How to make a Federation manageable

Christian Geuer-Pollmann

European Microsoft Innovation Center,
Ritterstrasse 23, 52072 Aachen, Germany
chgeuer@microsoft.com
http://www.microsoft.com/emic/

Abstract: Nowadays, the setup of seamless, cross-organizational working environments is a challenging task, mainly because of security-related problems. The available options for cross-domain collaboration are often dysfunctional, expensive and not in line with the organization's business needs. Despite the proliferation of claims-based mechanisms like SAML, WS-Trust or WS-Federation, additional guidance is necessary how to effectively apply these technologies. The architectural pattern and ideas presented in this talk are an attempt to solve a common class of problems in collaboration space.

The opinions expressed in this keynote presentation are my personal ones, and do not represent or endorse my employer's point of view in any way. This presentation is highly influenced by the current thinking and work inside the EMIC security group. In particular I would like to thank Laurent Bussard, Joris Claessens, Stéphanie Deleamont and Mark Gilbert.

1 How it is today – Surviving in security management hell

A problem statement

In this keynote talk, I would like to tell a story about the security management nightmares information workers encounter when they have cross-partner collaborations with people in other organizations. During our daily work, we often come into situations where we have to share data or services with our partners. A typical example (from our own site) is proposal preparation: multiple partners want to collaboratively share documents and other files, so that these data objects are accessible to a small set of people, and these files are continually changed by many people. Working with multiple companies is a common situation for many people: sales forces during contract negotiation, companies that expose services to government bodies, or consultants working for a customer.

The problem is that the set of collaborating people is an ad-hoc formed group that consists of people from different organizations. These organizations understandably shield their internal networks against the outside world. This is normally done by firewalls which limit external communication channels to well-understood protocols like HTTP and e-mail. The underlying assumption of this compartmentalization process is that the majority of interactions happen inside the organization’s network, so that it’s fine to constrain outbound and inbound resource access. The administrators have to balance the organization’s security needs with the employee’s user experience and comfort level.

While this compartmentalization helps administrators to assess their network’s attack surface, it is a serious obstacle for the people who want to collaborate. In situations where only few cross-partner interactions happen, sending documents forth and back via e-mail is a minor annoyance for users. In other scenarios, e-mail exchanges just don’t work or do not scale well. For instance, in situations where more sophisticated interactions are necessary, such as access to specific services, shielding of the internal network is dysfunctional, because users are forced to choose non-favorable (non-manageable or insecure) options. We’ve seen many options people normally turn to and I will quickly outline the two that are the most successful (but still fairly dysfunctional).

Dysfunctional option #1: Do-it yourself Service Hosting – Wrestling with the IT department

Imagine a scenario where the collaborating people need a service for shared file storage. Bob offers that his company could host this service in the company’s DMZ. The DMZ is the ‘demilitarized zone’, a network perimeter outside of the company’s internal core network. The DMZ has less restrictive permissive security policies than the internal network. One advantage of this approach is that the service is operated inside a controlled area by Bob’s own company, so that the service security is controlled by one of the partners. The other advantage is that the IT department is aware of the fact that people from other organizations have access to resources inside the company.

Unfortunately, there are also three downsides with this approach: First of all, somebody (Bob) has to persuade the IT department that there is a business justification to deploy this specific service inside the company’s DMZ. This is usually a hassle that few people in big organizations would like to go through. The ones that do are rarely successful in a time scale under three months of lobbying, which is not very appropriate for short term engagements.

The second problem is user management: somebody (usually Bob) has to ask all the collaboration partners about which of their employees have to get access to the service. For each of these people, a guest account has to be created. During the life cycle of the project, this set of users will change, thus requiring Bob to make sure that the company’s IT department gets notified of the changes.

The last problem is that the IT department also has to make sure that the access control lists for the service are in line with the business needs, so that Bob has to specifically instruct the administrators about access rights.

As a conclusion, this option respects corporate (security) policy, but has a high price tag with respect to management overhead and setup time.

Dysfunctional option #2: Service Outsourcing – Bypassing your company's security controls

Another option for the user is to host the service at a 3rd party site, e.g. at an application service provider. One potential advantage is that the costs of setting up the service may be slightly lower, because the company's IT department does not have to provide the infrastructure. Nevertheless, the costs for user management are the same, as accounts for people from different organizations have to be set up and maintained. One big advantage of this approach is that the potentially vulnerable services do not have to be hosted inside the company's network, thus reducing the threats to the network itself. The direct drawback is, that potentially critical and sensitive data that the partners want to share, is hosted outside their own trusted networks. In addition, this solution has very bad audit characteristics: neither the company's IT department nor the executives of the company may be aware of the fact that the company's business relies on outsource services, and if something goes wrong, it can be very hard to figure out what happened.

The next section describes our fundamental beliefs how collaborations work. Section 3 on *"How it should be"* outlines our thoughts how collaboration (and the necessary authentication and authorizations) should work ideally. The mechanisms that will help solving these problems are federations and 'claims-based security', described in section 4. Section 5 *"How to apply these tools"* provides an architectural overview how we believe federations and the claims-based security model can be utilized to solve our problem. Section 6 *"What it brought us"* assesses and outlines the benefits for the different stake holders. The document concludes with a brief outlook.

2 Our beliefs regarding collaboration

Before going into the details of the scenario, I would like to outline the beliefs we have and the assumptions we make, in order to know how a 'potential' solution could be:

1. Trust across organizations depends on people who trust each other

When it comes to cross-organizational collaboration, the decision to work together is often done by people who know each other personally. The fact that two or more companies collaborate in general may not directly help people during their daily work. The actions that are performed in a particular collaboration should be traceable back to these people. This means that actions in the collaboration are justified because two humans trusted each other and intended to collaborate. When for instance two people exchange information, this exchange should be tied to an existing collaboration. The collaboration itself is tied to the people in the organizations who rooted the trust and bootstrapped the project.

2. Whoever makes a decision should have the tool to enforce that decision

Much frustration arises from the fact that people like administrators have to make or enforce decisions that are beyond their duties. Business people who start a new project should control who of their colleagues works on 'their' project. This means that the business people should be responsible for assigning people and resources to their project and for defining the roles of these people and resources. In return, administrators should be freed from implementing these specific user and role assignments for the business people, while having the confidence that the network remains protected.

3. Collaborations must be visible and manageable inside the company

Another source of frustration is missing information, which people would need to get their job done. Ongoing collaborations (and their specific details) should be visible to various people in the company. For instance, the CEO should have a tool at hand to easily find out whether somebody inside her company collaborates with a specific partner. Administrators should be able to determine whether other partner organizations have access to a specific resource inside their network. Administrators should also have the chance to associate inbound messages to certain collaborations. That would enable administrators to temporarily block message exchanges with a certain partner, if they learn that this partner has a network security problem.

3 How it should be – A fairy tale

Now that I've described our beliefs, let me walk you through a quick scenario that describes how collaborations could work, and how technology can be used to help, instead of having to wrestle with it.

Bootstrapping trust

Alice and Bob have known each other for a long time and have worked together on past projects. Based on their past experience, they trust each other personally and plan to work together on a collaborative project. Alice works for the company 'Contoso Ltd', whereas Bob works for 'Fabrikam, Inc'. We also assume that both have a mid-level managerial position inside their companies' hierarchy, so that they are permitted to start collaborations on their own, on behalf of their respective companies. Alice and Bob agree to start a specific collaboration.

The first thing Alice and Bob have to do is to give this new collaboration between Contoso and Fabrikam a name, like 'Project X'. The name is necessary to distinguish between parallel projects that exist in the same partner organizations. It is necessary that these different projects can be distinguished from each other.

Alice and Bob have to exchange their 'corporate business cards'. A corporate business card is similar to the root certificate of a corporate certification authority, enriched with additional information like the network address of the company's 'security token service' (STS). An STS (described in the section 4) is a service that can issue and validate security tokens for a given trust domain. By exchanging the corporate business cards, the two companies' IT systems can validate each others security tokens, thus creating a federation.

Enacting a 'constrained' federation

After agreeing to start the project, assigning it a name and exchanging business cards, Alice and Bob instruct their corporate IT systems that the project now has to be started. Basically, Alice tells Contoso's IT system:

> *"I ('Alice@Contoso') have started a project with Fabrikam. My peer contact there is Bob@Fabrikam. The project is called 'Project X'. Here is Fabrikam's business card, so now you can validate tokens they issue. The project should be enacted now, and it should expire in three months from now. If somebody has questions about that collaboration, just ask me, because I (Alice) am the project's business owner from our side."*

Bob does the same inside his own network. After this step, the *constrained* federation is enacted, i.e. both companies know about the project and can validate each other's security tokens. 'Constrained' means that the federation must only be used in the scope of the specific project, i.e. it must not be used for other purposes.

Unfortunately, nobody can (yet) do anything inside the project, because neither people, nor resources are associated with it. Both Alice and Bob have to decide what people from their own organization are assigned to the project. 'Being assigned' to the project means that these people are authorized to request specific (branded) security tokens that can only be used inside that particular project. This also applies to resources: Messages with project-bound security tokens must only be forwarded to services that are associated to the project.

4 Federations and the claims-based security model – Our knight in shining armor

What's in a Federation

Nowadays, federated identity management is a solution for the user management problems in the above scenarios. Multiple technology proposals, ranging from SAML, Shibboleth and the Liberty Alliance to WS-Trust and WS-Federation, attempt to provide solutions for federating trust domains. Regardless of what specific technology is used, the main question is: *"What does it practically help, now that we've set up an* *identity federation with our partners?"* The easy answer is that a federation helps entities inside one organization to authenticate subjects from another organization. Simply speaking, when a service receives a security token, it can be validated whether the token was issued by a partner organization. This is very similar to signed e-mail in which a recipient can be sure that the mail originated in a particular organization. So using federation technologies, one can implement the first step to a cross-organizational single sign-on.

The unanswered (and tough) question is how a recipient (like a service) can validate whether the subject is *authorized* to perform certain actions, like invoking the service. The fact that an incoming e-mail comes from one of the collaboration partners does not mean that the sender of the e-mail is part of the specific collaboration project and is authorized by the business owner inside the partner organization. Without asking "Does your colleague Greg work on our project", it is impossible to validate whether an incoming request is authorized or not.

Claims-based security

In a 'claims-based' system, security decisions are performed based on 'claims' that are supplied by a requestor. In this context, a claim is "an assertion of the truth of something, typically one which is disputed or in doubt" [3]. SAML and WS-Trust/WS-Federation are claims-based systems. A claim could be an X.509 certificate, which asserts that the subject that holds the corresponding private key is 'known' to the CA under the given distinguished name. Other classes of claims could be a username, a SAML assertion or a capability (from a capability-based security system). In our example, a claim is a statement by one of the partners that a certain user or service is associated with the project. Such a claim can only be validated by parties which are part of the constrained federation.

Claims are statements by a claim provider about a particular subject. To ensure that these claims really originate from the claim provider, claims can be protected using data origin authentication mechanisms like digital signatures or message authentication codes. Multiple claims can be combined, in order to build a higher-order claim. These higher-order claims are called 'security tokens', i.e. a security token contains one or more claims. By their very nature, security tokens can have an arbitrary amount of complexity. Comparing this with an X.509 certificate, the X.509 certificate is very simple because of its well-defined semantics (ignoring that X.509v3 extensions and OIDs make it harder).

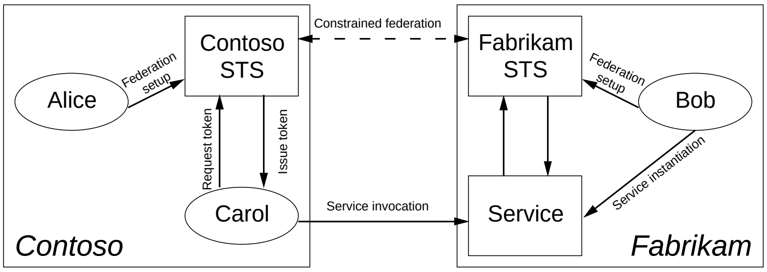
A client that aims to invoke a service may not be in possession of the appropriate security token that is necessary to invoke a particular service. Imagine a client that possesses a username/password pair or a Kerberos ticket that are only valid within the corporate network. With these security tokens alone, it is impossible to invoke a service in another trust boundary, because these security tokens are not understood or will not be accepted by the target service.

This dilemma is solved by token transformers, so-called 'security token services' (STS). An STS is a service that can take an existing security token (like username/password) and transform it into another token that will be valid inside another trust boundary. This transformation can be either issuing a new token or validating an existing token. To get a token issued, the client asks the STS: “Please, I have this token here that you can recognize, please give me a token that I can use over there.” During token validation, the recipient of a security token asks the STS: “I received this token here and I do not understand the claims inside the token. Could you please bring it into a form that I can understand?”

The concept of security token services, together with a very simple language to request and validate security tokens is specified in the WS-Trust specification [1]. The WS-Federation specification [2] is a guideline based on WS-Trust, which describes how to combine different STSs in order to implement identity, account, attribute and authorization federation, as well as delegation across different trust realms. For further reference, the ‘Laws of Identity’ [3] provide an excellent background how identity systems in general (and the claims used therein specifically) should be constructed in order to be acceptable for the different stakeholders.

5 How to apply the tools – A simplified architectural overview

The following illustration provides a simplified overview on the solution architecture: Both Alice and Bob setup the constrained federation inside their respective organizations.



The constrained federation is established by Alice and Bob inserting the entry about the project into their companies’ security token services. After this entry, each STS knows *that* the project exists and what the security token of the partner’s STS is. In addition to that, Alice instructs Contoso’s STS that Carol is part of the project. This means that the Contoso STS will issue project-bound security tokens to Carol if requested by her. On the peer side at Fabrikam, Bob dynamically creates a service and ‘adds’ that service to the project by inserting that association into the Fabrikam STS.

Besides just associating Carol to the project, Alice also needs a way to attribute Carol’s STS entry with roles statements or similar claims. So Alice needs to be able to express that Carol is an ‘editor’ or ‘reviewer’ inside the ‘Project X’ collaboration. With that additional claims embedded into the cross-organizational security token, the Fabrikam STS can decide not only whether the invoked service belongs to the ‘Project X’, but also whether the specific operation is permitted to users with the role. This decision can be taken by the Fabrikam STS because the security token contains role claims, and because Bob inserted the “‘Project X’-Editors can write on services associated to Project X, reviewers can only read” into the STS. This information can be only provided by Bob, because Bob is the business owner of the service and the collaboration, and should be able to answer (and maintain) such information.

One important aspect to note down is that ‘regular’ users (which are not system administrators) will be able to insert security-critical information into the company’s overall security system. This implies that ‘regular users’ will be able to ‘open the door’ to the company’s internal network to people which do not belong to the company. This implies a serious threat to the network: It must be ensured that only services which are associated to the project will accept incoming messages from external parties. In order to ensure this, the company’s network must have strong enforcement components that permit message delivery only to services that belong to the project. An additional mechanism to reduce the threat potential is to have dynamic service instantiation and strong process isolation, so that services are only associated to a single project. As a first step, conservative deployments could instantiate the exposed services inside the company’s DMZ or another compartmentalized area.

6 What it brought us – Problems solved?

To conservative security people, this scenario should be frightening: regular users (who usually have no security education) will be able to expose resources inside the company to external people. So what does this apparently risky idea bring? There are multiple stakeholders impacted:

IT and network administrators

- The first advantage for administrators is that they can concentrate on systems administration work, without being disturbed with user management and change requests to specific access control lists.
- Administrators have the complete overview of what resources are exposed to other companies inside the corporate STS. The corporate STS has a complete view on what constrained federations exist. For each of these federations, the central corporate STS provides information which services are associated with the federation. In addition, the STS provides the information who the other companies are that have access to these services. All this information is necessary to perform audits over the IT systems and to determine the potential attack surface and threats that the corporate network is exposed to.
- Each incoming message must have a valid security token attached. Messages without valid security tokens can be blocked easily. In case of suspicious messages, the attached security token enables administrators to find the business owner both inside their own company, as well as in the partner companies. Therefore, malicious messages that have been sent by people inside partner companies can, in corroboration with the partner company, be linked to an individual inside that partner company.
- Administrators can also decide that all interactions (both inbound and outbound) with a particular partner can be blocked as long as necessary. For instance,

collaborations can be ‘put on hold’ if certain partners have security problems with their IT infrastructures. The corporate STS is the single point of control to enforce such policies.

- As a last point, administrators have the confidence that proprietary data, that is shared across partners, is not stored outside the federation, e.g. on 3rd-party IT systems like an application service provider. This assures that this data is protected either by their own or by the partner’s network.

Top executives

- Top-level managers like e.g. a CEO can extract valuable business information from the central STS, e.g. “Who inside my own organization works together with this partner company?”
- If necessary, top-level managers can use the central STS in order to enforce business decisions, e.g. “We terminate (or suspend) all collaborative business with this specific partner.”
- Another interesting option is to prevent the setup of certain collaborations: “For the time being, no new collaborations with these specific partner organizations can be established without further vice president approval”.

Mid-level managers

The mid-level managers are people like Alice and Bob. These people are the core of the collaborations.

- With the approach presented in this paper, these people have an effective tool to establish new collaborations, which gives them full control over the business-related details of the collaboration.
- This tool provides a minimum-effort mechanism to directly associate employees and resources to collaborations.
- It ensures that their business relationships are automagically visible to the top-level management. This gives them confidence that each collaboration they establish will be in line with the company’s overall partner strategy.
- As a last point, the STS is *the* central tool to maintain their relationships with other partners. For instance, it is easy to determine “What collaborations and projects do me and my team own?”

7 Conclusions

This paper presented our view on how security management for distributed systems could be enhanced, in particular for situations where cross-organizational collaboration is a business necessity. The driving force during the development of this architecture was to focus on the business needs of the different stakeholders. We believe that the developed security architecture correctly reflects how human trust relationships in cross-partner collaborations work. The next steps will be to validate that such a system is manageable at a broad scale. This validation is expected to happen in the scope of collaborative research projects, like the European FP6 project TrustCoM.

One common aspect for aspect for software companies is that the employees have to use new software themselves before rolling the products out to clients. For ourselves, the main challenge will be to apply this new model to our own collaborative working environment.

References

1. Martin Gudgin, Anthony Nadalin: *WS-Trust*, (February 2005) http://msdn.microsoft.com/ws/2005/02/ws-trust/
2. Chris Kaler, Anthony Nadalin: *WS-Federation*, (July 2003) http://msdn.microsoft.com/ws/2003/07/ws-federation/
3. Kim Cameron*: The Laws of Identity*, (May 2005), http://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.doc