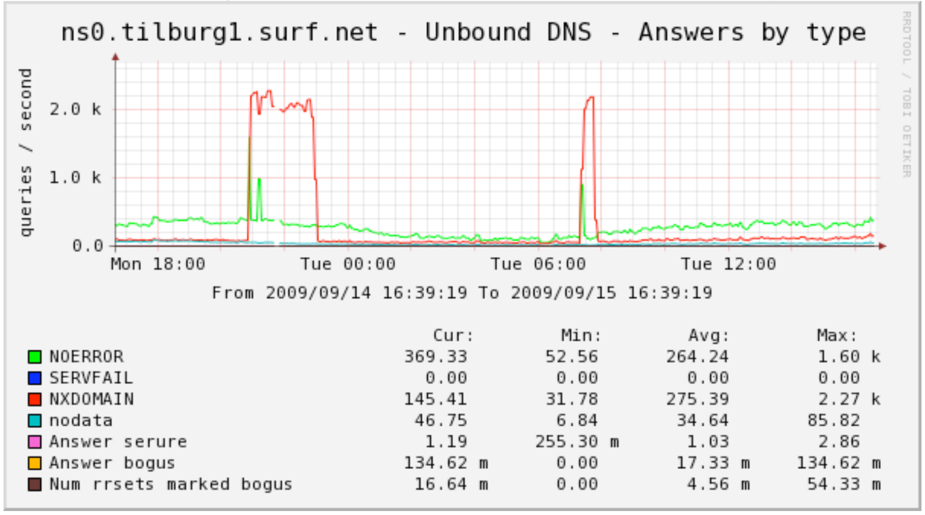


© Bert Hubert (PowerDNS)

The (slow) attack is happening
Scarce media reports about attacks

- Users from several large ISP’s and telco’s have suffered from misdirection and outages, but also specific spyware, spam and pay-per-click trojans.
- Customers of a large bank were redirected to fraudulent websites that attempted to steal passwords and install malware

Statistics show peaks for NXDOMAIN answers (courtesy of SURFnet)



From 2009/09/14 16:39:19 To 2009/09/15 16:39:19

	Cur:	Min:	Avg:	Max:
🟩 NOERROR	369.33	52.56	264.24	1.60 k
🟦 SERVFAIL	0.00	0.00	0.00	0.00
🟥 NXDOMAIN	145.41	31.78	275.39	2.27 k
■ nodata	46.75	6.84	34.64	85.82
■ Answer serure	1.19	255.30 m	1.03	2.86
🟧 Answer bogus	134.62 m	0.00	17.33 m	134.62 m
🟫 Num rrsets marked bogus	16.64 m	0.00	4.56 m	54.33 m

Countermeasures:

- Patch against attacks
  - add entropy by port randomisation, case sensitivity (DNS 0x20) or EDNS-PING
  - cache time-outs, ask twice/ask thrice, use TCP

- Monitor your network and servers
    - look for brute force attempts
    - count ‘near misses’ (correct port / failed ID)
    - Restrict queries to the intended users

Or implement DNSSEC

DNSSEC

A set of extensions to DNS which provide:

- origin authentication
- data integrity
- authenticated denial of existence

Metaphor (Olaf Kolkman)

- Compare DNSSEC to a sealed transparent envelope.
- The seal is applied by whoever closes the envelope
- Anybody can read the message
- The seal is applied to the envelope, not to the message

“The Domain Name System Security Extensions (DNSSEC) add data origin authentication and data integrity to the Domain Name System.” (RFC 4033)

Verifying authentic answers

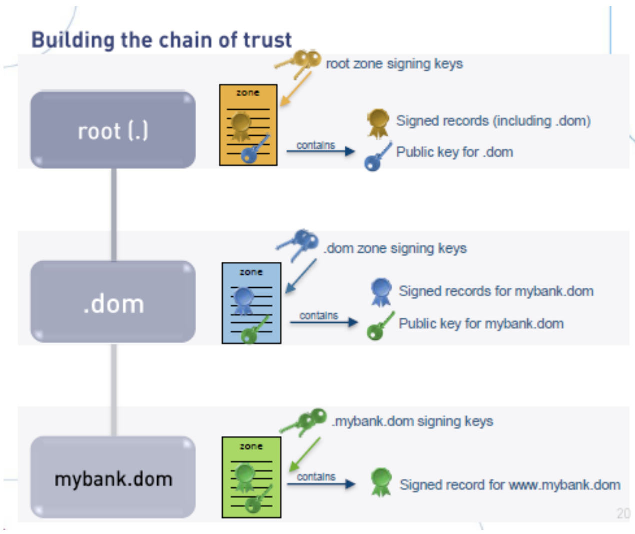
- Authoritative servers:
    - Add digital signatures to resource record sets
    - Add public key to domain zone
- Resolvers:
    - Validate the signatures to the public key
    - Only accept verified responses

Public key cryptography (RSA, DSA, (Elliptic Curve))

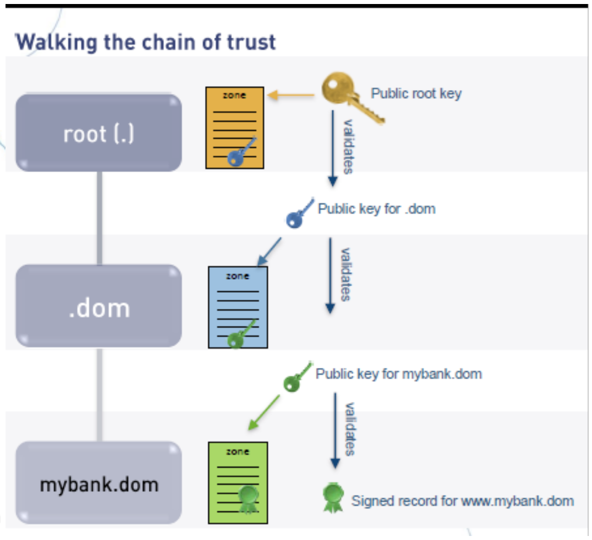
- Private key for signing (protected and hidden)
- Public key for verification (widely published)

Mini-howto (simplified)

- Create the keypairs
- Add the public keys to the zone
- Sign the zone
- Publish the signed zone
- Notify parent
-



Building the chain of trust

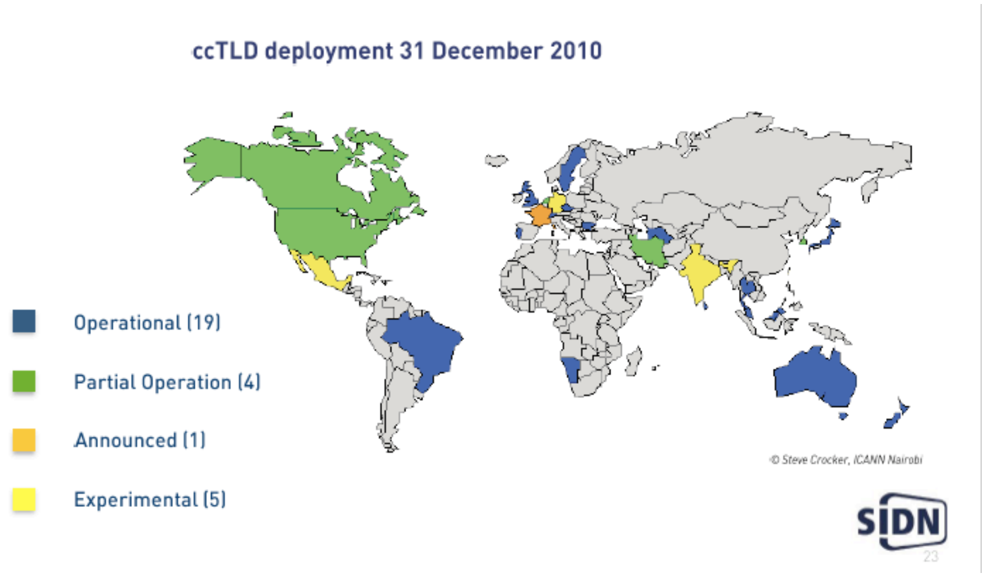


Walking the chain of trust

Validating and resolving

- Validating resolver needs configured trust anchors
    - Will only need the root zone key in the future
- Possible types of answers (security status of data – RFC 4035)
    - secure – chain of trust is built from trust anchor
    - insecure – chain of trust can not be built from trust anchor (confirmed)
    - bogus – chain of trust should be built, but can not be built from trust anchor
    - indeterminate – unable to determine whether an RRset should be signed

Resolver will only answer queries that are secure or insecure



ccTLD deployment 31 December 2010

🟦 Operational (19)

🟩 Partial Operation (4)

🟧 Announced (1)

🟨 Experimental (5)

© Steve Crocker, ICANN Nairobi

DNSSEC Software

Tools are being developed and have become available:

- OpenDNSSEC (www.opendnssec.org)
- Signers: Secure64, Xelerance
- PowerDNSSEC
- Windows Server 2008 R2
- other vendors have (announced) products

Resolver software widely available:

- Unbound (NLnetLabs)
- BIND 9.x and up
- Windows Server 7

OpenDNSSEC
OpenDNSSEC takes in unsigned zones, adds the signatures and other records for DNSSEC and passes it on to the authoritative name servers for that zone.

DNSSEC is complex

- Operational issues
  - Domain transfers result in outage
  - Key rollovers need coördinated timing
  - Multiple scenario's and policies at different parents
- One size does not fit all
- Small errors and bugs take time to resolve

DNSSEC does not

- protect against packet sniffing
- provide confidentiality
- protect against DoS attacks (contrary)
- protect against phishing, pharming, typosquatting

There will always be other risks!

No real alternatives

- SSL / TLS
  - Doesn't prevent cache-poisoning
  - Too heavy to be deployed for name servers
- TSIG / SIG (0)
  - Not scalable (shared secrets)
  - Only secures transactions, not records
- DNScurve
  - No operational implementation or widescale deployment yet

What you can do

- Start resolving and validating DNSSEC
  - Gain early real-life DNSSEC experience
  - Detect and solve issues before the root is signed

- Put DNSSEC-support in the requirements for new equipment
- Plan for the future
- Clean your house:
    - current DNS implementations and administration
    - network-components that impact DNS

Resources and further reading

- Hardening the internet – Whitepaper on DNSSEC. http://www.dnssec.nu
- OpenDNSSEC – Development of open source software for automation of zone signing. http://www.opendnssec.org/
- ENISA Good practices guide for deploying DNSSEC. http://www.enisa.europa.eu/act/res/technologies/tech/gpgdnssec
- DNSSEC Industry Coalition. http://dnsseccoalition.org
- Independent site with lots of information http://dnssec.net
- DNSSEC.nl - Platform aimed at finding solutions for open issues that are blocking widespread DNSSEC deployment in the Netherlands http://www.dnssec.nl

OpenID Certification Profile (W3B)

URL: http://iiw.idcommons.net/Certifying_Open_ID,_IdPs,_RP

Convener: Eric Sachs
Notes-taker(s): Eric Sachs

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Trusted Email Profile

Use case: Many websites with a large installed-base of accounts that login with Email & password would like to become OpenID relying parties. However they only way to support IDPs who provide a higher success rate then their current approach for both registration and login rates. Instead of each website identifying those IDPs, they would like a central neutral organization like OIX to maintain a list of IDPs who meet certain known requirements. The following profile lists those known requirements.

The IDP must:

- support a PAPE request to indicate that the IDP's assertion should follow this specific certification profile
- meet all requirements of the GSA OpenID ICAM Profile except the requirement to avoid sending PII to the RP
- use an authentication scheme that is at least as strong as the suggested best practices for the ICAM profile
- have a historic 99.5% uptime of its authentication and OpenID IDP system
- NOT require the RP to pre-register with the IDP or enter into a legal contract with the IDP to use that IDP API (similar to the model of SMTP)
- support OpenID discovery based on either the domain name (using directed identity) or an Email address in that domain (using webfinger)
- support AX requests for the AX "email" parameter and return that parameter on every request, even if it has not changed
- only return the email address that the logged in account receives over the open Internet via the IDP's SMTP service (and thus is equivalent to traditional email validation)
- return a global, unchanging and non-recycled OpenID claimed URL for the account
- show at most one page in 99% of the consent flows once the user is authenticated

- default to NOT requiring the user to re-enter their password during the OpenID flow if the user had already been authenticated by the IDP before the OpenID request was made
- default to auto-approving future logins by a user to the same RP
- support checkid_immediate
- support the PAPE openid.pape.max_auth_age parameter, though it can choose to always re-authenticate the user no matter what value is passed in that parameter
- auto-detect mobile and non-JS browsers and show consent pages that a friendly for them

SMART UMA (W3C)

URL: http://iiw.idcommons.net/SMART_UMA

Convener: Maciej Machulak
Notes-taker(s): Eve Maler

Tags for the session - technology discussed/ideas considered:

UMA site: http://kantarainitiative.org/confluence/display/uma/Home
SMART project slides: http://kantarainitiative.org/confluence/download/attachments/38371737/SMARTOverview.pdf
Screenshots of SMART prototype demo: http://kantarainitiative.org/confluence/display/uma/SMART+project+user+experience
UMA CV-sharing scenario: http://kantarainitiative.org/confluence/display/uma/cv_sharing_scenario

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

The SMART project at Newcastle University is for Student-Managed Access to Online Resources. It's based on UMA, and UMA is based on OAuth 2.0, such that an UMA requester has to present an access token to get access to a user's resources at a host. The user's authorization manager decides whether to hand out access tokens based on user policy.

The SMART project objectives are:

* Define a scenario that focuses on higher ed, and provide a comprehensive requirements analysis
* Develop an UMA-based solution
* ...

Newcastle University has 4500 staff members and 19,000 students. A lot of data (both personal info -- DOB, address, resumes, etc. -- and resources such as documents) is hosted by Newcastle. It needs an efficient, secure, and usable access management system that supports both data owners and data consumers. E.g., you may want to share your research selectively with some collaborators.

The project team integrates researchers, developers, and information systems management personnel.

The UMA "CV-sharing scenario" is the basis for the scenario being worked on in the project. Today, a student has to manually assemble a set of artifacts to provide to prospective employers. If the student is still in classes, some of this data needs to be refreshed (like their marks from classes they've taken).

Question: What about transitivity? If a professor writes a letter of recommendation for a student, and the student wants to include it in a prospective-employer resource bundle for further sharing, does the professor give access to the student in such a way that the student can then transitively grant access to another party without needing to go back to the professor? Yes, through a system of demanding claims.

In some cases, the materials are digitally signed, or may be packaged software.

Some job search websites have you upload a bunch of data, and then prospective employers go to the job search site to see it.

Question: Can the professor has read/write/append rights to the letter, the student has read/append rights, and others have only read rights? Yes.

The project team did an analysis of the ways resources are being shared in the university, and web applications being used for this. It turned out the web apps didn't support cross-university collaboration groups.

If there are two universities, A and B, each typically serves as an IdP for their own populations and their own web applications that respect that IdP. Some applications are allowed access to the resources of other universities by becoming relying parties to the other IdP. So a student at university B can access certain resources at university A, but only if A's web app can talk to the IdP of B.

So what happens right now is that the Grouper framework is used to manage groups of identities. A cross-university collaboration group could be created at Grouper, and the particular apps that need it are told about the group and how to connect to the Grouper server.

One goal of the project is to eliminate the Grouper entity, and replace it with an UMA authorization manager that works with the Shibboleth higher-ed federation as a repository of policies that govern access.

Another goal is to enhance the eScience system (which stores resources for collaboration with others) to allow it to point to resources "in the cloud" instead. This will allow researchers to use whatever web apps they prefer to create the research but also allow eScience to have access to that research. Today it's sort of like SharePoint :-), where you have to upload files. Through SMART, it will become "just another web app" in the research ecosystem.

The project started about five weeks ago, but they have already got a prototype/demo (shown live in this session and at Tuesday's demo session).

* You store photos on a particular host site.
* You tell the site that you want it to use "smartam" for protecting the resources hosted there, but giving it the URL of the AM.
* You get redirected to smartam and are asked to approve the connection between this host and this AM, in an OAuth 2.0 user delegation flow.
* Thereafter, on the AM, you can browse around a description of the resources that are now protected at that host.
* You provide the URL of a protected resource to some requester.
* The requester has to learn where the AM is and go through an UMA dance to get permission to obtain the resource.
* For the purposes of the demo so far, the requester is asked to log in at the AM to prove their suitability for access, but the ultimate goal of the project is to have them prove this by means that are not tied to AM authentication/identification.
* In the case of the second protected resource, it demands that the requester agree (by checking checkmarks) that they are over 18 and agree to the further sharing constraints imposed by the authorizing user.

All the code will be open-sourced, and full documentation will be made available. They want to provide a solid set of UMA libraries.

Question: What about CMS's that use LDAP today? Could this software work as a wrapper? A: It wouldn't be a wrapper, but there is a goal to integrate with LDAP.

Catalyst Interop Planning (W3E)

URL:

Convener: Mary Ruddy, John Bradley, Drummond Reed, Don t.
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

PCITF: Payment Card Industry Trust Framework (W3I)

URL:

Convener: Sid Sidner
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

THIS Session has a PowerPoint Slide Deck

7 Deadly Sins of Distributed Authentication (W4A)

URL: http://iiw.idcommons.net/7_Deadly_Sins_of_Distributed_Authentication

Convener: Brad Hill
Notes-taker(s): Brad Hill

Tags for the session - technology discussed/ideas considered:

Security, Security Flaws, Cryptography, Protocols, Implementation

http://groups.google.com/group/iiw-common-problems-dist-authN

iiw-common-problems-dist-authN@googlegroups.com

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

This session was based on the idea that the many common problems, protocols and implementations of distributed authentication and identity systems also lead to a common set of flaws, pitfalls and shortcomings. The goal was to propose a set of "most common mistakes", in the spirit of the OWASP Top 10 Most Common Web Application Flaws, (http://www.owasp.org) to inform designers, implementers and deployers of these technologies on the informal security history and academic literature of similar technologies.

The discussion started with the following outline, supplied by Brad Hill:

7 Deadly Sins of Distributed Authentication:

1. Unconstrained Delegation
   - Passwords
     - Give them to somebody, they can use them to authenticate as you to anyone else.
   - Bearer Tokens
     - Message level security needs to have a target scope: Who was this artifact intended for?
   - Kerberos Delegation v1
     - Constrained sources, unconstrained targets for those sources. Abuse of "trusted subsystem" model.

2. Forwardable Credentials
   - NTLM
     - Not delegatable, definitely forwardable.
     - Protocols that don't provide server authentication or scope and verify artifacts are vulnerable.

3. No Channel Binding
    - More sophisticated subset of credential forwarding.
    - Auth the transport channel one-way
        - Server to client (TLS)
        - Auth inside the client the other way
        - Client to server (NTLM, Kerberos bearer tokens, SAML)
        - No binding between the two creates confused deputy possibility, forwarding.
    - Possible even with client certs – recent renegotiation bug.

4. Bearer Tokens
    - Yes, it matters
    - Kerberos is a great example
        - Mutual auth, strong crypto, key exchange
        - Use it as a bearer token, it becomes vulnerable

5. Unscoped Authority
    - Example: Boeing & Air Force federate
        - Air Force wants to accept Boeing's identity assertions
        - Boeing should be able to assert joe@boeing.com
        - NOT: admin@airforce.mil or joe@lockheed.com
        - This is especially a big problem for systems that provide mutual authentication in a federated manner.
        - Server names must be scoped!
    - Kerberos SID Filtering

6. PKI, PKIX and SSL/TLS
    - Problems of unscoped authority.
        - Any public authority can assert any name.
        - Can any PKIX roots assert enterprise or other non-public identities?
    - Acceptance policies.
        - Key usage, EKU, etc.
            - Is this a server cert? A client cert?
        - Algorithms, key strengths
    - Trusting useless assertions
        - ***NO ASSURANCE*** for client certs or server certs for non-public TLDs as of 2005-9.
    - Wear a Belt & Suspenders for very high value services.
        - e.g. Windows Update

7. No Upgrade Path
    - Yes, adoption is important.
    - Build a strong and a weak system
    - Price them differently

Good Practices

- Scoped, self-describing artifacts
- Mutual authentication
- Forwarding-resistant credentials
- Channel binding
- Key agreement
- Proof-of-possession
- Incentives to use high assurance protocols

Implementation Gaffes

- Not verifying signatures.
  - Trusting encryption when integrity is needed – e.g., in SAML messages.
    ```
    <saml>
    <encrypted>  …  </encrypted><yourPublicKey/>
    <signed>  …   </signed><myPublicKey/>
    </saml>
    ```

- Weak Crypto / Incorrect Crypto
  - Stream ciphers. Don't.
  - Using RSA to Encrypt and Sign arbitrary data.
  - ECB mode.
  - Encryption without Integrity
- GUIDS, UUIDS and Randomness
  - Just make it random unless you have a good reason why it shouldn't be.
    - Only need to seed and occasionally update PRNG with good randomness.
  - Minute possibility of a collision in a 128 bit random space is exactly the guarantee you're hanging your hat on.
  - Old IETF draft of GUIDS not really that random – don't use it.
- HMAC Verification Timing
  - Don't use a standard string or array comparison function to verify HMACs.
  - They all short-circuit as soon as they find a mismatched character.
  - Timing difference between good validation and bad validation.
    - Brute force correct HMAC one character at a time.

C and machine language comparisons that look like constant time may be OK, but test. Java and .Net are subject to optimizations that make code that looks like it should produce constant time results not. Adding a randomized delay was suggested. Suggestions from the group for better algorithms included double-hashing the HMAC before comparing, or starting the comparison at a random location in the HMAC string.

Other proposed additions to the list of common sins included:

Key Distribution

- Administration
- Rollover
- Key Lifetime
- “Heavy” Keys

Failure to Manage the Ecosystem and Dependencies

- Verifying the implementation and policy of relying parties / consumers

Failure to Explicitly Distinguish Implementation vs. Policy

- Acceptance policy
- Issuance policy

“Writing Your Own”

- Developers have learned in the last 10 years they shouldn’t write their own crypto algorithms. Need to learn now that they shouldn’t write their own distributed authentication and identity protocols.
- IdPs and protocol implementers should provide and consumers should use standard libraries.
- Usability should focus on providing good APIs and libraries. Making a protocol that anyone can implement in Perl overnight should be a non-goal, or at least subordinate to making a protocol that gives participants strong security guarantees.

Significant interest was expressed, at least among the attendees at the session, in formalizing the discussion into a paper, including a good bibliography.

Google Group created to facilitate discussion around creation of such a paper, at:

http://groups.google.com/group/iiw-common-problems-dist-authN

iiw-common-problems-dist-authN@googlegroups.com

Suggested sources immediately relevant to the discussion included:

Ross Anderson - Robustness Principles for Public Key Protocols
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.44.3669

Martin Abadi and Roger Needham – Prudent Engineering Practice for Cryptographic Protocols http://portal.acm.org/citation.cfm?id=229714

Carl Ellision and Bruce Schneier – Ten Risks of PKI
http://openweb.or.kr/10_risks_of_pki.pdf

Afghan Anarchy: How Can I Help Paranoid People Share/Trust TODAY (W4C)

URL:

Convener: Cam Hunt
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

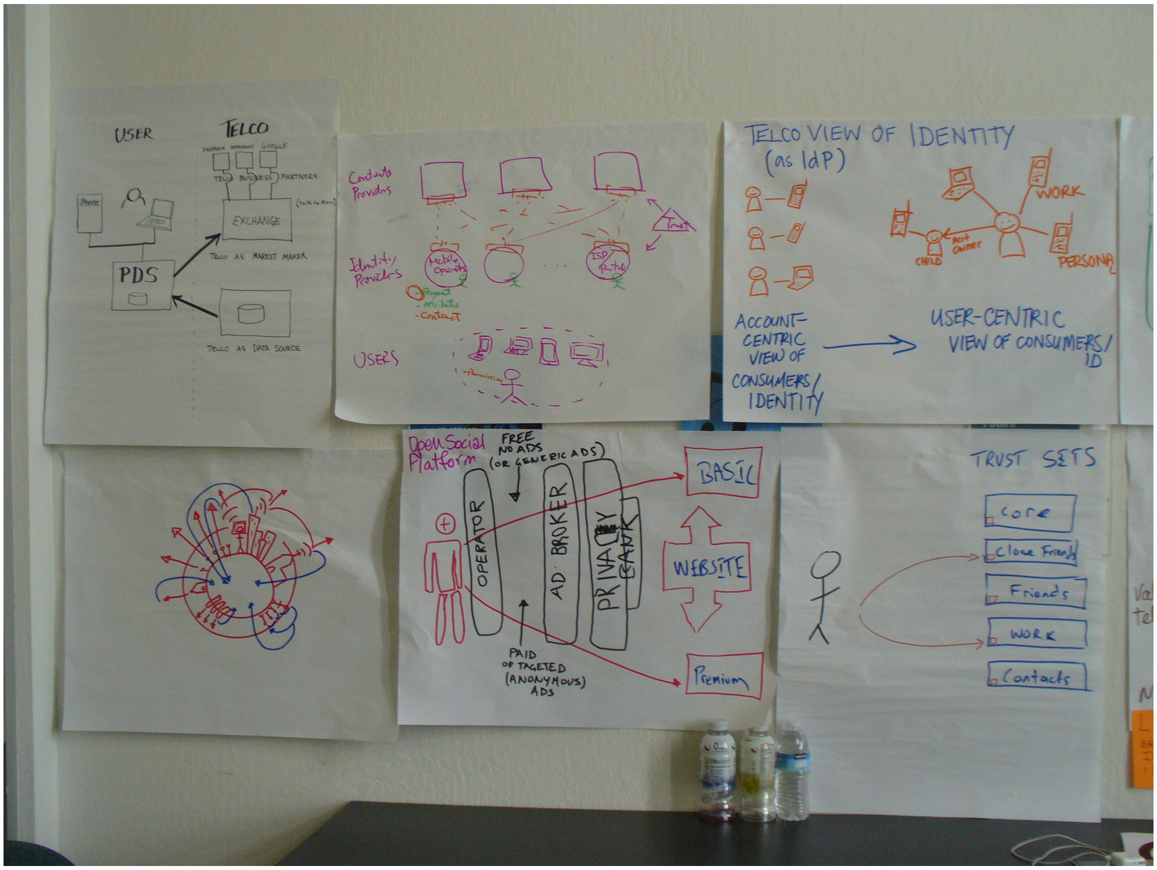
TELCO/WEB/DATA – Meta Story (W4G)

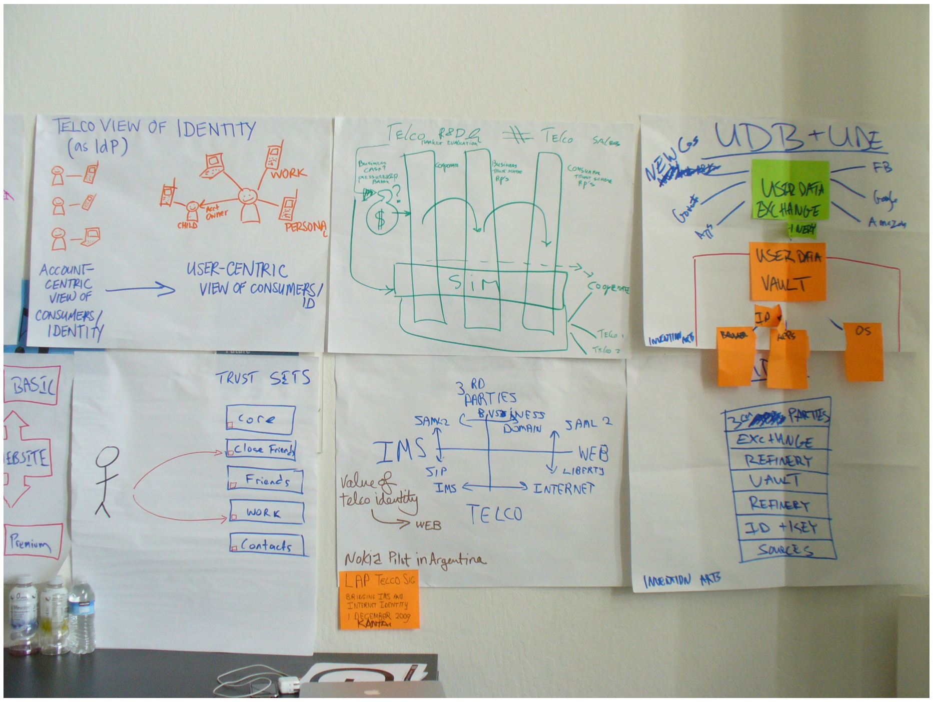
URL: http://iiw.idcommons.net/Telco/Web/Data_Meta_Story

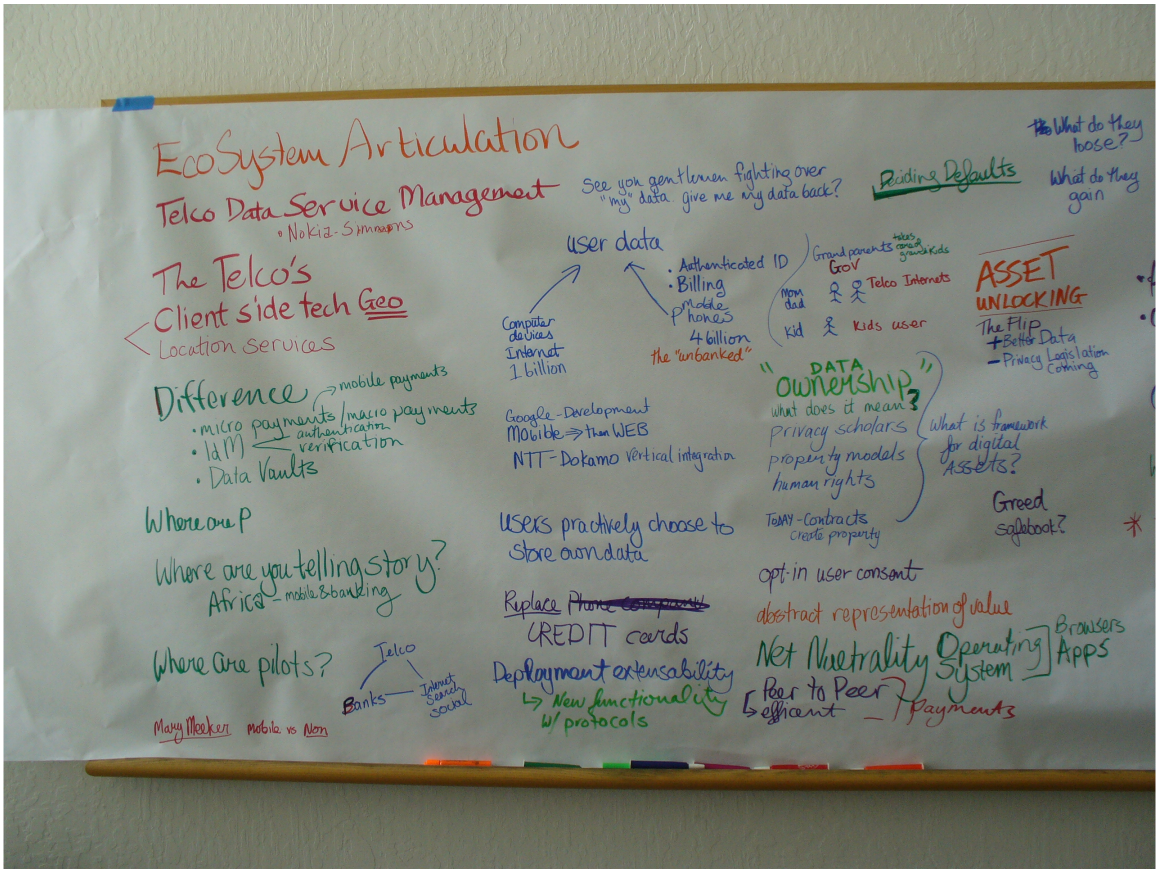
Convener: ?
Notes-taker(s): Jim Fenton (sent photos)

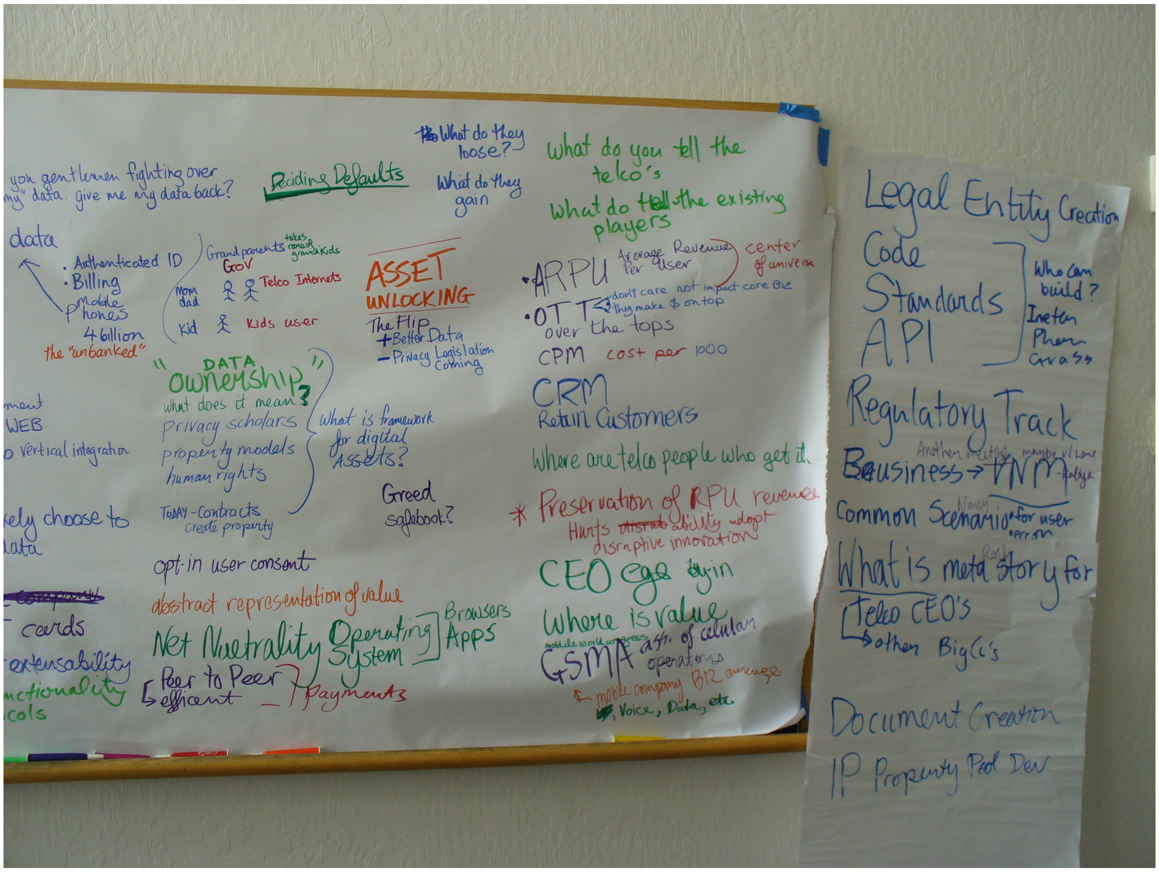
Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

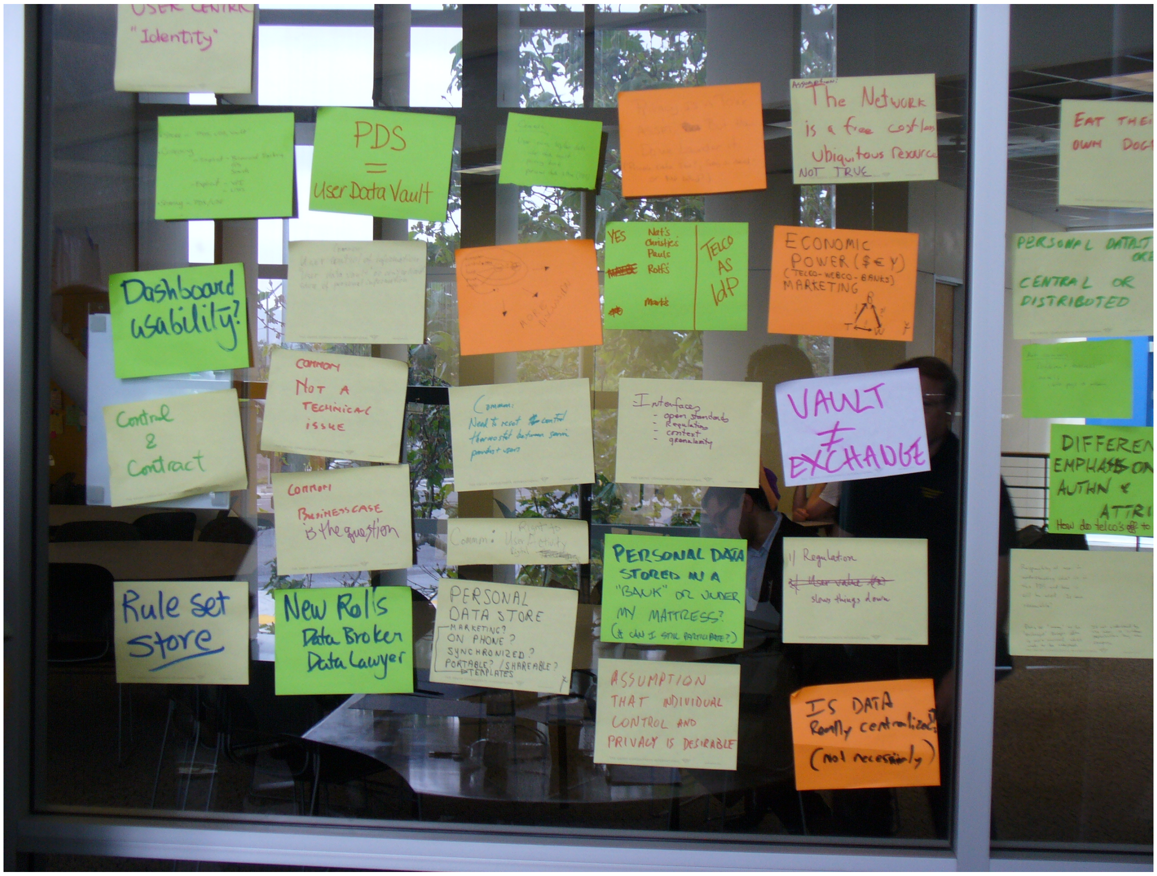


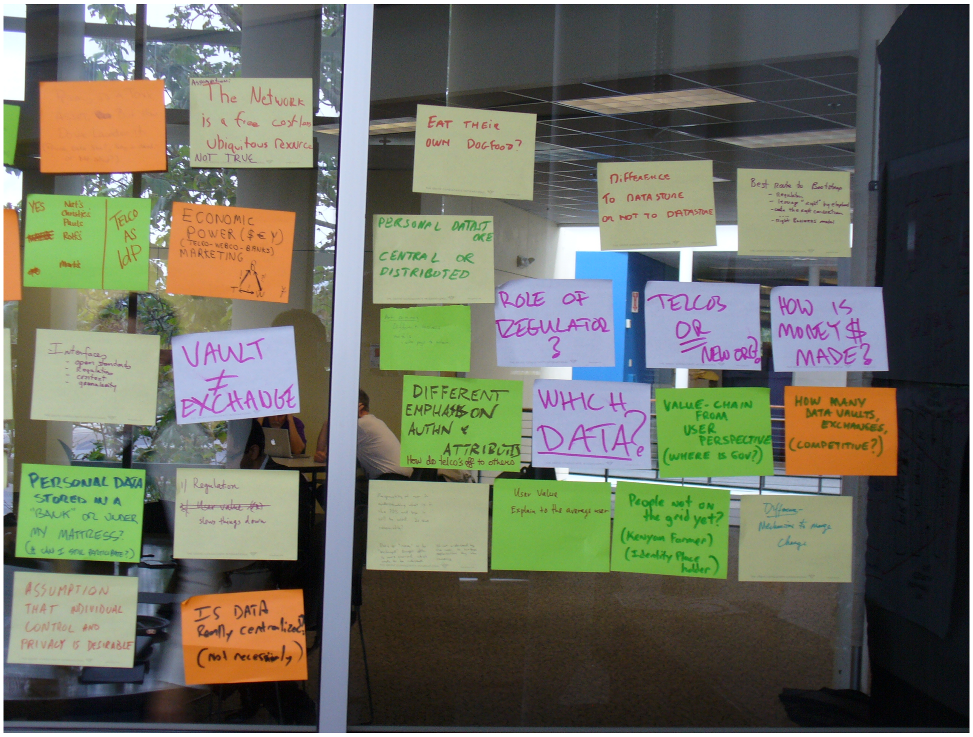

TELCO VIEW OF IDENTITY (as IdP)
WORK
Acct Owner
CHILD
PERSONAL
ACCOUNT-CENTRIC VIEW OF CONSUMERS/IDENTITY
USER-CENTRIC VIEW OF CONSUMERS/ID
Telco R&D Market Evaluation
# Telco Sales
Business Case? pressure/partnership Banks
$
?
Corporate
Business Trust scheme RP's
Consumer Trust scheme RP's
SIM
Cooperate
Telco 1
Telco 2
NEW Cos
UDB + UDE
USER DATA EXCHANGE
FB
Google
Amazon
Govt
Apps
REFINERY
USER DATA VAULT
ID
BROWSER
APPS
OS
INVENTION ARTS
BASIC
WEBSITE
Premium
TRUST SETS
core
Close Friends
Friends
WORK
Contacts
3RD PARTIES
BUSINESS DOMAIN
SAML2
SAML 2
IMS
WEB
SIP
LIBERTY
IMS
INTERNET
TELCO
Value of telco identity
WEB
Nokia Pilot in Argentina
LAP TELCO SIG
BRIDGING IMS AND INTERNET IDENTITY
1 DECEMBER 2009
KANTARA
3rd PARTIES
EXCHANGE
REFINERY
VAULT
REFINERY
ID + KEY
SOURCES
INVENTION ARTS


EcoSystem Articulation
Telco Data Service Management
• Nokia-Simmons
The Telco's
Client side tech Geo
Location services
Difference
mobile payments
• micro payments / macro payments
authentication
• IdM
verification
• Data Vaults
Where are P
Where are you telling story?
Africa – mobile & banking
Where are pilots?
Telco
Banks
Internet Search social
Mary Meeker
Mobile vs Non
See you gentlemen fighting over "my" data. give me my data back?
user data
Computer devices Internet 1 billion
• Authenticated ID
• Billing
Mobile Phones 4 billion
the "unbanked"
Google – Development Mobible ⇒ than WEB
NTT-Dokamo vertical integration
users practively choose to store own data
Replace ~~Phone Company~~
CREDIT cards
Deployment extensability
New functionality w/ protocols
Grandparents takes care of grandkids
Gov
Mom dad
Telco Internets
Kid
Kids user
"DATA ownership"?
what does it mean?
privacy scholars
property models
human rights
TODAY – Contracts create property
What is framework for digital Assets?
opt-in user consent
abstract representation of value
Net Nuetrality
Operating System
Browsers
Apps
Peer to Peer
efficent
Payments
Deciding Defaults
~~The~~ What do they loose?
What do they gain
ASSET UNLOCKING
The Flip
+ Better Data
– Privacy Legislation Coming
Greed
safebook?
*


you gentlemen fighting over "my" data give me my data back?
Deciding Defaults
~~The~~ What do they loose?
What do they gain
what do you tell the telco's
what do ~~tell~~ the existing players
data
Authenticated ID
Billing
mobile phones
4 billion
the "unbanked"
Grandparents takes care of grandkids
Gov
Telco Internets
mom dad
kid
kids user
ASSET UNLOCKING
The Flip
+ Better Data
- Privacy Legislation Coming
ARPU Average Revenue Per User
center of universe
OTT don't care not impact core Biz they make $ on top
over the tops
CPM cost per 1000
"DATA ownership"
what does it mean?
privacy scholars
property models
human rights
What is framework for digital Assets?
TODAY - Contracts create property
Greed
safebook?
CRM
Retain Customers
Where are telco people who get it.
* Preservation of RPU revenue
Hurts ~~disrupt~~ ability adopt disruptive innovation
ment
WEB
o vertical integration
ely choose to
data
opt-in user consent
abstract representation of value
cards
extensability
nctionality
cols
Net Neutrality
Operating System
Browsers
Apps
Peer to Peer
efficient
Payments
CEO ego buyin
Where is value
mobile world congress
GSMA 45% of celular operators
mobile company B2 average
Voice, Data, etc.
Legal Entity Creation
Code
Standards
API
Who can build?
Ineten
Phen
Grass
Regulatory Track
Another meeting maybe w/ Law Polyn
Business → VNM
Common Scenario for user
Never
error
What is meta story for
Role
Telco CEO's
→ other BigCo's
Document Creation
IP Property Pool Dev

what do you tell the telco's
what do ~~tell~~ the existing players
ARPU Average Revenue Per User
center of universe
OTT over the tops
CPM cost per 1000
CRM
Retain Customers
Where are telco people who get it.
* Preservation of RPU revenue
CEO ego buyin
Where is value
GSMA 45% of celular operators
Voice, Data, etc.
Legal Entity Creation
Code
Standards
API
Who can build?
Regulatory Track
Business → VNM
Common Scenario for user
What is meta story for
Telco CEO's
→ other BigCo's
Document Creation
IP Property Pool Dev
FIRE
META Stories
1 → Payments x 4
2 → Social Platform 5
3 → DATA Monetization 6
People
Execs


USER CENTRIC "Identity"
PDS = User Data Vault
Assumption: The Network is a free cost less Ubiquitous resource NOT TRUE
EAT THEIR OWN DOG
Dashboard usability?
YES Nat's Christie's Pauls Rolf's Mark's
TELCO AS IdP
ECONOMIC POWER ($€¥) (TELCO-WEBCO-BANKS) MARKETING
PERSONAL DATA STORE CENTRAL OR DISTRIBUTED
Control & Contract
COMMON NOT A TECHNICAL ISSUE
Common: Need to reset control thermostat between savvy paranoid users
Interfaces - open standards - Regulation - context - granularity
VAULT ≠ EXCHANGE
DIFFERENT EMPHASIS ON AUTHN & ATTRI
How do telco's to
COMMON Business case is the question
Comm: User Activity
PERSONAL DATA STORED IN A "BANK" OR UNDER MY MATTRESS? (& CAN I STILL PARTICIPATE?)
// Regulation ~~User value~~ slows things down
Rule set store
New Rolls Data Broker Data Lawyer
PERSONAL DATA STORE MARKETING? ON PHONE? SYNCHRONIZED? PORTABLE? /SHAREABLE? TEMPLATES
ASSUMPTION THAT INDIVIDUAL CONTROL AND PRIVACY IS DESIRABLE
IS DATA Really centralized? (not necessarily)


Assumption: The Network is a free cost less Ubiquitous resource NOT TRUE
EAT THEIR OWN DOGFOOD?
DIFFERENCE TO DATA STORE OR NOT TO DATASTORE
Best Route to Bootstrap - Regulation - leverage "right" by elephant - make the right connection - Right Business model
YES Nat's Christie's Pauls Rolf's Mark's
TELCO AS IdP
ECONOMIC POWER ($€¥) (TELCO-WEBCO-BANKS) MARKETING
PERSONAL DATA STORE CENTRAL OR DISTRIBUTED
ROLE OF REGULATOR ?
TELCOS OR NEW ORG?
HOW IS MONEY $ MADE?
Interfaces - open standards - Regulation - context - granularity
VAULT ≠ EXCHANGE
DIFFERENT EMPHASIS ON AUTHN & ATTRIBUTES
How do telco's to others
WHICH DATA?
VALUE-CHAIN FROM USER PERSPECTIVE (WHERE IS GOV?)
HOW MANY DATA VAULTS, EXCHANGES, (COMPETITIVE?)
PERSONAL DATA STORED IN A "BANK" OR UNDER MY MATTRESS? (& CAN I STILL PARTICIPATE?)
// Regulation ~~User value~~ slows things down
User Value Explain to the average user
People not on the grid yet? (Kenyan Farmer) (Identity Place holder)
Difference - Mechanism to manage Change
ASSUMPTION THAT INDIVIDUAL CONTROL AND PRIVACY IS DESIRABLE
IS DATA Really centralized? (not necessarily)

XDI & RDF Graph Model (W5A)

URL: http://iiw.idcommons.net/XDI_and_RDF_Graph_Model

Convener: ?
Notes-taker(s): Markus Sabadello, Cameron Hunt

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

I think nobody took notes, because pretty much everybody in the session was somehow involved in holding it :)

Well yes we went over some of the similarities and differences between XDI and RDF. Here is a quick list that is greatly simplified, but maybe it would be useful for the notes:

What the models share:
- Semantic data, organized in triples, forming a graph
- There are "resources" and "literals"
- Various ways of querying and reasoning exist

Where they differ:
- In XDI, there are semantics not only in the graph, but also in the identifiers (e.g. $v $1)
- XDI has XDI messaging, RDF has SPARQL, Linked Data, FOAF+SSL
- XDI has built in access control (link contracts)
- RDF has blank nodes, XDI has inner graphs
- RDF has ontologies, XDI has dictionaries
- RDF can have multiple literals per subject/predicate, XDI only one
- XDI can use persistent identifiers (i-numbers)

Regarding a bijective mapping between XDI and RDF, this is a topic that has haunted Drummond, Paul and me for years.. The short story is that a quick mapping is extremely easy (because in both models you just have triples), but the more "complete" you want the mapping to be, the harder it gets.

Such a mapping is actually implemented in several Higgins components.
See here for a glimpse at this endeavor: http://wiki.eclipse.org/IdAS_XDI_Mapping

Markus

On Jun 1, 2010, at 12:19 PM, Cameron Hunt wrote:

Honestly, I started to take high level notes, but pretty soon I was jumping in (and up - using the white board). The discussion was pretty brief, and really just gave a high over view of XDI, then I responded with how those same things are being addressed by RDF-oriented solutions.

Specifically, while RDF alone certainly doesn't cover the broad capability set provided by XDI the Linked Data/SemWeb crowd is using RDF-oriented methodologies (SPARQL, Linked Data, FOAF+SSL, and even the proposals for RDF2) that claim to address those same sets of capabilities.

And while there are some pretty strong personalities that are difficult to engage, even those personalities are on record as claiming that the principles are more important than the implementation.

I myself (thanks to Randy) learned about the key weakness of FOAF+SSL (it doesn't separate the access token from the authorization token) - but I think there are some folks working on FOAF+SSL in particular and SemWeb/Linked Data in general that might be open to dialogue - I'm thinking in particular of Nathan (http://webr3.org/blog/) who is actively involved in those spaces.

TELCO – WEB – DATA - User Model Scenarios (W5F)
URL:

Convener: Nancy Frishberg
Notes-taker(s): Christie Grabyan

Tags for the session - technology discussed/ideas considered:

We shared ideas about what the scope of the discussion should be:

- User scenarios/use cases specific to the Web/Telco interaction discussed previously
- What scenarios does a personal data store concept enable?
- What use cases are there to drive adoption by users of a personal data store?
- What is the narrative/story between a user and scenario (how many players are there? Is there a specific case to delve into?) An overview where parts can be storyboarded

We chose to focus on the perspective of types of users and their interactions with the web and/or Telco infrastructure. We want to discuss both the macro and micro (global vs. local) interactions.

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Terminology check:
User profile = user persona
Use case = user scenario

Example user scenario: Woman who has a single cell phone in a South American village and she makes a living renting out her cell phone to others in the community to make calls. (We have assumed a one-to-one relationship of person-to-phone). This is not a technical challenge; it is that there is not perceived value by the users to uniquely identify themselves.

Example user scenario: A colleague goes to China and tries to figure out how her product's companies will fit into the Chinese culture and lifestyle. Not everyone (the China user base) had a computer, but pretty much everyone had at least one phone. Many users had many phones. It is perceived at useful for one person to have multiple devices. Often these phones are pay-as-you-go structure. Different phones are used in different contexts.

Example user scenario: We suspect that usage behavior will vary by age group. For example, younger users may not pay for their service (paid for by parents), and they may text much more than they call. Conversely, different phones and plans are marketed towards different groups/peoples.

One application for the personal data concept is that it limits that monopolization of data (by Facebook, etc). But, for people who are not on any social networks, what is their “personal data store”?

Family historians today share family information, but often “offline”. But if this were digitized, there could be more of a need for personal data store for this population. Marketers and advertisers are interested in data like recent browser searches, not always information considered personal, like the family historian artifacts.

Adoption is usually driven by either ease-of-use. People often don’t trust claims of privacy and security.

Not only “what is the killer app to get people on board with personal data store”? But also what is the killer app to get more people to be “social”? The discussion is that everyone is social, but perhaps not digitally social.

A user might not understand the use of a personal data store until they understand what they will gain from it. They need to understand what scenarios will be the reasons they would want to protect and/or share their personal information.

It’s not just about what data to share, but how easy is it for data that already exists about you to be shared back with you. (i.e. the digitization of medical records in the U.S.). There is also the international scenario of people who move countries, and information (residential, health, etc) is essentially lost or no longer usable.

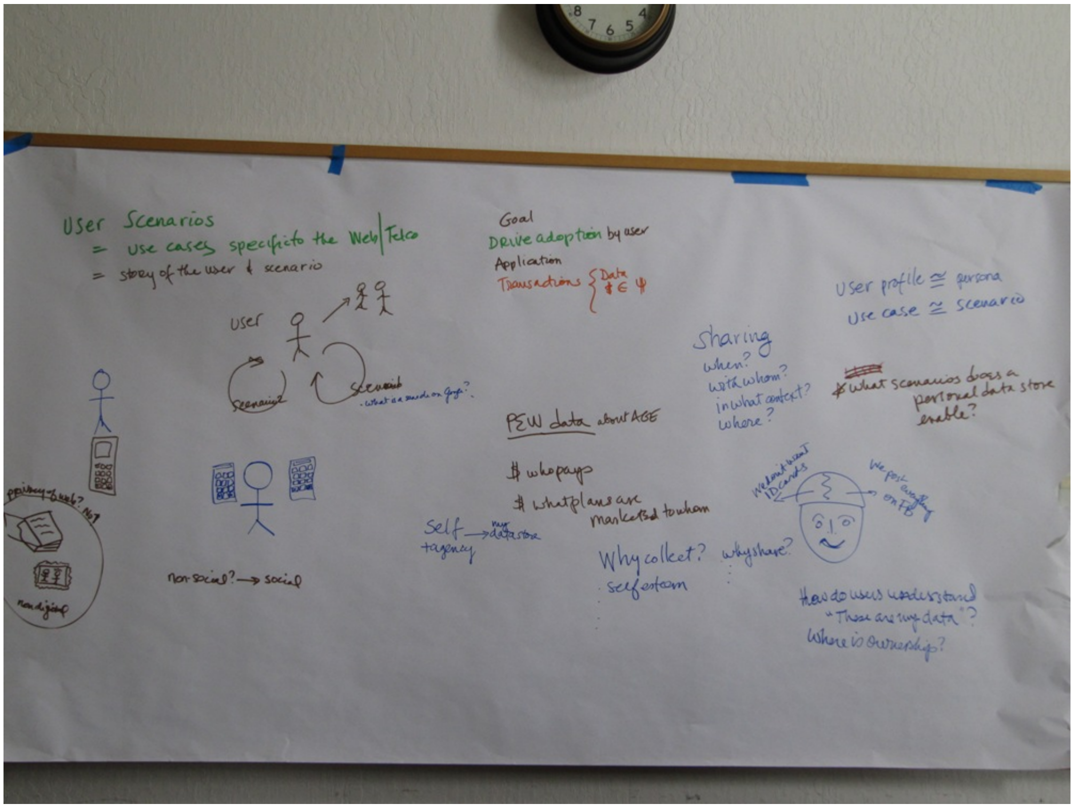
In Singapore, there are national ID cards that are assigned when born and then that number is used on ID cards when you are an adult. There is efficiency in the system, but obviously a lack of user control over your own information or the aggregation of information.

The aggregation of data is the scary part to users, even if it’s the aggregation of data that already exists. It’s the same political issue as the resistance against the government having a national ID card scheme. There will definitely be an education effort for the average consumer to understand what the personal data store means, and why it is necessary, useful, beneficial, etc. The negative incidents that occur are what will give consumers the awareness required to care about these kinds of issues. It’s not that a bank account has to be compromised, but it’s that someone can take and exploit your aggregation of digital data. The emotional impact on the public/ consumer base will drive the adoption of change of behavior.

Scenario: what is your last/recent web searches? How does this interact with or integrate with your personal data store.

If you don’t know what is IN the personal data store, you won’t be in a position to decide what and how to share.

Image of notes board is pasted below:



IIW – What's Next? The Community is Evolving – How Is IIW-11 Better for Noobs & Olds? (W5H)

URL: http://iiw.idcommons.net/IIW_What's_Next%3F

Convener: Kaliya Hamlin
Notes-taker(s):

Tags for the session - technology discussed/ideas considered:

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps: