For a New Internet of Subjects Manifesto

The objective of the first Internet of Subjects Manifesto was to explore how we can create an **Internet of Subjects** tailored to the needs of emancipated, self- and socially-aware individuals. This document is an attempt to carry further the reflection and is an invitation to the writing of a new Internet Subjects Manifesto.

Why do we need digital integrity?

The influx of digital technologies and online services is leading to an ever-increasing fragmentation of personal data circulating over the Internet. This fragmentation, which is a consequence to today's Internet architecture, is detrimental to our *digital integrity* and therefore our ability to establish trustworthy relationships on the Internet.

Providing true *digital integrity* requires an architecture where people, and organisations, are fully empowered to create and manage tangible representations of themselves, control these representations while interacting with other people and organisations.

Unfortunately, most current initiatives addressing digital integrity can be classified as *first order changes*: they are attempts to adapt an architecture primarily designed to manage documents[1] at a time where bandwidth, storage and computing power were scarce and expensive.

Moving away from a *document-centric* towards a *person-centric* Internet can not be achieved by simply 'patching' the current architecture: this requires a second order change, a Copernican revolution.

What are the limits to identity representation on the Internet?

One of the main limits of today's identity representation on the Internet is that, while a number of solutions address the issue of *identification* ***of*** (*who are you?*) there is a wide gap to address the issue of *identification* ***to*** (*who am I in relation to others?*).

Identity Access Management (IAM) technologies were developed to authenticate users to grant or deny access rights to data and system resources. They are focused on:

- *authentication*: the identification of a person or service in order to attribute relevant access rights (Who is the user? Is the user really who he/she represents himself to be?)
- *authorisation*: the enforcement of access rights in relation to the user (Is user X authorized to perform operation P on resource R?)

---

[1] The main protocol used on the Internet is HTTP, Hypertext Transfer Protocol, which is also used to transport personal data...

The development of IAM technologies has had major consequences on the common understanding of identity that ranges between:

- *identity* reduced to a mere *identifier*, like *identity cards*[2]
- *identity as a* set of attributes that could be isolated from the rest of the world by high and thick *privacy walls*, like in *personal lockers* or *personal data stores*.

Unfortunately, like the story of the tail wagging the dog, IAM technologies have created a distorted view on identity that has now become central to many efforts related to privacy and trust.

Identity is at the same time self-identity (Giddens) and identity through others (Laing), a self-narrative that explains the past and informs the future, the invention of self (Kaufmann) and the outcomes of interaction with others.

Far from being limited to any sets of attributes stored in some kinds of personal data stores, any identity is distributed over a network of relationships with other individuals, organisations, businesses, ideas, values etc. Our identities are distributed across places, some under our own control, many under the control of others: public authorities (e.g. identity card provider), businesses (e.g. credit card provider, university as diploma provider), colleagues (testimony providers), clients (testimonial providers) or foes (trouble providers).

It is in the nature of identity to be *distributed* —which is very different from being *fragmented*, as a result of an obsolete architecture.

The distributed and social nature of identity means that it is not possible to *locate* someone's identity within a particular location, and therefore to *protect* it behind high and thick privacy walls. This does not mean that we cannot protect our identities, but that in order to be able protect them we need exist as *autonomous empowered entities*, have the means to defend ourselves, as individuals and in association with others to defend the common goods that are part of our collective identities.

It is now time to design and implement an architecture that fully embraces what we know about identity.

What are the limits to digital integrity?

In the recent developments relative to trust and privacy technologies, one of the goals was to provide people with the means to protect their personal data: *personal data is mine!* was the motto. This has naturally led to the idea of personal lockers and personal data stores: if all of my data is in my personal locker, then I can decide who has access to it and under what conditions (for how long, to do what, etc.). Personal Data Stores rapidly became the Holy Grail for the most advanced actors in the field of personal data management, from ePortfolios to personal health records. Personal lockers and personal data stores helped us to understand that an Internet based on a clear separation between storage of personal data and services creating/exploiting it would revolutionise the Internet. Empowered users would be at the centre of an ecosystem they control. "The Semantic Web & THE POWER OF PULL", by David Siegel, admirably describes the transformations one should expect from the systematic use of personal lockers.

However radical and transformative, personal lockers and personal data stores have their limits. One is to be found in the initial statement: *personal data is mine!* Data, the product of social interaction and processes, is generally shared with other people and organisations: I share the name of my parents, the review of a paper submitted with

---

[2] c.f. OpenID, Opencard etc.

reviewers and conference organisers, the diagnosis of my illness with a doctor, a laboratory and a drugstore. Even my intimate thoughts can be shared when I commit a freudian slip... If most data is shared with others then we might want to rewrite the initial statement with: *personal data is ours!* Translation into technology of this statement might lead to something radically different from personal data stores as *personal information silos*.

Moreover, moving storage of personal data from silos controlled by online services (National Health Service, Monster, Linkedin, Facebook, Sony, etc.) to personal data store hosting services only provides an arithmetic advantage in terms of personal data protection: hackers will still be able to find a way through walls, whatever their height and thickness. And whatever technology is being used to minimise correlation, like onion routing, anonymisation and encryption, the simple fact that a series of personal data is stored at the same location and shared with others from the same location is a privacy threat.

What are the limits to existing trust technologies?

Are there architectures that are more conductive to trust than others? Is *trust* a natural property of an architecture or something that can be added to it? Is it possible to patch an architecture that is not trustworthy by design, e.g. today's Internet, to make it trustworthy?

One of the most common misconception with *trust* is that it is derived from *security*. In fact:

- Trust is about *relationships* with other Agents while security is about the *infrastructure* on top of which relationships can being established.
- There is not need to have a system 100% secure to establish trustworthy relationships while, conversely,
- If security of a system is too invasive, trust can be impossible to establish or it could even be destroyed.

Trust implies the acceptance of a certain level of risks (*I trust you to do that, but if you fail I'll have to accept that I might have made a bad choice*), while security aims at limiting risks and their consequences (*you are contracted to do that, and if you fail you will have to pay penalties*). Trust might contains elements of *faith,* while security generally rests on a *contractual* element.

The information one needs to establish trust and the rationale on which trust decisions are established are fully idiosyncratic to individuals, organisations, the communities they belong to, history etc. And it is most likely that most individuals are not able, nor willing, to make this information and rationale explicit, even less in a format that could be exploited by a computer programme or a *trust engine*.

What solutions to identity representation and trust?

If one's identity cannot be located in a personal locker because it is distributed as a result of social interaction, if personal lockers can become a security threat, what are the solutions for creating a natively trustworthy Internet?

Wouldn't it be wonderful if we were able to exploit the natural property of personal data and identities as a connection to people (places, ideas etc.) while preserving the need for privacy, anonymity and enabling trust?

Personal Anonymous Data Stores

Imagine that, instead of storing our personal data in personal lockers, we store it in Public Anonymous Data Stores (PADS). When I store a piece of data in a PADS, I receive in exchange a key that allows me to edit the data and open my mailbox, so if someone wants to contact me, or if my profile matches a search engine query, a message is left in the box connected to this anonymous piece of data.

My data can be distributed over an unlimited number of PADS and I'm the only one to know that it is my data. I am the only one able to connect my identities to any piece of data stored by me. For the rest of the world, my data is just a drop in an ocean of anonymous data. No need for fancies technologies to protect it —*why break in safes if money grows on trees in public parks?*

To create a trustworthy Internet respectful of privacy, shouldn't we simply make our personal (meta) data public?

What initially sounds like a paradox might become a founding principle of a trustworthy Internet that could announce the demise of the business models on which Linkedin, Facebook and Google are built...

Of course, PADS are mostly relevant to metadata and data than do not provide direct connection to our identities, unlike some photographies, pay-slips or a bank statements that could be stored personal data store. It should also be possible to have PADS with different policies defining the 'public' that have access to it. In order to make it possible, we need to add another piece to our trust puzzle: *Personal Agents* of *Personal Proxies*.

Personal Agents or Proxies

Imagine that, instead of existing on the Internet as simple *users* placed at the periphery of the system, behind a browser while being online, we were able to have a tangible representation of ourselves through an *Agent* or *Proxy* that would stay active on the Internet even when we are off-line. Imagine now that every entity, person, network or organisation, is represented by such an Agent[3] and that interaction with other people, organisations, services is operated through their own Agents. Imagine that it is possible to create as many Agents we want, and when they are created they join the society of Agents, just like a new Internet site joins the other websites (or a DNS joins the community of existing Domain names Servers).

We would therefore be able to establish *a society of Agents* that could be the foundation for a fully trustworthy Internet.

Agents could grow through different maturity levels:

1. Gateway: single access point to the Internet
2. Aggregation: unified access to fragmented data
3. Integration: unified transactions

At level one, Agents provide their owners with a single access point, a gateway to the Internet, whatever machines, systems or applications they use. For service providers, there is no functional difference between a client using a proxy or a browser. For a person using 3 different computers and 2 different phones, she does not have to install 5 different firewalls on 5 machines, but only one, on the Agent through which all her communications

[3] One person could have multiple Agents to manage their identities through the connection with other Agents.

go through. If she has more that one Agent, which should be the general case, then she installs one on each of them.

At level two, Agents are mainly used to provide a single interface, a kind of dashboard, to provide and control access to currently fragmented data. Agents act like aggregators through which their owners are able to share and update data they already have on Facebook, Linkedin, National Health Service or stored in PADS. Agents provide a single entry point, so requesters of personal data do not have to bother with the idiosyncrasies of every system where it is being stored. If I want to share a piece of data I have on Facebook, I go and fetch it myself to give it to the person or service asking for it. I can also decide to write that piece of data to a PADS to make it accessible to all, while preserving anonymity.

While at level one and two communications remain asymmetrical (Agents communicate with existing services directly) at level three communications are fully symmetrical: all communications go through Agents. This brings a number of qualitative changes leading to the possibility of establishing trustworthy relationships.

Each Agent has a profile of its own: date of birth (creation), connections to other Agents, history of activities, metadata created by themselves and by others that are stored in PADS, etc. These ever changing and maturing profiles create the conditions for a fluid, real time computable identity based on information provided by the Agent and all those it is connected to. Identity can be computed in real time based on information distributed across Agents.

To the difference from what already exists, like eBay reputation engine, Agents are fully independent from any kind of service. It is therefore possible to generate many different reputation indicators, based on the aggregation of many different data using many different algorithms.

How is it possible to know whether a vendor is really a good vendor when vendors and buyers are not all registered on the same service? More generally, how can we be sure that, like too many hotel and restaurant reviews, that reputations are not faked by the vendor himself, friends or accomplices[4]? In other words, how can we correlate data, in order to have some kind of indication about the trustworthiness of a statement used to create reputation data?

Within a *society of Agents,* when an Agent makes a statement about another Agent, a reputation engine could verify the reputation of the Agent making the statement: have they been in relation in the past (customer)?, how close are they (friend, family)? etc.

Let say that after a stay at a hotel, I write a bad review and I store it somewhere in a PADS. An *hotel-reputation-engine* would not be able do much with it as it is written in plain English and might have been produced by anyone. Now, let say that each time I use my credit card to buy a good or a service, the credit card company invites me to rate the transaction. I can ask the credit card company to write the statement in a PADS for me, so I remain anonymous, while I use the credit-card company as a trustworthy witness of the transaction. With this witness testimony the *hotel-reputation-engine* now knows that this statement is the result of an actual transaction. Another way to proceed and eliminate the need to the *hotel-reputation-engine* to know that the testimony comes from a specific credit card company, it is the PODS itself that computes a trustworthiness indicator and stores it.

---

[4] Let's say that you want to build a good reputation for cheap: put goods for sale, create fake identities that buy your goods with your own credit card. For an external eye, it will look like a number of satisfied clients and it will have costed you the commissions paid to eBay and the Credit card company...

Now, let say that I want to share my experience with hotels and restaurants with others. I create an aggregation with all my reviews and decide to keep it anonymous, or not. In order to confirm that they are my reviews and not others', the aggregator simply checks that I have the corresponding keys to all the data aggregated.

The *hotel-reputation-engine* would then be able to use this aggregation to moderate my reviews (I could have a tendency to score higher or lower than the average punter) to generate an indicator.

Of course, there are many other ways the same scenario could be implemented. The description provided is just here to show demonstrate the power of the combination of Agents and PODS.

Towards the creation of a holographic social memory

Imagine that all the events in our life were generating universal unique identifiers (UUIDs) and that we were able to attach the UUID generated for a specific event to all the people, places, ideas etc. associated to this event. For example, when someone is born, the UUID generated at the time of birth is associated to the parents, midwife, location, time etc. Now imagine that one continue to record all the events associated to one's life by simply 'tagging' all the things associated to those events. By simply using the history of UUIDs it would be possible to recreate the film of one's life from many different perspective: places, people, ideas, etc. Now imagine that we were able to combine different individual movies into one, the result would be a collective movie.

Like the holopixel containing information about the whole image from its own unique perspective, the representation of one's identity contains information about the whole society. Each individual records our Agents store in PODS tell something about us, others and the society as a whole.

Let call 'holonoma' (from the Greek ὅλος, "whole" + ὀνομα, "name") such an architecture which represents the collective memory of society and the associated identities holoIDs (HID). Individuals have full control over their HID, which is computed in real time through their Agents. Different Agents would deal with the different identities a person have, like citizen, parent, student, patient, etc.

This collective holographic memory, co-constructed by Agents co-operating in a trust network would would not only improve the outcomes of search engines (accuracy), but empower individuals to decide whether or not to be discovered[5]. *Search engines* might become *discovery engines*.

From search to discovery

Search engines have transformed the way we work and think and developments on semantic web and semantic search engines are bringing us one step further. On the other hand, search engines are extremely inefficient with personal information:

- they cannot respond to simple queries like: *is there a Java programmer that lives in South Australia and is fluent in Japanese? How many people have bought a TV set last week? How many people are looking for buying a new car over the next semester?*

---

[5] If someone is looking for how many people match a specific profile, and if there are 100 matching profiles, the search engine might return "10" if 90 people do not want to appear in the results. The results to the same query could vary, depending on who initiates the query.

- they facilitate public access to data that should have remained private: hire and fire decisions have been made by employers based on data found on Facebook

Early attempts to address this issue comes from the reflection on Vendor Relationship Management (VRM) a reality. The goal of VRM is to provide customers with the means to be fully independent from vendors and better means for engaging with vendors. In its description of ProjectVRM, the Berkman Center says "*The primary theory behind ProjectVRM is that many market problems (including the widespread belief that customer lock-in is a 'best practice') can only be solved from the customer side: by making the customer a fully-empowered actor in the marketplace, rather than one whose power in many cases is dependent on exclusive relationships with vendors, by coerced agreement provided entirely by those vendors.*"

While the classical approach to CRM (Customer Relationship Management) created data silos (one per service provider) requiring important resources to keep it updated, VRM believes that it is for the client to manage his/her own information. While CRM systems keep each client within separate records of a database, VRM allow clients to meet and coalesce in order to negotiate discounts for grouped sales. While CRM data is used by one organisation, VRM data can be used by many different organisations, in particular competitors. A typical example would be fidelity card data from different providers that would be stored in a space controlled by the client and could be shared, or not, with every service providers.

One of the limits of VRM is to be found in the name itself: if it is a *vendor* relationship management, then what about someone who is at the same time a client and a vendor, something usual on auction sites? Do we need 2 different services, one as client, one as vendor? One way to address this question is to replace *Vendor* with *Trust*, so VRM becomes *Trust Relationship Management*. TRM is by nature symmetrical, so it should work just as well for vendors and clients.

The other limit of VRM (rechristened TRM) is one encountered earlier with personal data stores: moving the control of data records from CRM to VRM systems might not be sufficient, or necessary, to establish trust relationships.

To understand the issue, let say that we want to *find all the people that have bought a TV last week in France*. We will explore two different methods:

- **search**: find data sources, crawl them to find a match and display the result. It is asymmetrical: on the one hand, active search engine, on the other, passive data records.
- **discovery**: crawl data sources and leave a search notification when there is a match; matching parties are notified, and decide whether or not to be part of the query results. It is symmetrical: results are the outcome of negotiation among free Agents.

With a search engine, to *find all the people that have bought a TV last week in France* one could either query all French TV vendors (some might not want to give access to that information or people might have bought TVs from another country), or search VRM records of all French citizens.

With a discovery engine, the agent initiating the search notifies a trusted crawler to explore all PODS and leave a search notification in all the records containing "France" in an address field, in all the records containing "TV" in the "owner of" field and in all the records with a field "start of warranty" containing a date less old than a week. At this stage of the process, the agent initiating the search has no response to its query.

As Agents regularly poll the places where personal data is stored anonymously, they will soon discover that another agent has left a search notification. Contained in the search notification is the query, so the Agents will only respond if all criteria are being matched. If the target Agents accept to respond to the query, they respond the the discovery engine, which in turns consolidate and present the results to the Agent that launched the query.

With the query results is an handle that the Agent cant use to send a message directly to the other agents, while remaining anonymous.

Of course, as there are millions of queries running concurrently, the mechanism described above would require some adjustments in order to reduce the payload on servers and networks. What is important to see at this stage is how an architecture based on PODS and Agents empower individuals and active participants in query mechanisms.

And it will be possible to respond to the question: *is there a Java programmer that lives in South Australia, is fluent in Japanese and is allergic to sushis?* if there is a match and if the target trusts the person who made the query. A discovery engine will not be able to tell whether a profile does not exist, but if if exists it will provide the owner of the profile to be discovered.

And now?

Technologies to implement PODS and Agents already exist. PODS could easily be created using some kind of wiki and Agents starting from proxy software. We could use more sophisticated or more performant solutions, but that does not really matter as the system is obsolescence-proof: data can migrate to better PODS later and we should be able to switch to new Agents technology without too much trouble or efforts.

Once the initial elements of the architecture in place, services can start to grow on initial data and add more data to PODS.

When every entity interacting on the Internet does so through an Agent (personal or organisational), we could then implement global policies enforcing basic social behaviour, like not spamming: in order to send a message to 1 million Agents, we could have a rule such as the requirement to be in existence for more than 3 years. So if a new bio-engineering company discovers a new molecule and, through metadata harvesting, finds out that there are potentially one million people that could benefit from it, it would not be allowed to send a message to more than 100 (unknown) people and in order to reach the million found (but not identified!), it would have to go through a patients' association, a national health service or some other trusted organisation.

People will have multiple agents, share metadata across them, transfer them one to another, providing individuals the ability to reinvent themselves. In order to prove an employer that one has a diploma through an anonymous 'job finding proxy' (ethnic discrimination is not anecdotal employment practice in France) the Agent of the employer simply checks with the Agent of the University that the statement is genuine -e.g. the university Agent simply checks wether the diploma database knows the Agent of the applicant, so there is no need to reveal names.

While those Agents would not require any kind of Identity provider to exist (they join "the society of Agents," just like a new DNS joins the network of Domain Name Servers), they would have a strong identity: date of creation, social networks, history of activities, etc. It is this 'strong anonymous identity' that would create the base for massively, anonymous, meaningful and trusted interaction.

As suggested with the maturity levels the architecture can grow organically with the actors contributing to its construction. The architecture makes it possible for personal data to migrate progressively from todays' silos to PODS, for services to progressively accept to deal with personal Agents, then operate through their own Agents.

Why not start today?