My name is Henry Story. This is the talk I gave in at the EID conference in Biel, Switzerland on the 27 March 2012, organized by EEMA, the European Association for E-identity and security [1].

Let me first quickly introduce myself. My background is in both philosophy and engineering. I received a BA in analytic philosophy in London and started an MPhil, before moving to computing. From 1996 to 2001 I worked at AltaVista where I developed the Babelfish Machine translation web service. I then later worked for 6 years at Sun Microsystems, where I developed a deeper knowledge of Web Architecture, Protocols and the Semantic Web. During 2011 I was Chair of the W3C WebID Incubator Group as an Apache member.

This talk is licensed under a Create Commons Attribution License, for all except some of the graphics that were not made by me. (Noted in text)

[1] https://www.eema.org/Events/?eventId=5e56c490-c515-44c1-aa9a-480ca0364b4d
This short talk is divided into three parts. First we cover a short history of the evolution of networking technologies from the telephone to the web. Then we look into the recent development of the web of linked data, the WebID protocol, its use in access control, and how this can be used to build a distributed secure social web. Finally we consider how states, institutions, and other companies form a Web of relationships very similar to those found in the distributed social web that can be built with WebID and LinkedData, and we show how this institutional web can be used to create a trust layer useable by commerce that can dramatically increase the security of transactions on the web.
Let us start with the telephone. In itself a telephone – the physical object – is of no value unless it is connected to the network.
Metcalf’s Law:
\[ n(n - 1)/2, \]
which is proportional to \[ n^2 \]

The value of the telephone itself grows as the number of possible connections that can be made with it increases. A network with only one link is only as valuable as the value of a communication between the two nodes can be made to be.

With 5 telephones, the value of the network for each user is the value he puts on communicating remotely with any of 4 other people. The value of the network is the combined number of such connections, namely 10.

With 12 phones the value of the network is 11 for each user and so 66 for the whole network.

This is known as Metcalf’s law: the value of the network grows proportionally to the square of the nodes that can be connected.
This explains the value of standardization of communication protocols and their interconnection to the point that nowadays one can call anyone on earth whatever telecoms provider they are subscribed to.
All one needs to know is telephone number of the person one wished to call: a globally unique identifier for their phone. The telephones need not even be physical ones: the above number can be used contact me on skype for example. In some countries one can even switch operator and keep the same number.
With the advent of the internet e-mail was conceived to allow one to send messages to anyone in the world, whatever their provider.
Again e-mails rely on globally unique identifiers: identifiers of inboxes that are easy to write down which can be used to route mail across the world whatever institution the receiver belongs to.
Finally consider the World Wide Web.
The web is build on the internet, a system that allows anyone to get a computer and tie it into the global network of computers which can communicate through various protocols with servers listening to well known ports. Each of us can get a computer, and name our boxes using DNS (Domain Name System). The name identifies these boxes globally, without needing to know which provider is used to connect the box to the internet, abstracting even its physical location. The computer could be in the cloud, in your basement (of if you are Swiss in your bunker), in a pocket sized personal computer devices, in your cell phone, .... whatever. We don’t care. We can just draw them as boxes.
Each of us already has such a box at home that enables our computers, be they laptops or other devices to connect to the internet. Each of these boxes is given an ip address, and could easily be given a domain name too.
Eben Moglen, author of the GPL, the license that the Linux operating system is based on – the Operating System that runs all of Google’s server – Eben Moglen called over a year ago for the development by open source community of a freedom box, that could act as a home gateway device, hosting a number of services – such as e-mail, a web server, voice over-ip, and one’s own node in what will be the distributed social web. The aim of the project, already well underway, is to create a device that anyone could use as their personal communication server, which would store information the user would be fully in control of, and which could only be inspected against the user’s will if required to by a valid court order.
This would allow each of us to have a device with our own personal web server, place documents on those servers, and thereby allow anyone to access them as web page with a globally unique name, a URL such as mine http://bblfish.net/ for example.

Each page on these web servers gets its own name, enabling them then to be linked from html pages on other servers, sent by e-mail, or even printed on advertising bill boards on the side of busses.
Once we have servers containing documents identified with URLs, it is easy to create global identifiers (URIs) for people and other objects. The URI specification states that the meaning of URLs with #fragments is given by the document without the #fragment.
The fragment identifier component of a URI allows indirect identification of a secondary resource by reference to a primary resource and additional identifying information. The identified secondary resource may be some portion or subset of the primary resource, some view on representations of the primary resource, or some other resource defined or described by those representations. A fragment identifier component is indicated by the presence of a number sign (") character and terminated by the end of the URI.

As stated by RFC 3986 “Uniform Resource Identifier (URI): Generic Syntax” in section 3.5
The document gives the sense (meaning) of the URI, i.e., the method to identify the object referred to by the full URI. In particular, we can tie the meaning of a URI identifying a person—known as a WebID— to a public key. The document can then describe the referent of the WebID as being the person who can prove that they have access to the private key corresponding to the given public key (illustrated by the lock in the document on the server in the illustration here). The sense provides then a proof procedure for identifying the agent.
Once we can identify people using global names, we can then also create distributed social networks, using LinkedData served over http or https, using semantic web standards defined at the W3C. Such social networks have already been used, and millions of such interlinked profiles for users already exists. With a global authentication system such as WebID access to pages can be controlled, making it easier both for people to participate in social projects such as wikis, by limiting the possibility of spam, and also by enabling architectural privacy: the user can place the data on his server and only give access to it to those who need to know it, in full confidence that no body else can access that information.
Of course placing web servers on small devices does not exclude placing them on very large computers to satisfy the need of larger agents, such as banks, book stores, states, universities, transportation providers etc...
It suffices then to notice that such large agents are not fundamentally different to their smaller counterparts. States form social networks among states, which they recognize in various ways through bilateral agreements as peers. States recognize entities such as banks, businesses, universities, stock exchanges, ... each of which is then bound to the laws of the land and which citizens can use knowing that they will be protected by its legal system. Banks for example form networks with businesses, businesses network with their providers, their outlets, their customers, as well as with legal, state institutions and universities. Some of this information will be public, some private. Creating such a formal linked data network is technically very easy. It just requires placing RDF documents on web servers protected by TLS (https).
Distributed Social Networks require privacy and security. The WebID Protocol enables global distributed authentication. It works in all existing desktop browsers and more and more cell phones, given that it builds on the widely adopted TLS/SSL protocols that power all commerce. The following 10 minute video http://bblfish.net/blog/2011/05/25/ gives an overview of how the protocol functions technically as well as from the user’s perspective. The spec is available at http://webid.info/spec/. I recommend that you watch that video at this point, as it shows some screen casts revealing just how easy it is to create a WebID and how easy to use and how useful it is.

In summary one can create an X509 certificate in one click in a browser, and then authenticate on any web site securely in one click, without having to type an identifier. It also shows how easy growing a social web is.
Up to now the WebID Community (http://www.w3.org/community/webid/) has only explored placing a WebID in the Subject Alternative Name of an X509 Certificate. But the X509 standard in addition to a SAN slot, provides for an Issuer Alternative Name slot (highlighted here in yellow), which can also take a number of URIs. This would for example allow larger organizations to give out certificates for their members, and identify themselves as the issuers of those certificates. Universities could generate certificates for their students, Banks for the accounts of their clients, business for their employees and clubs for their members. Such a certificate can be created with a simple command line as shown above using Java’s keytool, which creates a hypothetical WebID for user of UBS, the Union Bank of Switzerland. For the end user of course, given the proper backend, this can be simplified down to clicking a button on a web page using HTML5s keygen tag.
So if government web sites (eg companies.gov.us here) published a list of all the trusted banks, identifying them using the bank’s WebID, or by linking to equivalent lists in foreign countries (such as firmen.de), then a commercial entity (shop.com in the diagram) could look up such lists on a regular basis and build a database of officially useable banks wherever they were in the world.

The bank could publish on its servers its own WebID Profile document, declaring its public key and other meta data, such as the current CEO, and board members. It could then also publish there the profile document for each account of its clients. Here we show the account numbered 234ad242 with a link to a payment form, and minimal personal information.

The account owner would then be able to authenticate to a commercial web site, such as shop.com, with its bank account WebID in one click as shown in the video mentioned previously. The shop he was connecting to would then both verify his identity, and verify the ID of the issuing authority the bank, and from the shopper’s account it could find the payment form by looking up the profile on the users’ web site, and in some way still to be explored make the transaction using a simple RESTful service. The bank itself could identify the shop by looking up the shop’s WebID in a list of commercial sites in a manner very similar to the way we described this above. In short the protocol is recursive.
Imagine then that Bob, who has an account with bank.com, authenticates to Shop.com with his Bank account certificate in order to make a purchase in step 3 of the diagram above. The WebID verification proceeds exactly as described by the WebID spec in 4. The Issuer Alternative Name of the certificate is dereferences in 5, and the returned public key is compared with the one sent in the certificate. But in addition to verifying the WebID of the user, shop.com can verify that the certificate is correctly signed by the Issuer (bank.com), by checking that the public key found at the Issuer’s WebID did the signing (point 6–7 in the diagram). This would rightly be called the Authentication part, that is the verification of the AUTHORSHIP of the certificate.

Having done this the shop only knows that the Issuer signed the certificate and that the WebID is valid. In order to TRUST the bank as a legally valid institution it can the look up the bank’s WebID in a list that is regularly updated using the Institutional linked data web at (step 8). It would then know precisely what legal status the bank had, which legal system it was bound to, and so whether it could do business with it.
Because WebID uses only well established standards it can work with existing hardware security tokens such as the Swiss ID pictured here. Swiss ID uses X509 Certificates which could easily be enhanced with Subject and Issuer Alternative Name URIs, when coined, with no other changes elsewhere. The private key located on this card cannot be accessed by a software virus on the computer, and cannot be copied. To get the private key the card itself would have to be stolen, making its physical absence visible.
Things will only be improving in the space of TLS security. There are a number of shortcomings of the current Certificate Authority Model, that will be fixed by the deployment of DANE which uses DNS-SEC, the secured DNS, to publish the public key for each secured service (domain:port pair). This will remove a whole level of trust issues of corrupt or flaky CAs, make it much easier for institutions to rapidly deploy services by publishing the self signed certificates in the DNS, removing the problem of browsers or operating systems having differing lists of trusted CAs, and much more....
Also worth noting is the development of the SPDY protocol, an improvement to HTTP transport developed by Google and implemented in recent versions of Chrome and Firefox, that is already being used by Twitter, Google, and that is finding its way into web servers such as netty, the core of JBoss. SPDY works by default with HTTPS, and is aimed to massively reduce the number of open connections required to run a dynamic web site. This is clearly going to help TLS gain a lot wider adoption.
Identity is social.

http://webid.info/

bblfish.net

In conclusion: Identity is Social. Just like the telephone the individual, whether it be a person or an organisation, is inextricably a part of a web of relations, that are in very important respects social. Organisations can declare their trust relations, the legal frameworks they are accountable to, their partners, their distributors, and indeed even their own employees. Similarly countries can list the organisations they recognise directly and indirectly creating official trust networks backed by legal systems. By publishing such information as linked data governments could enhance the trust users have in the web sites they visit, allowing smaller companies to participate in the cybernetic revolution, decentralise banking, and commerce, and increase the overall security of the web in a flexible manner allowing different countries to declare their trust preferences. More information can be found on my home page at bblfish.net, and pointers to recent webid developments at webid.info.

Thanks for listening.