# Do we really need identity propagation in SOA and Clouds?

by Michael Poulin

## Problem Statement

IT developers and, especially IT Security specialists struggled for years trying to create an identity control at the enterprise level. The most known initiative and model in this area is Single Sign On (SSO) where an identity of an end-user can be propagated between systems and can be recognized across an enterprise.

In today IT reality, security has to deal with service-oriented environment and Cloud Computing (Clouds for short). Many corporate security specialists and vendors concern with how to construct SSO for services and Clouds. However, a simple question - “Is it feasible having SSO in SOA and Clouds from a business perspective?” – is still to be answered. Before developing such SSO, we have to understand that the security execution context for identity controls had changed when we stepped into so-called Service-Oriented Ecosystem (SO Ecosystem) and Clouds.

## Constraints of Service Ownership

A notion of service ownership is not well-known in IT but it is not a stranger in Business. OASIS Reference Architecture Foundation (SOA-RAF) [1] defines SO Ecosystem as positioned between Business and Technology, and within each one of them. Following SOA-RAF, we will explain service ownership boundaries and consequences for aspects of authentication and authorization for services. In Clouds, both aspects are affected even more due to financial relationships and constraints between Clouds providing businesses.

Creating an SSO in an enterprise is already a challenge because of a mixture of old and new systems, applications with embedded security and with no security at all. This is why security specialists are so proud if they finally build SSO solutions for their companies. Unfortunately, this variety of software is not the only characteristic of security context in an enterprise. Another characteristic is that the majority of contemporary enterprises has a single IT department, i.e. security specialists are not only responsible to all business units (BU) for the safety of the information systems but also have unconstrained access and the right to work with every system in IT. This is called a single security ownership or a single security realm. This is the key for managing authentication and authorization in particular centralized or decentralized manner depending on the industry, company business, regulations and investments in security.

### Service Ownership Boundaries

SOA-RAF states that in SO Ecosystem, each service may have its owner, not only provider, with its own rights and policies. Since the most important services in the enterprise are Business Services, they usually have business owners such as BU, groups or teams, departments or line of business (LOB) outside of IT authority. IT in this case is only a development and maintenance arm, not an owner. And this affects security solutions a great deal.

SO Ecosystem comprises multiple security realms that are formed in line with business ownership and related boundaries. One realm may cover one or many business ownerships, which recognise the realm’s Authentication Authority (regardless implementation). Also, there may be one security realm for one business or even service ownership.

A concept of business service ownership may be summarized via “Knight Rules of SOA Ownership” [2]:

* **A Service of my Service, is not my Service**
* **A Consumer of my Consumer, is not my Consumer**
* **A Partner of my Partner, is not my Partner**
* **A Supplier of my Supplier, is not my Supplier**

In other words, services or consumers, which are the roles of the same entity in SO Ecosystem, have knowledge and visibility in only their immediate contacts; they do not care about how their contacts operate and who they serve in their business transactions.

### Business Ownership Boundaries

The problem with business ownership is that its boundaries are not transparent to the technical solutions including services and related security controls. Within a security realm, participants may agree to recognise an identity of others and the whole system works like in any enterprise with SSO. Equally, participants may disagree to open their boundaries. Every business/service ownership is totally independent from another one (otherwise, it is not an ownership). This means that even security specialists have to obtain permission from the owners if security related changes have to be made to the services.

Furthermore, IT cannot simply enforce using specialized Security Services like Policy Enforcement, Authentication Service, Authorization Service, Identity Management Service and alike unless business services and their consumers are in an agreement. The latter exists in a form of a Service Contract, defined in SOA-RAF. Actually, a chain of Service Contract replaces centralised authentication and authorization.

### Service Contracts and Financial Boundaries

Service contract includes service interfaces, interaction and execution policies, special functional agreements, communication channels, expected results and SLAs, as a minimum. While service contracts are the foundation on which consumer-service relationship and interactions are based, in Clouds it plays an additional special role.

Service Contracts connect business entities in the Clouds and demonstrate different interests and objectives of these entities. Even if inside an enterprise it is possible to agree on sharing identity tokens across service ownership boundaries (because of the authority of the enterprise itself), in Clouds financial boundaries between different independent businesses would not trust foreign identities (as an adoption PKI has demonstrated) and would not allow dealing with external unknown consumers.

Each Cloud Provider preserves its financial interests above all. Financial feasibility in Clouds is more important than a sensible technical solution. For instance, if a Cloud Provider maintains an Authentication Authority for its consumers, why it would agree that external consumers would pay security fees to an external Authentication Authority? Why those consumers would not pay to this Cloud Provider instead? It is a matter of business tactic, not technology, whether to lose security fee but charge for the service or find other consumers who ready to pay both charges.

### Propagate or not Propagate?

The only overarching conclusions we can draw from the reality of ownership and commercial boundaries is that a propagation of identity in the chain of interacting entities in SOA and Clouds makes sense only between directly communicating consumers and service; further propagation is useless because service in different security realms would not know what to do with a foreign identity of an end-user. Recall that a service does not have a Service Contract with consumers (end-users) of its own consumer.

We can hear arguments about a need of an audit along business transaction passing through several services. This is a right requirement if services belong to the same Authentication Authority control. If not, no audit exists above these services and this is the core difference from what IT security specialists used to think of. Each service deals with external audit by itself as an independent business entity (within own business boundaries). Also, there is no a single business transaction crossing boundaries; instead we can talk about a chain of multiple transactions in this case. SOA-RAF says, “*The OASIS Reference Architecture Foundation for SOA (this document) is … used for understanding different elements of SOA… and considerations for reaching across ownership boundaries where there is no single authoritative entity for SOA and SOA governance.”*

Thus, each independent business service has to care about its own immediate consumers and providers. It does not need any visibility onto management of other services . This, however, does not prevent services from receiving whatever external data including an identity of a 3rd party.

## Federated Identity vs. Identity Federation

Service and commercial boundaries constitute a new context and reality for an identity control. Well-known Federated Identity (FID), which utilises an identity “token” associated with an entity – person, or system, or application. This token is called federated because it is trusted in different identity realms. Such trust is usually based on a super-Authority situating above identity realms, which cannot exist in a service environment separated by ownership boundaries.

Working in Cloud, we cross the line from Technology to Business where values, intents and solutions have different reasoning and justification, where a value of revenue prevails over a value of technical dexterity. For instance, Tim Mather published in RSAConference.com, “*When it comes to federated identity management (FIM) in the (public) cloud, FIM is 'broken'.  It is not the technology that is broken… It is the business process (i.e., credential acceptance) for FIM that breaks*” [3]. Talking about a 3rd Party authentication based solution, he explained, “*The problem has not been technology, but business considerations, and specifically legal issues around accepting an authentication credential issued by another organization.  (OpenID effectively skirts this issue by not being trustworthy enough for business transactions*.)”

A Cloud PaaS illustrates the problem with FID: PaaS is supposed to offer specialized development tools to work on the platform. Unfortunately, no PaaS Provider can offer you the full spectrum of development and testing tools you use today in “internal” IT. Either this Provider or you will end-up with a dozen of different offering of development tools from different Clouds Providers. Are you going to open your security realm to all of them? Plus, if just one of those Providers appears to be a Public Cloud Provider, you risk opening your systems and applications to the entire world. Ask your Business what they think about such an opportunity.

This, however, does not disturb Microsoft’s Identity Architect Kim Cameron who announced a modern Microsoft’s vision on FID named Identity Management as a Service (May, 2012) that is based on Azure Active Directory [4]. This is the same Active Directory but on dopes now. It offers the same centralised identity control ignoring service independence and ownership boundaries. This solution brings no additional value to a Clouds consumer beside an additional high risk of getting all enterprise identities out of the enterprise’s control. About 6 years ago, Eric Norlan wrote, “*companies would find identity data too important to hand-over to others*” [5]. This statement still stands.

In contrast with FID, an Identity Federation (IDF) is a structure of distributed Authentication Authorities that serve only their own members but can federate an authentication request based on trust relationships between them. Members of these authentication realms rely on their own authentication mechanisms (resident Authentication Authority) that include two layers. In the first layer, every service in the identity realm engages its Authentication Authority to confirm an identity of the user or requester. The second layer may have dual implementation: if an identity of the user is unknown to the particular Authentication Authority, the request for identity verification may be propagated through a federation of Authentication Authorities until it would be recognised and verified or finally denied.

Alternatively, initially engaged Authentication Authorities are able to upload identity portfolios from each other during a bootstrap and reconcile periodically, i.e. they will be able to verify requested identities locally. Actually, the latter implementation is not a good idea because 1) it is insecure to collect all data of different Authentication Authorities in one place, and 2) it does not scale well because a number of federated realms and volumes of their content are constrained by the technical capabilities of each participating authority.

## Solution based on the Identity Federation Model

Among “SOA Design Patterns”, there is a pattern-candidate known as Federated Identity Pattern [6]. Unfortunately, this pattern suffers from the same illness of a vague name as Web Services technology does. While the name of this pattern points to FID, its content is closer to IDF. The pattern respects an independence of authentication realms and does not try to cross them with a unified token. Instead, it proposes having a Cloud Authentication Broker between different authentication realms, e.g. A and B, with their own Authentication Authorities. This Cloud Authentication Broker utilizes its trustful relationships with A and B and may convert an identity token from A in an identity token for B. Then, the Cloud Authentication Broker may provide converted B identity token to the requester in A that now can directly engage the services in B with B identity token.

At the same time, this pattern says nothing about what might happen if a requester in A shares the B identity token with his or her friend C, and this friend propagates the token to foe who might be interested in working around the trust and control required between B and its consumers. In this scenario, there is no need for fighting intruders; Federated Identity Pattern opens the door for them by its own hands.

An IDF solution is based on an agreement between resident Authentication Authorities but does not need a 3rd Party Cloud Authentication Broker. Assume you control the Authentication Authority in your identity realm. Each external partner’s realm may be represented by a realm representative (RR) in your realm. The RR is your internal entity and may act only according to your internal policies and rules. This RR is also a member of the external identity realm.

The RR represents business capabilities of services (covered by your identity realm) that the service owners want to offer to the external consumers. Simultaneously, this RR represents all requests from external consumers for the capabilities of your services in your realm.

The RR plays a role of an ambassador or delegate. Your identity realm may have as many RR as many different identity realms of consumers and services you work with. The main difference between the RR and centralized FID is in that authentication entity in FID exists on its own business while RR is under the residential identity control and, therefore, may be trusted by the business participants.

Finally, we would like to outline that a notion of RR is very well in line with the concept of SOA in Business. The RR fulfills one of the fundamental requirements of service-oriented Business –“*in order to conduct a business with someone, we have to know this one*”. That is, before a service may be engaged, a consumer has to set trustful relationships with the provider or owner of that service. In IDF, this is the relationships between a real consumer and related RR in the role of service delegate as well as between the RR and a service in another identity realm.

A RR may be viewed as an intermediary, which reminds us about ESB. Technology tends to place an ESB system between consumers and services to de-couple them to the level where they do not know about each other. It is regrettable because it violates [7] both business interaction requirement and the service interaction model defined in SOA-RAF. If an ESB wants to hide the service and manage anything in the service interaction beside message routing and endpoint resolutions, it has to become a service or service delegate itself, not a part of infrastructure, i.e. possess and provide all related business responsibilities to the consumers like an RR does.

## A Practical Example

Let us summarize what we just discussed – we know about Knight Rules of SOA Ownership and about implementation of Identity Federation based on representatives of one authentication realm in another. A realm representative is a logical abstraction that may be implemented in many different ways. Here is an example that we observe frequently in the real world.

Assume we have a consumer company Ace Corporation and two services, Catering and FoodPro. The Ace and Catering belong to an authentication realm Town while FoodPro belongs to an authentication realm Vill. Ace has established trust and agreed on certain contract with Catering to serve Ace’s canteen.Thus, Ace may invoke certain functions/operations of Catering. The contract between Ace and Catering states that Ace is interested in and ready to pay an agreed price for local food ingredients only and will not cover delivery from outside of Town.

Apparently, Catering has to engage an additional capability similar to the one provided by FoodPro because of the size of the Ace orders. Catering learns about FoodPro from a local Truck service. Neither Ace nor FoodPro knows about each other. That is, Catering cannot negotiate lower prices with FoodPro (to save on delivery from Vill) by referring to Ace. So, Catering contracts a Truck service, which runs in both Town and Vill. Truck served FoodPro before and can get some discounts on the delivery, i.e. it can use certain functionality of FoodPro available for the Vill residents only. Truck uses its presence in both Town and Vill to cross the boundary of “residency” between them.

When Ace sets a service contract with Catering, the latter sets supporting service contract with Truck, which continues the chain and signs related service contract with FoodPro in Vill. It is possible that Truck signs a service contract with Catering even before signing supporting service contract with FoodPro. When FoodPro receives a contractual request from Truck, FoodPro does not ask questions about how it was triggered because the request is made in Vill. So, if FoodPro accidentally receives an identity of Ace from Truck, the FoodPro simply would not know what to do with it.

## Thoughts to Take Away

In real-world SO Ecosystem and in Clouds, propagation of end-user identity is useless and even insecure (one business service can learn who are the consumers of another business). We have established the fact that different authentication realms, as well as different service and business ownerships are in commercial competition in Clouds because of antagonistic commercial interests. So, investments into the propagation of end-user identity beside the scope of directly interacting entities are wasted.

We have learned that services concern about their own immediate consumers and suppliers and do not deal with consumers of others. Identities of consumers of other services are unnecessary extras. Services or consumers may be in need for functionality provided by services in foreign identity realms and the only commercially sound solution for crossing the realm boundaries (as well as the ownership boundaries) is Identity Federation. In this federation, all services and consumers stay under the controls and protection of their residential Authentication Authorities while the latter federate with each other via shared authority representatives..

## Resources

1. Reference Architecture Foundation for Service Oriented Architecture Version 1.0 . Committee Specification Draft 03 / Public Review Draft 02, July 2011. On-line resource: <http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra.pdf>
2. Michael Poulin, “Knight Rules of Ownership in Service-Oriented Ecosystem” Business Ecology Initiative & Service-Oriented Solution, ebizQ, Jun 2012. On-line resource: <http://www.ebizq.net/blogs/service_oriented/2012/06/knight_rules_of_ownership_in_service-oriented_ecosystem.php>
3. Tim Mather, “Federated Identity Management in the Cloud” Experienced Security ,Dec 2010. On-line resource: <https://365.rsaconference.com/blogs/tim-mather/2010/12/20/federated-identity-management-in-the-cloud> - [Federated Identity Management in the Cloud](http://365.rsaconference.com/blogs/tim-mather/2010/12/20/federated-identity-management-in-the-cloud)
4. Kim Cameron, “Identity Management As A Service” Kim Cameron’s Identity Weblog, May 2012. On-line resource: <http://www.identityblog.com/?p=1205>
5. Eric Norlan, “Identity management as a service” Digital ID World, Apr 2006. On-line resource: <http://www.zdnet.com/blog/digitalid/identity-management-as-a-service/15>
6. Federated Identity Pattern . SOA Patterns, a Community Site for SOA Design Patterns. On-line resource: <http://www.soapatterns.org/federated_identity.php>
7. Michael Poulin, “Does ESB fit with Business Service?” Business Ecology Initiative & Service-Oriented Solution, ebizQ, Aug 2009. On-line resource: <http://www.ebizq.net/blogs/service_oriented/2009/08/does_esb_fit_with_business_service.php>