Provisioning scenarios in identity federations

Project: GigaPort3
Project year: 2010
Project manager: Remco Poortinga-van Wijnen
Author(s): Martijn Oostdijk, Bob Hulsebosch (Novay), Niels van Dijk, Roland van Rijswijk, Hans Zandbelt (SURFnet)
Due date: July 2010
Version: v1.0, final

Summary
Supplying information for applications and services, related to a user account within a federation is called provisioning. Deprovisioning is the opposite process. In identity federations (de)provisioning is not trivial as it involves cross-domain identity communication. Moreover, though mostly the identity information can be provided during authentication, there are applications that need to be provisioned before the user logs in. Examples of such applications are dynamic group management services that are common to e-Science. The fact that standardised support for provisioning is far from ideal for federated environments makes it even harder to implement.

This report gives a state-of-the-art analysis of provisioning products and standards and of the, still ongoing, federated provisioning debate. It classifies different types of applications and different types of provisioning scenarios in order to come up with a framework, which is helpful when selecting a strategy for dealing with federated provisioning. The results are validated by exploring (at a suitable level of abstraction) a case study on dynamic group management.

This publication is licensed under Creative Commons "Attribution 3.0 Unported". More information on this license can be found at http://creativecommons.org/licenses/by/3.0/
Colophon

Program: GigaPort3
Workpackage: EDS
Activity: Federations
Deliverable: EDS-4
Access rights: Public
External party: Novay, www.novay.nl

This project was made possible by the support of SURF, the collaborative organisation for higher education institutes and research institutes aimed at breakthrough innovations in ICT. More information on SURF is available on the website www.surf.nl.
1 Introduction

Provisioning (of services with user account information) and federated identity management form two distinct areas of identity management. The communities studying these different areas come from different backgrounds. Whereas provisioning has its root in intra-enterprise identity management, mostly dealing with making sure that distributed applications within a corporate network have a common and synchronised view on identity information, federations have their roots in communities wanting to collaborate by transcending organisational boundaries and have focused mainly on authentication and related trust issues.

Deprovisioning is the process to eliminate an existing user account, which includes user account deregistration and potentially cleaning up. As such, deprovisioning can be considered as a special kind of provisioning.

Intra-organisational provisioning and deprovisioning is already difficult. Different applications store identity information in different ways and many legacy applications lack support for identity externalisation in central directories. In identity federations where multiple partnering organisations trust each other and share identity and attribute information across domains (de)provisioning becomes even harder to achieve. Though federation is enabled through the use of open standards such as SAML there are little to no concrete solutions for identity provisioning in federations. SAML can be used for provisioning of identity information but lacks specific features that are required to support e-Science applications in which groups of researchers work together. Moreover, available standards for provisioning such as SPML find little adoption and seem to lack cross-domain functionality as they are focussed on enterprise provisioning.

This report gives a state-of-the-art analysis of provisioning products and standards. It also looks at the ongoing federated provisioning debate. It classifies different types of applications and different types of provisioning scenarios in order to come up with a framework that helps selecting a suitable strategy for dealing with federated provisioning. The results are validated by exploring (at a suitable level of abstraction) a case study on dynamic group management.

1.1 Reading guide

This deliverable scans several provisioning implementation scenarios in federated identity management environments with a particular focus on e-Science applications. Section 2 introduces the provisioning and deprovisioning problem and the current debate. Section 3 provides a state of the art overview of provisioning standards. Section 4 describes provisioning in the context of a federation and classifies scenarios in terms of trigger moments and archetype applications. Section 5 discusses specific provisioning issues in e-Science applications by examining a case study. Finally, Section 6 concludes with the major findings of this work on provisioning.
2 The provisioning problem

2.1 Provisioning

Applications need data, and for applications that distinguish between individual users this includes data that is related to users: identity information in the form of attributes. Keeping identity attributes up-to-date across different applications can be a challenge at times: attributes keep changing as users start to work on different projects, move to different departments (or even different organisations), or are assigned new roles and responsibilities. Provisioning (and deprovisioning) is the sub-domain of Identity Management which tackles this challenge.

Provisioning is the creation or updating of identities. The purpose of provisioning is to allow newly created identities to start using services or resources provided by the application and to ensure identity information is consistent across a range of distributed applications. Depending on the viewpoint provisioning is sometimes also referred to as user-, account-, or service provisioning. Provisioning mostly concerns creating user accounts initially when the user first needs access to various applications. Apart from that, it also deals with reconciliating existing user accounts from different sources and keeping user accounts up-to-date.

A related problem, deprovisioning, is often studied in parallel. Deprovisioning involves removing user accounts from applications and data that is related to such accounts when they are no longer needed. Deprovisioning should happen when a user no longer has a right to use certain services or resources provided by the application.

Consider, for example, an enterprise. When a new employee starts to work, the HRM department adds information about the employee in the enterprise’s HRM systems. This means, amongst other things, that an entry is created in the database, a message is sent to the payroll administration application (which may be hosted by a third party), an email account is created for the new employee, and perhaps a personal Web page on the company website. But also non-digital resources such as a company cell phone, a laptop, or a company car could be provisioned for the new employee: an application may place a purchase order or send an email message to the IT department or a third party prompting either an automated system or a human operator to take action.

When an employee’s contract has been terminated, it is equally (and perhaps even more) important to adequately deprovision the employee and his/her data that is no longer needed. The HRM department – again – is responsible for initiating a chain of actions which leads to removal of identity information in certain applications. From a security perspective, deprovisioning is more important than provisioning: a new employee who cannot access the company’s computer systems cannot start work and is therefore cost-ineffective; yet a (disgruntled) former employee who is still authorized to access company assets is a security threat.
Automated support for provisioning and deprovisioning improves overall efficiency of the processes within an organisation but is especially helpful in light of three issues: Information Security, compliance to rules and regulations, and resource management. First, the need to deal with information security incidents results in organisations wanting tighter control over who gets access to what resources. Typical identity attributes of users include the setting of roles and rights for various applications. Controlling such attributes from a central location makes it much harder to accidentally give users privileges beyond what they should have based on their role in the organisation. Properly deployed provisioning systems help to keep identity attributes (and thus authorisation information) in a central place while still allowing distributed applications to access that information.

Second, (somewhat related to the information security point) the demand for explicit and provable compliance to rules and regulations, in areas such as privacy and intellectual property rights (IPR) management drives organisations to implement tools to prove compliance. A provisioning system, in the context of an organisation, which is held to such rules and regulations, helps the organisation stay compliant and able to maintain an audit trail to prove compliance. When applied to IPR, if user accounts are not properly provisioned and deprovisioned it becomes much harder to determine who claims to own certain content, especially when external parties start claiming otherwise.

Third, for services where users can own certain objects that correspond to scarce resources, automated deprovisioning can improve efficient use of resources. Some services allow users to store large amounts of data, think of services such as Dropbox.com, YouTube or SURFmedia. Such data is associated with a user account. If the user no longer has the right to use the service or never returns to that service, then his or her account should be deprovisioned including its data. Unnecessary allocated resources become available for new users. Such deprovisioning may be triggered by an expiration time of the account or upon de-authentication, i.e. removal of the account when the user has left the company.

Provisioning has been recognized as an essential part of the identity management stack. Forrester, in their 2009 version of the “Forrester Wave: Identity and Access Management” report [4], indicates that companies looking to deploy identity management solutions choose to tackle provisioning first. The contribution of the introduction of adequate provisioning and deprovisioning processes to the efficiency, compliance, and security of the companies is apparently much greater than other identity management initiatives. Forrester calls Oracle, CA, and IBM the leaders of the pack, with Novell, Courion and Sun Microsystems following at close distance. Microsoft, SAP, and Hitachi ID Systems lag behind (at least with respect to Forrester’s rating model).

Gartner, in its 2009 version of the “Magic Quadrant on User Provisioning” report [5], indicates that provisioning is at the core of identity management. Provisioning products drive the other activities typically related to identity administration. Gartner puts the provisioning products of Oracle, IBM, Sun Microsystems, Novell, CA, and Courion in the leaders quadrant. The report does indicate, however, that it is expected that market forces will drive vendors closer together.
Implementing provisioning and deprovisioning is far from trivial. Applications will not readily support the emerging standardised provisioning technologies (outlined in Chapter 3), and it is not at all clear what a future-proof provisioning deployment strategy should look like. The trend towards cooperation of organisations within multi-organisation federations (as is the case, for example, in research and educational organisations) only adds to the complexity.

2.2 The federated provisioning debate

Federated identity management has become the standard way to deal with authentication (and somewhat related processes such as attribute sharing and authorisation) across organisational boundaries.

The topic of this report is how provisioning is to be implemented within the context of a cross-organisational identity federation: Federated Provisioning, so to speak. An example of such an identity federation is the SURFfederatie for higher education and research in the Netherlands, which connects the institutes of higher education as identity providers (IdP) with a number of service providers (SP).

Provisioning within the context of an identity federation is a relatively new opportunity. An identity federation makes provisioning both easier as well as harder. Easier, since authentication and the management of some attributes has already been externalised to a dedicated identity provider so the next step, federated provisioning, becomes possible. Harder, since provisioning and deprovisioning suddenly are done across the borders of an organisation which means that semantics, trust and security issues have to be dealt with.

Trusted and secure communication channels between the IdPs and SPs have to be established to arrange for account synchronisation. Somehow the SP must be informed that a user has joined or, more importantly, left a federation partner IdP, e.g. the university. Moreover, IdPs and SPs in the federation may have different provisioning and deprovisioning capabilities. Some standardisation may be required to ensure interoperability.

Judging from business whitepapers and scientific articles that have appeared, federated provisioning appears not to be a very hot topic (exceptions are a PingIdentity [18] paper and a SURFnet report [2] (based on [13]) discussed in Chapter 4). But judging from blogs in early 2009 it is clear that the topic does provoke emotional reaction in experts in the fields of identity and enterprise provisioning:
• Ian Glazer\(^1\) of the Burton group writes on January 7, 2009 in a blog entry: “Down with federated provisioning”. Essentially, Glazer says that federated provisioning does not exist. Provisioning in-house applications and SaaS applications should be equally difficult as neither have standards-based interfaces for provisioning (i.e. SPML). In providing SPML interfaces to their applications, SaaS vendors would do everyone a service.

• Nishant Kausik\(^2\) replies to Glazer's blog on February 3, 2009 by saying that federated provisioning does exist and is especially useful when used in tandem with security assertions arriving at the SP. Kausik thereby establishes just-in-time (JIT) federated provisioning (i.e. during authentication).

• To the surprise of Ian Glazer\(^3\), as he posts on February 5, 2009, the just-in-time scenario described by Kausik does exist in practice. Glazer claims that it is confusing for the end-user and is problematic with respect to deprovisioning. Glazer says that JIT provisioning is not provisioning.

• Pamela Dingle\(^4\) (of PingIdentity) joins the discussion on February 5, 2009 explaining that deprovisioning can be activated on de-authentication in a federated setting. She thinks that it is ironic to think that batch processing could make a comeback.

• Dave Kearns\(^5\) responds to Dingle’s blog post on February 5, 2009 by writing “The always intriguing Pam Dingle has come up with what I believe is an entirely new feature for IdM systems - self-service deprovisioning!”

• James McGovern\(^6\) adds a new dimension to the discussion on February 7, 2009 by explaining that provisioning entails more than just identity information. He sees opportunities to “kick off” business processes based on federated identity products.

• Pat Patterson\(^7\) (at the time at Sun Microsystems) on February 14, 2009 thinks that a feature of the (now defunct) Liberty Alliance standard, the Liberty Alliance Employee profile, may be useful for JIT provisioning. A SAML assertion created on authentication of the user can contain a reference to an endpoint to the Employee Profile service of the IdP, which can be used by the SP to pull the provisioning details from the IdP.

\(^2\) [http://blog.talkingidentity.com/2009/02/the_thing_about_federated_prov.html](http://blog.talkingidentity.com/2009/02/the_thing_about_federated_prov.html)
\(^3\) [http://www.tuesdaynight.org/2009/02/05/will-the-real-federated-provisioning-please-stand-up.html](http://www.tuesdaynight.org/2009/02/05/will-the-real-federated-provisioning-please-stand-up.html)
\(^4\) [http://eternallyoptimistic.com/2009/02/05/federated-de-provisioning/](http://eternallyoptimistic.com/2009/02/05/federated-de-provisioning/)
\(^5\) [http://vquill.com/2009/02/self-service-de-provisioning.html](http://vquill.com/2009/02/self-service-de-provisioning.html)
\(^7\) [http://blogs.sun.com/superpat/entry/federated_provisioning_liberty_to_the](http://blogs.sun.com/superpat/entry/federated_provisioning_liberty_to_the)
• Jeff Bohren\textsuperscript{8} on February 7, 2009 points out that JIT provisioning and deprovisioning just will not be sufficient for many applications because in some cases the identity information needs to be synchronized before the user performs his or her first authentication. This can be true of any application where the users interact with each other.

• Pamela Dingle\textsuperscript{9} responds to Bohren on February 8, 2009 wondering in how many cases JIT provisioning and deprovisioning might be sufficient.

The discussion seems to focus mostly on the advantages and disadvantages of the notion of just-in-time (JIT) provisioning whereby user account information is provisioned during user authentication. SAML seems a suitable candidate for such federated or cross-domain provisioning. Yet, in most federations SAML is primarily used for communicating authentication assertions that typically do not contain sufficient attributes for the systems on the SP side to create the necessary account. Moreover, provisioning usually requires a lot of identity data (like attributes, roles, entitlements, group memberships, etc.) to create accounts in applications. It is questionable if such information should be compressed in a SAML authentication assertion.

From more recent blog posts, it can be observed that the federated provisioning discussion seems to change its focus to cloud provisioning\textsuperscript{10}. The cloud-computing model shows strong resemblances with the identity federation model as it requires externalisation of identity and has to able to deal with a large set of remote and heterogeneous application services. Like in federations, provisioning in the cloud must be able to flexibly deal with application heterogeneity and be scalable.

\section*{2.3 Summary}

Provisioning deals with the synchronization of identity information across different applications. Provisioning is an important process in many organisations, for a variety of reasons, amongst which are information security, compliance to rules and regulations (privacy, IPR), and efficient handling of resources. Analysts see provisioning as a driving force for related identity management processes.

Combining provisioning with federated identity management, which addresses the trend towards inter-organisational cooperation, is a possible next step. Federated provisioning has recently been the topic of debates on the public blogs of experts in the field. While it is dangerous to draw conclusions from discussions in the ‘blogosphere’, at the very least one can say that federated provisioning is of interest to federated ‘Identerati’ and provisioning experts alike. Federated provisioning looks promising but the disciplines of identity federation and (traditional intra-enterprise) provisioning enter the discussion from different backgrounds.

\textsuperscript{8} http://idlogger.wordpress.com/2009/02/07/janus-versus-vulcan-in-federated-provisioning/
\textsuperscript{9} http://eternallyoptimistic.com/2009/02/08/silo-sync-vs-service-sync/
\textsuperscript{10} http://blog.talkingidentity.com/2010/06/my-gluecon-talk-on-federated-provisioning-and-the-cloud.html
The federated provisioning debate focuses primarily on JIT provisioning. This is an important yet somewhat technical issue concerning the mechanism for initiating provisioning and deprovisioning requests. It remains to be seen if the JIT-provisioning model is suitable to support all federation use cases. Furthermore, more fundamental problems, such as trust (see [1]) and security issues when scaling provisioning to identity federations, have yet to be addressed.
3 State-of-the-art of provisioning standards

Provisioning can be seen as a relatively simple synchronisation issue between applications: Information about user accounts needs to be distributed to reflect operational changes within an organisation. At the same time provisioning also has aspects of identity and, to be more specific, authorisation, since it implicitly deals with attributes describing the roles and rights of users. Deprovisioning in particular tries to solve a major security problem and security incidents related to lack of proper deprovisioning has been number one on various security incident top ten lists.\(^{11}\)

Any standards and transport mechanism could be used to ad-hoc provision a target system (for instance, one could send comma-separated ASCII files sent over FTP). Obviously, not all of these constitute advanced provisioning standards. This chapter describes those standards which are explicitly marked or used as provisioning standards in the literature. Chapter 4 expands the list of standards with federation standards which could be used to implement federated provisioning as well.

Technically, identity information can either be pushed from the identity provider towards a target to be provisioned or it can be pulled from the identity provider by the target when it is needed. Traditional provisioning systems take the first approach. Often solutions were implemented by creating ad-hoc connectors for very specific applications, or by involving manual procedures.

Push-style provisioning is standardized in OASIS’ Service Provision Markup Language (SPML, version 2.0 dates back to 2006 and is described in [16]). SPML defines the interface for both requester and target. Not many competing standards exist, apart from the Liberty Alliance project producing its own provisioning specification [12] which targets a very specific situation (provisioning trusted modules, rather than user accounts) and some cloud applications providing their own proprietary APIs. So, SPML appears to be the only viable candidate for a truly open provisioning standard. SPML is described in detail in Section 3.1. Chapter 4 describes provisioning in the context of identity federations and the role that some other standards (notably SAML and OAuth) could play in provisioning.

Pull-style provisioning has long been the domain of standards such as the Lightweight Directory Access Protocol (LDAP, version 3 dates back to 2006 and is described in [8] but is based on the X.500 standard dating back to 1993). In regular terminology this is never referred to as provisioning. Rather, applications externalise identity attributes to an (organisation wide) central directory. When attributes are updated centrally, provisioning and deprovisioning can be facilitated for more efficient. Central directories and abstraction mechanisms such as virtual directories are discussed in Section 3.2.

---

\(^{11}\) See, for example, the annual global security survey by Deloitte, available from http://www.deloitte.com/view/en_GX/global/services/enterprise-risk-services/security-privacy-services/article/decaf4021a001210VqnVCM100000ba42f00aRCRD.htm which lists user’s excessive access rights as the top security audit finding.
3.1 SPML

The Service Provisioning Markup Language (SPML, [16]) deals with service provisioning. SPML has been standardised by OASIS. Version 2 of the specification dates back to 2006 (version 1 was published in 2003).

Provisioning and deprovisioning terminology in SPML involves three stakeholders. First, the requesting authority (RA, i.e. IdP) decides that an existing application should be (de)provisioned for a certain user account. Second, the provisioning service (PSP) is asked by the RA to provision the user. Third, the provisioning service target (PST) is the application which accepts requests from the PSP and provisions the so-called provisioning service objects (PSOs). In essence a PSO contains the user's attributes.

Figure 1 shows the various roles and the flow of provisioning requests from RA to individual PSOs.

![Figure 1: SPML roles, terminology, and request / response flow. The diagram is based on introductions to SPML by Sun12 and IBM13.](http://www.sun.com/bigadmin/content/submitted/images/using_spml.fig1.jpg)

In the SPML 2.0 specification, the following operations are defined as so-called core capabilities:

- ListTargets, which lists the available targets of a provisioning service point
- Add, which adds a provisioning object to a target
- Modify, which modifies attributes of an existing object within a target
- Delete, which deletes an existing object within a target
- Lookup, which searches for an object given an object identifier

---

Every SPML 2.0 provisioning service point will implement these core operations. Additional capabilities may have been implemented. Requesting authorities can examine the response to a ListTargets request to find out what other capabilities are supported. Optional capabilities have mnemonic names such as `async`, `batch`, `bulk`, `password`, `suspend`, and `search`.

The latter non-core capability, search, is much more powerful than the core lookup operation and features complex filter expressions to select a set of provisioning objects. It features an asynchronous iterative process to get the search results from the PSP back to the RA.

SPML 2.0 also incorporates mechanisms to extend the set of core operations. Moreover, it supports different profiles (also referred to as "data models", similar to schemas in LDAP). The XSD and DSML profiles come with version 2.0 of the standard. Initially also a SAML profile for SPML was being worked on\(^\text{14}\), indicating some interest in integrating federated identity management and provisioning, yet that project seems to have been abandoned since 2006.

### 3.1.1 Working with SPML in practice

To get some hands-on experience, the authors of this report installed Sun Identity Manager 8.1\(^\text{15}\), and wrote some custom Java code against the OpenSPML API\(^\text{16}\) to add, modify, delete, modify and search for user accounts, see Figure 2.

![Figure 2: Sun's Identity Manager in the process of being provisioned from an external OpenSPML based Java application.](https://openspml.dev.java.net/)

---


\(^\text{16}\) See [https://openspml.dev.java.net/](https://openspml.dev.java.net/).
The purpose of a product like Sun Identity Manager is to manage a number of so-called resources (databases, LDAP directories, or custom proprietary applications for which a connector can be created) and reconcile existing user accounts on those resources. Essentially, Sun Identity Manager acts as provisioning service point.

Some observations made during this practical exercise:

- Identity Manager exposes only a single target through SPML (even though multiple targets (i.e. the resources) are supported at the back-end).
- This target only supports the DSML profile.
- Yet it does support many non-core capabilities: async, batch, bulk, password, suspend, search.
- In order to use the core operations RA needs a reference to a PSO. The non-core search capability is essential in “discovering” such references.
- Security is based on username/password (of a special Configurator account), which may be sufficient for an intra-enterprise scenario, but not in a scenario where Identity Manager is exposed to the outside world.

The last point is indicative for SPML 2.0. The mechanism that the various parties (RA, PSP, PST) involved in a provisioning scenario should employ to validate the identity of each other, i.e. the trust infrastructure, is out of scope of the standard. The standard specifies what requests and responses should look like. Since SPML is an XML based Web Services standard it can use WS-Security for message level security. But the standard does not specify how to set up a secure context between RA and PSP.

3.1.2 The future of SPML

SPML 2.0 has been around since 2006, which makes it as old as the SAML 2.0 specification (SAML will be discussed in Chapter 4). Where SAML has gained acceptance in the federated identity management community (SAML assertions form the de-facto standard language for exchanging identity information in many real-world federations), SPML has remained somewhat marginal. Of the major vendors of identity software only Entrust, IBM, Siemens, Sun and Symlabs offer support for SPML.

SaaS services such as Salesforce.com and Google Apps do not support SPML at this moment. Both of the named SaaS providers opt for a much simpler SOAP or REST API for provisioning. The fact that these vendors choose to develop their own APIs could indicate that the SPML standard in its current form is too complex or not flexible enough to be applied in such contexts (though other motives to opt for proprietary APIs cannot be excluded).

---

According to Burton Group’s Mark Diodati\textsuperscript{18} SPML is “on life support”. Diodati sees SPML in its current form and with its current support in products as simply not ready for building real world provisioning services. Many implementations lack, according to Diodati, the search capability, which is too complex to use and support. Adaptation of SPML would, again according to Diodati, be greatly helped if a reference implementation would exist and if a standard “inetorgperson”-like user schema were available. In the long run a simpler standard and harmonisation with federation standards such as SAML is needed.

Nevertheless, some of the typical SPML terminology (requesting authority, provisioning target) is used throughout this document.

### 3.2 Central Directories

By keeping all user information inside an organisation-wide central directory, there is no need to keep different applications synchronized, all attributes are always up-to-date. The Light-Weight Directory Access Protocol (LDAP) is the de-facto standard to query and modify such a central directory.

LDAP’s successor is the Directory Service Markup Language (DSML) which was published as an OASIS standard in 2002. DSML is mostly an XML variant of LDAP, with no new features. There seems to be little advantage of using DSML over LDAP’s native format (LDIF) except for DSML being XML based. It is not clear if DSML will replace LDIF in the near future. Note that SPML – described in Section 3.1 – also has a DSML profile.

A central directory for user attributes allows applications within an organisation to externalize identity information. Ideally, all of the identity attributes are stored in the central directory and in the central directory only. Applications that need up-to-date values of attributes simply pull them from the directory, which makes it trivial to keep attributes in sync across these different applications. In such an ideal situation there is no need for provisioning and deprovisioning, at least not for those application that do not keep user data within the applications themselves.

Unfortunately, in many cases it is not possible to externalise all of the identity information that applications process to a central directory service. Application specific non-identity data, ‘owned’ by the user, should, for example, be deleted (or marked as such) if a user no longer has the right to use that application. While this can be achieved using central directories (at a price that the directory then contains application specific information), it seems much more natural to implement such a scenario as push-style deprovisioning (using, e.g., SPML as described in Section 3.1) than as a pull-style directory approach. Also, some legacy applications (not willing or able to externalise their identity information) will always exist. These will still require out-of-band provisioning.

\textsuperscript{18} See \url{http://identityblog.burtongroup.com/bgidps/2010/02/spml-is-on-life-support-.html}. 


Another drawback to the central directory approach is that it does not scale well beyond the borders of a single enterprise for a number of reasons. Firstly, the directory server would become a potential bottleneck from a performance and availability point of view. Secondly, intra-enterprise directories tend to use their own structure to represent the hierarchical structure of the organisation, and do not necessarily use the same schemas (standardised sets of named attribute types) in use at other organisations. Thirdly, securing access to a directory is a non-trivial task. Most deployed LDAP servers use simple username-password authentication, often not even over SSL. Exposing such a server to the Internet poses a security risk. Even if it were possible to completely secure the directory servers of the different partners within a federation, precisely regulating access to each of the directories so as to only implement provisioning and deprovisioning would be complex.

Nevertheless, central directories for internal use within organisations are popular. Many organisations today use Microsoft’s Active Directory service, which is a specific implementation of an LDAP directory.

A technique called virtual directories (VDs) may partially solve the scalability problem in federated environments. A VD server consolidates multiple back-end directories so that from a client’s point of view there is only one virtual directory. The VD server acts as a proxy/gateway between clients and the back-end directory servers. It can translate names and route requests. A partner organisation can thus get a tailored view of the combined directories of the other partner organisations. A necessary pre-condition for the deployment of a VD within a federation is that there exists a trusted third party that can access all of the directories of all of the partners. Such a ‘federation operator’ may only exists in certain non-commercial homogenous communities (such as in higher education or e-Science federations).

KuppingerCole’s Felix Gaehtgens, in a blog post in July 2009, identifies Optimal IdM, Symlabs, Radiant Logic, SAP, Oracle, and RedHat as the main commercial VD manufacturers. Some open source VD alternatives (MyVD\(^\text{19}\), Penrose\(^\text{20}\)) exist, though it is not known how these perform compared to the commercial offering.

VDs, however, do not solve the provisioning problem for legacy applications that do not externalize their identity information to a directory. Such applications need adaptations before they can be provisioned and deprovisioned. In general it will be easier to adapt them to push-style provisioning and deprovisioning. Virtual directories do form a valuable abstraction mechanism which may potentially help lift identity information sharing from an intra-organisational context to a larger trust infrastructure. Yet, VDs do not solve the trust problem by themselves.

\(^{19}\) See \url{http://myvd.sourceforge.net}.

\(^{20}\) See \url{http://v1.safehaus.org:8080/display/PENROSE/Home}. 
3.3 Summary

Automated user provisioning and deprovisioning requires coupling of systems of different vendors, and thus standards. There main standard is this area in SPML, but the support in identity management systems and especially in application is very minimal.

The use of central directories is another approach to facilitate provisioning and deprovisioning. Ideally, externalisation of all attributes, for instance in central directories, gets rid of the need for provisioning and to some extent also deprovisioning. This is not a very realistic option, though, for several reasons. First, the amount of type of data for some application (like SURFmedia) is such that externalisation itself is not an option. Second, although central directories are an alternative to provisioning, but even with virtualisation these do not scale well outside of the native trust infrastructure of an organisation, moreover some applications will simply not support externalisation of identity attributes. The VD and traditional provisioning approaches are largely orthogonal.

Both approaches focus primarily on intra-organisation provisioning and seem not suitable for cross-organisational provisioning because trust aspects largely are out of scope for them. Some analysts doubt that SPML in its current state is mature enough for wide scale adoption, even in intra-organisational situations where trust is not (such) an issue. SPML is complex, not very flexible, and its future is unclear. Particularly in the SaaS community there is little support for SPML; Salesforce and Google apps offer their own proprietary provisioning interface. Yet, it is the only standard available for provisioning.

The next section discusses possible enhancements of these approaches to make them more suitable for a federative context and looks into the use of existing federated identity standards such as SAML for provisioning purposes.
4 Provisioning in the context of a federation

4.1 Identity federations

Identity federations are formed by connecting a number of identity providers (IdPs) and a number of service providers (SPs) bilaterally or via a hub-n-spoke approach. Users within the federation have credentials that can be validated by one of the IdPs (thereby authenticating the user). SPs will require a user to first authenticate with one of the IdPs before they will deliver the service to that user. The SPs effectively outsource the authentication process to the IdPs. This means that there must be a trust relation between SPs and IdPs within a federation.

It has been generally recognized that the only way to deal with large scale cross-organisational federations is to adopt open standards. Examples of such standards are SAML [15], Information Card, XACML [14], WS-Fed/WS-Trust/..., OpenID, and OAuth [17].

In the remainder of this report, the following assumptions about identity federations are made:

- There are four roles: IdP, SP, federation operator (i.e. hub), and user.
- Authentication is initiated by the user when he/she visits an SP.
- Authentication is directly followed by attribute release.
- Attribute release is from IdP to SP via the hub or a back channel, sometimes with explicit consent of the user.

4.1.1 Authentication

Federation is about asserting the identity of users. In a typical Web-based SSO federation scenario a user will request a service from an SP, be forwarded to his/her IdP, authenticate to the IdP, and be redirected back to the SP with the authentication assertions. The SP will grant access based on these assertions.

4.1.2 Attribute exchange

As a confirmation of a successful authentication, the IdP can release, with or without consent from the user, a specific set of attributes to the requesting SP. Although authentication is often the primary purpose of the federation, attribute exchange is valuable in its own right and can be viewed as a mechanism to provision an SP with user account information.
There are different options for the implementation of attribute exchange. If the SAML Web SSO profile is used, the attributes are typically sent in the response of the IdP, i.e. as part of the authentication flow of messages. Alternatively or additionally, attributes can be exchanged by having the SP perform a SAML assertion query/request. The SP needs to have an endpoint at the IdP to which to direct such a request. Moreover, such a separate SAML assertion query/request can only take place in the context of an existing user-SP session, i.e. the user has been authenticated.

Another option to exchange attributes is to use OAuth [9]. With OAuth a user can, temporarily, allow one service provider (the “OAuth consumer”) to access functionality of another service provider (the “OAuth provider”) in the user’s name. The functionality offered by the provider could include some kind of attribute exchange interface.

The meaning of exchanged attributes must be understood by both the IdP and SP. Often standard schemas are used, such as “inetorgperson” or “eduperson”, representing federation-wide accepted sets of attribute names and types. Attributes, may include information which is interpreted by the service SP as rights, roles, or memberships of groups, i.e. information which is used in making authorisation decisions. A more elaborate scheme for authorisation uses policies to capture some of the meaning of attributes exchanged and is specified in the XACML standard [14].

4.1.3 Potential standards for federated provisioning

The question thus arises: what standard is best suited for provisioning in the context of identity federations. From a technological point of view several standards could fit the bill. The SPML and directory-based provisioning standards described in Chapter 3: LDAP and SPML are certainly candidates. On the other hand federation standards, such as SAML, support attribute sharing and can be used for provisioning as well. Also, emerging technologies such as OAuth, being pushed by social networking site communities, should also be considered.

It may be convenient to use the central LDAP storage facility as a data access interface to attributes, as it is already well-supported in many tools. However, cross-domain attribute exchange via LDAP is not specified in terms of trust and security.

If SPML is to be used within a federation context, then the SPML roles (RA, PSP, PST.) should be mapped onto the federation roles user, IdP, and SP. As the IdP is usually responsible for the user’s attributes (the IdP is an attribute provider) it makes sense to map RA to IdP. The PSP component may reside at the IdP site as well or at the hub (i.e. federation operator site); the PST will reside at the SP site.

The ways in which some federation standards deal with the trust infrastructure forms an important differentiator: for LDAP and SPML trust issues are out of scope. SAML and WS-Trust based standards such as Information Card deal with trust inside the specification of the standard.
SAML is widely used in modern federations to communicate user authentication and attribute assertions. Often SAML authentication requests made by the SP are returned with responses that, besides an authentication assertion, also include several attribute assertions that can be used for subsequent provisioning and authorisation purposes. The SAML specification also defines a mechanism to separately query for attributes at the IdP via a back-channel. Both solutions make SAML extremely suitable for JIT-provisioning.

Other possibly relevant standards are, for example OpenID, (OpenID Foundation) and other APIs of the so-called “Open Stack” (OAuth, OpenSocial). Although these federated authentication frameworks are technically comparable to SAML, they are different with regards to the trust infrastructure that is natively supported. Another difference is the degree to which they deal with aspects such as user-centricity.

The OpenID Attribute Exchange functionality allows the federation SP to request additional attributes from the OpenID Provider (IdP) during the authentication flow. Essentially, when the SP detects that the user doesn’t have an account, it could validate the claims it received as part of the token, and if it needs additional data, then it could add a request for those to its authentication request. This can solve the data retrieval challenge, and more or less positions OpenID as a JIT-provisioning protocol.

OAuth allows a user of one service (called the provider) to authorize another service (called the consumer) to connect to the provider on behalf of the user and use some of its functionality. OAuth is part of the so-called “Open Stack” and is driven by a community of developers overlapping with the communities behind OpenID and OpenSocial. These standards tend to be community driven (although some big vendors are now supporting these initiatives), fast changing, focussing on single aspects and limited use cases of the identity management process, and relatively simple to support from a technical perspective.

OAuth can be used to facilitate provisioning as well, and has as benefit that the user is typically asked for consent before initiating a transaction. OAuth facilitates provisioning by granting access to an API at the OAuth provider. This could work as follows: When the user first goes to the SP, the SP can initiate the creation of an OAuth connection with the IdP (or federation hub), facilitated by the user. With this connection between the parties established, the SP can now retrieve the data it needs. OAuth does not define a specific protocol that can be used for attribute queries; instead it is a protocol framework that allows any REST-based request-response protocol to be able to use OAuth for session setup. Federations will need to specify their own REST-based attribute query protocol.
Usage of OAuth by itself does not result in an interoperable provisioning standard, as it does not specify the actual API that each provider need to support. Moreover, the underlying trust model of OAuth is based on the registration of OAuth Consumers (i.e. SPs) at the Attribute Authority. Clients will then obtain a “key + secret” that ensures trusted communication between a client and the Authority. This would be independent of the identity federation trust model and would consequently not impose any changes to existing federations. As the existing SAML trust fabric may not be easily re-used, the OAuth trust model would not benefit from the already established scalable trust management in federations. As an alternative approach, and in order to make the trust-model scalable, acceptance of new SPs could be left to the end-users. Since OAuth is still under development and appears to gain momentum, it may inspire other standards focusing on specific applications of OAuth\textsuperscript{21}. This may lead to OAuth, in combination with other standards running on top of it, to become a feasible standard for implementing provisioning\textsuperscript{22}.

So, it seems that several standards can be used for federated provisioning and that each one has its pros and cons:

<table>
<thead>
<tr>
<th>Standard</th>
<th>Advantage</th>
<th>Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP (or its XML cousin DSML)</td>
<td>already in place</td>
<td>central directories do not scale well, security is problematic, no deprovisioning of user application data</td>
</tr>
<tr>
<td>SPML</td>
<td>designed for provisioning</td>
<td>Not widely deployed/supported, intra-organisational minded, also does not deal with trust</td>
</tr>
<tr>
<td>SAML</td>
<td>already in place (in a federation), flexible</td>
<td>attribute exchange is not the primary purpose, not designed for bulk attribute sharing (synchronous, iterating through search responses etc.), no deprovisioning of user accounts or user application data</td>
</tr>
<tr>
<td>OAuth</td>
<td>possibility for user consent, open specification</td>
<td>specification is moving target, typical use cases in social networking sites (not designed for provisioning, different ad-hoc API calls for each IdP), support in other applications unclear, no deprovisioning of user accounts or user application data</td>
</tr>
</tbody>
</table>

Pinpointing a single standard is not trivial and may even be impossible as becomes clear in the following sections.

\textsuperscript{21} An example is User-Managed Access (UMA). See \url{http://kantarainitiative.org/confluence/display/uma/Home}.

\textsuperscript{22} See also \url{http://blog.talkingidentity.com/2009/11/can-oauth-do-what-spml-haasnt.html}. 
4.2 Federated provisioning literature

From the blog discussion shown in Chapter 1 one can conclude that federated provisioning is still being discussed. Federated provisioning has also been the topic of some white papers.

PingIdentity has a white paper on federation and provisioning [18]. The paper distinguishes two dimensions according to which provisioning solutions can be characterized: Trigger indicates when to provision, which can be either Just-in-time or Batch. Multiplicity indicates how many users to provision at once, which can be either Single or Bulk. The two dimensions are presented in Figure 3 (taken from that white paper). The size of the spheres is an indication of how suitable the combination of trigger and multiplicity is. The central conclusions of the PingIdentity paper are that a combined approach featuring both (SAML-based) federation and (SPML-based) provisioning can be beneficial to organisations looking for cross organisational provisioning, and that JIT provisioning with SAML (provisioning triggered by users signing on to an SP) is easily implemented in an existing identity federation.

![Figure 3: Multiplicity against trigger. Taken from 'Federated Provisioning: The Synergy of Identity Federation and User Provisioning', PingIdentity 2007.](image)

SURFnet’s Niels van Dijk in a report [1] based on Mikael Linden’s memo [13] on the subject of federated provisioning makes a similar characterisation of provisioning solutions. Van Dijk and Linden distinguish between: manual or automated, push (from the IdP) or pull (by the SP) and on-the-fly (JIT), one-by-one or mass.
Table 1: Different provisioning scenarios according to [1]. Here ‘X’ stands for suitable, and ‘XX’ stands for very suitable.

<table>
<thead>
<tr>
<th>Provisioning</th>
<th>Deprovisioning</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Provisioning scenarios</td>
<td>Deprovisioning</td>
</tr>
<tr>
<td></td>
<td>Mass</td>
<td>One-by-one &amp;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Just-in-time</td>
</tr>
<tr>
<td>On-the-fly using SAML</td>
<td>X</td>
<td>XX</td>
</tr>
<tr>
<td>Manual push</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Automated push, bilateral</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automated push, federation assisted</td>
<td>XX</td>
<td>XX</td>
</tr>
<tr>
<td>Pull, SAML Attribute</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Query</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Push, SAML NameID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Management</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Van Dijk and Linden consider provisioning, deprovisioning and a hybrid mix of the two (i.e., deprovisioning followed immediately by provisioning, e.g. when a user changes identity provider) and examine different ideal solutions for these processes when applied in an identity federation. Linden focuses on general identity federations, while Van Dijk focuses on hub’n’spoke federations. Van Dijk’s conclusion is that, within the context of a hub’n’spoke federation, for all processes a federation assisted scenario (which uses provisioning standards such as SPML, but re-uses the established trust provided by the identity federation’s infrastructure as much as possible) will work best.

4.3 Classifying federated provisioning scenarios

This section deconstructs the PingIdentity and SURFnet approach in an attempt to come up with a unified model. A number of different triggers for provisioning and deprovisioning (the notion of trigger is inspired by the PingIdentity paper) are defined. And a number of different application archetypes are defined.

4.3.1 Triggers

A federated provisioning request is triggered by one of the federation parties: the user, the IdP or the SP. For the different triggers the precise transport mechanism, be it SAML, SPML, LDAP, or some other proprietary format, is not important (although in some cases a candidate choice may be obvious). In the diagrams in this section triggers are indicated with dashed arrows with circle-shaped arrowheads, while attribute exchanges are indicated with solid arrows.
4.3.1.1 **Sign-on attempt of user at SP ("JIT")**

- A successful sign-on of the user at the SP triggers the IdP to provision the SP with updates for the user.
- A clear advantage of this method is that it uses the standard federated sign-on flow (i.e. attribute exchange).
- A disadvantage is that limitations of the federated SSO protocol/standard are inherited. The SSO protocol/standard is not meant for provisioning and consequently may lack expressiveness, may not always take into account all user attributes as the focus is on authentication/SSO, and may lack provisioning performance.
- Another disadvantage is that for some applications attribute exchange on sign-on may simply not be frequent enough. Deprovisioning may be a problem for certain applications, for example, as sign on attempts may not occur frequently enough.

4.3.1.2 **Certain use of (certain) identity attributes at SP ("Use @ SP")**
• The SP could pull updates for certain user attributes from the user’s IdP (e.g. via a SAML assertion query) at any moment in time that the SP thinks is appropriate. The SP, after all, knows best if and when up-to-date attributes are needed inside its processes.
• For this to work, the IdP needs to implement an end-point (call-back) interface. And this needs to be known to the SP.
• An advantage is that up-to-date attributes will always be available for a critical set of attributes to be determined by the SP.
• A disadvantage is inefficiency, this generates a great amount of events; the IdP may be queried for up-to-date attributes even if attributes never change.
• Another disadvantage is that the SP becomes rather complex. It is no longer a simple provisioning target, but takes the initiative to initiate provisioning actions. This requires a secure communication back-channel between the IdP and SP which is not trivial (i.e. involves proper certificate management) and complicates the possibility to ask for user consent. There is no defined SAML2.0 profile where the attribute query protocol is allowed to be used with front-channel bindings.

4.3.1.3 Change in (certain) identity attributes at IdP (“Change @ IdP”)

• The IdP notices that some attribute has changed and decides to push updates for that user to all relevant SPs. The IdP, in its role as attribute authority, is the first to know when attributes change.
• This time the SP needs to implement an end-point (call-back) interface for this (however, a simple SPML add/modify/delete interface is sufficient), and this interface needs to be known to the IdP.
• The federation operator (the hub) may assist in selecting which SPs are relevant (in case the IdP does not keep such state for its users). Some form of subscription service could be implemented on the hub. This makes the hub stateful though.
• An advantage is that up-to-date attributes are immediately available to SPs that need them. This method has the highest possible granularity, i.e. having the right attributes at the right time.
A disadvantage is that this may generate a large amount of events. If attributes change frequently and SPs have no need for up-to-date attributes for some or most of the users, this leads to unnecessary traffic. Another disadvantage is that already running processes at the SP may not be able to take updated attributes into account during these processes.

4.3.1.4 Timer at IdP or SP ("Timer")

This is traditional “batch” provisioning. The IdP pushes updates for all users to all (interested) SPs at regular intervals (every hour, every night, every month, ...).

A clear disadvantage of batch provisioning is that attributes may be outdated.

Alternatively an SP (or some third party acting on behalf of the SP) could pull updates for all users from all IdPs at given times, i.e. the timer process can generate trigger events at either the IdP or the SP.

An advantage is that the amount of traffic is reasonable predictable and can be controlled.

Another disadvantage is that it always uses a back-channel, thus there is no possibility to ask for user-consent.

4.3.1.5 Expiry of attributes at SP ("Expiry")

This is traditional “batch” provisioning. The IdP pushes updates for all users to all (interested) SPs at regular intervals (every hour, every night, every month, ...).

A clear disadvantage of batch provisioning is that attributes may be outdated.

Alternatively an SP (or some third party acting on behalf of the SP) could pull updates for all users from all IdPs at given times, i.e. the timer process can generate trigger events at either the IdP or the SP.

An advantage is that the amount of traffic is reasonable predictable and can be controlled.

Another disadvantage is that it always uses a back-channel, thus there is no possibility to ask for user-consent.
• When attributes carry expiration dates (or time-to-live) the SP can decide to only pull updates from the IdP for those attributes that have expired.
• Typically used in conjunction with some other trigger (in the figure the first trigger is a JIT trigger, which sets the timer, which causes the second trigger)
• An advantage is that this solves the problem of users who never sign in. I.e., it solves the deprovisioning-resource problem: after expiration of a certain time the account will be removed and its contents deleted. If SPs throw away the attributes after expiration, then it will be very privacy preserving as well.
• Disadvantage: not possible to get consent from user. Consequently, it may harm the privacy of the user as SP may start pulling attributes before expiration time.

4.3.2 Application archetypes

Not all applications at all SPs need up-to-date values for all attributes all of the time. Some applications will need user attributes before any action can take place, before users even get the chance to sign in to the application. Other applications can be provisioned just-in-time as suggested in the literature discussed in Section 4.2.

It does not matter why an application needs user attributes or how these are used, what matters is that an application may need up-to-date attributes at certain times in order to perform its job.

4.3.2.1 Archetype I ("No identity information needed")

Archetype I applications have no persistent user state, and no need for intra-session attribute updates.

For this archetype it is natural to not store any persistent identity information about the user at all. This means that identity information is transferred to this type of application from the IdP at the start of a session (i.e. at sign-in time).

As an example, consider a publisher of scientific texts with contractual agreements with some IdPs (but not all) in a federation for higher education. When users approach the SP they are redirected to their IdP to authenticate. The only attribute of interest is whether the authentication succeeded at an IdP that warrants access to the publisher. There is no need for the SP to remember any identity attributes. The next time a user returns to the SP this process repeats itself.
This may be considered as a very ‘pure’ application archetype from an identity management perspective. All relevant persistent attributes have been externalised from the SP and reside at the IdP only. No persistent user state needs to be stored at the SP. Many current web-based services in federations are of this simple archetype, essentially using only the authentication features of the federation. This may change as the need of applications for up-to-date attributes changes, or when these applications add more personalization features. An example of the latter is when a publisher add recommendations based on previously accessed content.

4.3.2.2 Archetype II ("JIT provisioning is sufficient")

Archetype II applications do have persistent user state, but no need for intra-session or extra-session updates of attributes.

Even though this archetype does store identity attributes at the SP, there is no need to get more frequent updates than the sign-in points in time.

An example of an application in this category could be an e-commerce website (say, a web store that sells software products as part of a federation of higher education where students and staff of certain IdPs get discounts). At this SP signed-in users can look at previous transactions or items they have put in their shopping cart during a previous session. Up-to-date attributes for a specific user are only needed when that user signs-in.

There is no need for the SP to handle user data outside of user sessions.

4.3.2.3 Archetype III ("Identity information needed during user session")

Archetype III applications with persistent state, and a need for intra-session up-to-date attributes (for the current user only).

This archetype does store persistent identity information of users and needs to make sure, during the user session that some attributes are up-to-date.

An example could be a presence or location service, or a blog / forum website which also sends notifications to registered users.

Another example of an application in this category could again be an e-commerce web site which sells products for which the user needs to pass some criterion that can only be validated by querying the identity provider in real time (e.g., applications which employ some form of step-up authentication fall in this category).

4.3.2.4 Archetype IV ("Identity information needed, anytime")

Archetype IV applications do have persistent state, and a need for intra- and extra-session up-to-date attributes.
This archetype needs to make sure that attributes are up-to-date independent of user sessions. In some cases up-to-date attributes are needed for processes at the SP before the user even gets his or her credentials. Or up-to-date attributes of other users, than the user that is currently using the service, are needed.

An example could be a resource planning or workflow system which needs to assign certain tasks to users even if they have not signed in to the system recently (or ever).

Another example is an application in which a user can send an email to certain, perhaps dynamically created, groups of users. The application needs to be provisioned with the most recent email addresses of these users, as known only by the IdP. This specific example is explored in Chapter 5.

### 4.3.3 Archetype assessment

It is difficult to assess the number of applications that fall into each category. Many current services in the SURFfederatie will probably fall into category I or II and can be provisioned in a JIT-style manner, yet it is unclear whether this is the case because of lack of more advanced provisioning techniques or if there is really no need for category III or IV applications.

In Table 2 the different provisioning triggers and applications archetypes have been plotted against each other. In simple cases JIT is sufficient for provisioning. In some cases, for example where JIT can be “abused” to codify complex inter-user relationships such as for example groups, it is possible to follow a JIT strategy but at a very high cost. Chapter 5 describes an example case in more detail.

<table>
<thead>
<tr>
<th>Table 2: Provisioning triggers and application archetypes.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>JIT</strong></td>
</tr>
<tr>
<td>-----------------------</td>
</tr>
<tr>
<td>OK</td>
</tr>
<tr>
<td>Use @ SP</td>
</tr>
<tr>
<td>Expiry</td>
</tr>
<tr>
<td>Timer</td>
</tr>
<tr>
<td>Change @ IdP</td>
</tr>
</tbody>
</table>

The “Expiry” strategy can be a suitable for “Use @ SP” but it depends on how well expiry times are chosen and how high the demand of the SP for true real-time up-to-date information is. The “Timer” strategy can be adopted by all other application archetype offerings, again depending on the resolution of the timer, and against a price in performance (it could generate vast amounts of unnecessary interactions).
For the more advanced categories of applications JIT is not sufficient, but this does not mean that federation assistance is not needed. In fact, federated sign-in attempts can be used, for instance, to reset expiry triggers (for “Expiry” trigger style provisioning) or to communicate interface end-points (for “Use @ SP” and “Change @ IdP” style provisioning) to the relevant parties. For instance, users can be forced to login after e.g. half a year of absence otherwise their account will be deleted.

Table 3: Triggers and provisioning standards.

<table>
<thead>
<tr>
<th>Triggers</th>
<th>Standards</th>
</tr>
</thead>
<tbody>
<tr>
<td>JIT</td>
<td>SAML, OpenID</td>
</tr>
<tr>
<td>Use @ SP</td>
<td>LDAP, SPML (Search), or OAuth (2- or 3-legged), or SAML assertion query/request</td>
</tr>
<tr>
<td>Change @ IdP</td>
<td>SPML (Add,Delete,Modify), or OAuth (2-legged)</td>
</tr>
<tr>
<td>Timer</td>
<td>See “Use @ SP” or “Change @ IdP”</td>
</tr>
<tr>
<td>Expiry</td>
<td>Depends on other trigger</td>
</tr>
</tbody>
</table>

There is a clear correspondence between the different triggers in Section 4.3.1 and the different standards in Section 4.2. Table 3 lists the most likely standards to be used for each trigger type.

4.3.4 Provisioning within a hub’n’spoke federation

Some federations work with a central entity which connects IdPs and SPs. Such a central entity, called a hub in this report but sometimes also referred to as federation operator, serves as a facade to both IdPs and SPs. From an IdP’s point of view the hub is an SP, from an SP’s point of view the hub is an IdP.

The hub is run by a trusted third party. The hub itself facilitates trust between these parties, i.e. connected parties no longer need to setup security infrastructure (certificates, etc.) but can use the hub to communicate. The hub’s primary goal is to make identity federation easier, e.g. it facilitates protocol translations.

The hub could also play a role in provisioning of SPs. The main conclusion from [1] is that federation assisted provisioning works best in a hub’n’spoke federation. The hub can, for instance, keep track of which users have authenticated with an IdP in order to use the services of which SPs. This helps in scenarios where the IdP needs to push updates to only relevant SPs which is the case for archetype III and IV applications. This, however, requires the hub to maintain state about users and the services they have used and conflicts with the desire of many hub managers (including SURFnet) to minimize state. An example from e-Science virtual organisations where a hub plays a central role is explored in Chapter 5.

Provisioning scenarios in identity federations

27
4.4 Summary

Different applications have different needs for the timeliness of identity information. For some applications just-in-time provisioning is sufficient. Other applications require a great deal of up-to-date user information without the associated users even being signed in. For the latter category of applications additional out-of-band provisioning is needed. Both the IdP and the SP are candidates for generating the trigger for initiating such an out-of-band provisioning process. The SP knows whether up-to-date attributes are really needed (and can pull updates); the IdP, in its role as attribute authority, knows when attributes were last updated (and can push updates). Attaching expiry information to attributes may help the IdP or the SP to determine when updates are necessary.

That out-of-band provisioning cannot be fully replaced by JIT provisioning does not mean that provisioning and deprovisioning cannot benefit from already deployed identity federations. In situations where a hub’n’spoke identity federation is deployed, the central entity can be used as a provisioning service point. The hub can, for example, keep track of which SPs need provisioning in case of IdP trigger provisioning.
5 Application of provisioning within e-Science

Dynamic group management [3] is important for many e-Science services [7]. The ability to form ad-hoc groups of researchers is a key enabler for collaboration between researchers at different sites. Federated identity management is an essential enabler for dynamic group management. Federated provisioning of group-related attributes is expected to be one of the first problems that need solving before such collaboration scenarios can be implemented.

In higher education and academic research institutes federated identity frameworks for authentication are already deployed. If existing identity infrastructures at the traditional scientific institutes (i.e. the IdP members of a federation) can be leveraged to efficiently provision collaboration efforts, which are often called virtual organisations, then dynamic group management is made much easier.

Group management involves having the capabilities to define and maintain groups, and re-use this for different purposes/applications including authorisation, e-mail lists, phonebooks, etc. Group applications need to be provisioned with this information. Contrary to other attributes, group membership is not something determined by the IdP/university administrators, but by a researcher, and are distributed over multiple institutions. "Normal" provisioning approaches as discussed earlier in this report therefore can not simply be used.

5.1 Dynamic groups and virtual organisations

A virtual organisation can be created by an individual user within a federation. That user becomes the group manager and can invite other users (as long as their home institution is part of the federation) to join the virtual organisation.

Provisioning plays an important role here. Attributes need to be up-to-date irrespective of the login behaviour of the individual users (e.g. daily, weekly or never). Moreover, as many other e-Science relevant services need to know of the group context, provisioning this context automatically relieves the user from the burden of creating the same group at every service.

A typical scenario for creating a virtual organisation is the following:

- A researcher has access to an important resource at his home institution and wants to share this access with fellow researchers at other institutes (possibly those institutes are not part of the same federation)
- She creates a virtual organisation and thereby becomes the manager of it
- She sets the virtual organisation access policy so that members have the rights to access the resource
- She invites colleagues from other institutes to join the virtual organisation, perhaps sharing her role as manager with some of them

Some characteristics of virtual organisations, which make them fundamentally different from traditional organisations:
• Relatively small groups
• User attributes will be distributed across different identity providers
• Self-service (a mere ‘researcher’ performs administrator tasks)
• User-consent via invitations
• Shorter lifecycle

The self-service and user-consent character might make new standards such as OAuth [9] suitable. Furthermore, a technique like virtual directories may be appropriate as well, especially since the groups are relatively small.

5.2 Provisioning of groups in a hub and spoke federation

As a case study consider the following problem. The hub-based SURFfederatie has a new groups service called ‘SURFteams’. Any user within the federation can start a group by inviting other users. The initial user adds the email addresses of the users he or she wants to invite to a list and the “Groups service” sends emails asking the users to authenticate with their IdP and confirm that they want to join the newly formed team. This application falls into archetype III “Identity information needed during user session” of Chapter 4.

Other SPs in the federation may also have an internal notion of group, for example so as to allow management of privileges within that service on group level. Those SPs should be completely provisioned in a federation assisted way via the Groups service as well.

If a JIT provisioning strategy is chosen, this can be simply implemented by having the hub interact with “Groups service” during every sign-on and have it add an attribute for each group that the user is a member of. The hub thus alters the attribute assertion sent to the Collaboration service SP. This is depicted in Figure 9.

Figure 9: JIT provisioning of user attributes and group information during login via SAML.
For other applications, the complete group needs to be provisioned to the SP before sign-on can take place. For example, the SP may offer an application to send an email to all group members. That would be a good trigger for the SP to make sure that it has the most recent email address of all users.

This could be implemented as follows:
- The hub adds an attribute for each team that the user is a member of to the attribute assertion during sign-on, just like in the JIT case
- The hub also adds an SPML end-point in an attribute
- When the SP needs up-to-date email addresses it sends an SPML Search request to that end-point
- The SP iterates through all the responses, and updates its internal database of email addresses

Note that the JIT approach only works well in the presence of a hub that keeps track of the groups and their members.

Figure 10 shows the message flows of such an SPML-based scenario. Note that providing the SP with an SPML end-point is very similar in strategy to the approach taken in Liberty Alliance’s Interaction service [11].

Also note that the SP, which needs to be group-provisioned (the Collaboration Service), plays the role of an SPML RA instead of a target. This is somewhat counterintuitive at first: the SP is a target (but not in the sense of SPML), which provisions itself by sending complex search queries to an authority, which controls attributes (while this attribute authority is an SPML target). Perhaps this can be changed so that the hub takes on the responsibility to perform the search queries and in turn uses much simpler add or modify requests to provision the Collaboration service instead. Such a scenario would make the hub much more complex, of course, yet would allow service providers to be provisioned by implementing a relatively simple SPML interface.
An alternative solution would be to bypass the hub completely and facilitate direct provisioning communication between the Collaboration Services and "Groups service" (see Figure 11). Again, the SP that needs to be provisioned would then need to discover the SPML endpoint in a Liberty Alliance Interaction service style transaction.

The hub could play a role in this discovery process or the SP could use the federation metadata that includes all the end-points of the federation members (including amongst others the Groups service). The latter approach may motivate the inclusion of specific provisioning end-points in the metadata (besides SAML end-points).

![Figure 11: Direct provisioning scenario.](image)

### 5.3 Summary

In several cases the IdP does not have all the necessary attributes that relate to a certain identity. Group-related attributes are examples of such attributes and are typically managed and provided by another provider, i.e. the Groups service. The existence of such a separate attribute service provider complicates the identity and associated attribute provisioning.

By focusing on a concrete use case, i.e. an e-Science collaboration service which needs dynamic groups provisioning, it becomes clear that there are many possible technical solutions to tackle the provisioning needs of such a service. Each of these solutions has advantages and disadvantages. Naive JIT-style provisioning is likely not a very efficient solution as great amounts of unnecessary data would need to be transferred on sign-on events of each and every user.
That is not to say that provisioning in such a use case cannot benefit from the identity federation infrastructure that is already in place. In all solutions a central coordinating entity (i.e. hub) has a role to play in provisioning. In the simplest scenarios the hub is used to “discover” the relevant targets acting as attribute authorities for other parties that act as requesting authorities. The flow of events in such cases is somewhat reminiscent of Liberty Alliance style identity management. In other situations the hub plays a more prominent role.

Combinations of provisioning with other standards (OAuth, XACML) have not been studied in this chapter, but it may very well be the case that, for applications which use these technologies, the motivation behind architectural decisions changes.
6 Conclusions

User provisioning and deprovisioning is a key part of the identity management process. Despite the existence of open and industry backed standards (such as SPML) it remains a challenge to implement automatic provisioning and deprovisioning in real life scenarios in a maintainable way. Where intra-enterprise provisioning based on these standards is already difficult, cross-organisational provisioning is even more difficult primarily because trust aspects largely fall out of scope of these standards. Some analysts even doubt that the standards are mature enough, even in situations where trust is not an issue. Also semantic aspects are more difficult in a cross-organizational case as in an intra-enterprise case.

Standards for identity federations do in general support trust aspects in ways the provisioning standards do not. Federation standards such as SAML could therefore in principle be of great help in setting up provisioning solutions in a federated context. In such a context, certain identity information will already be externalized from the SPs towards identity providers. This means that identity federations make provisioning less urgent: in some cases so-called just-in-time provisioning (based on attribute exchange, immediately following authentication) is sufficient.

However, as a study into typical application archetypes in Chapter 4 shows, it depends very much on what kind of provisioning interactions are required by the application, and what information about identities is used within such interactions, whether provisioning based on just-in-time triggering is enough. If an application requires a great deal of up-to-date user information without the associated users being signed in (this application archetype is referred to as Archetype IV in Section 4.3.2), then it is not enough and additional out-of-band provisioning is needed. It is expected, as federations support a wider range of applications, that the combination of federated identity management and provisioning such non-trivial applications will need addressing.

For many current web-based applications, just-in-time is sufficient. However, most collaboration-based applications fall into the non-trivial category of applications. This category includes e-Science applications that enable collaboration between scientists in so-called virtual organizations. These virtual organizations are typically deployed by SPs who rely on IdPs (the home institutions of the scientists). A group service case study (worked out on the conceptual level) shows that by combining, for example, SPML and SAML it is possible to do group (virtual organisation) provisioning, though one can expect that the specific character of this combination will be different for different cases. The use of OAuth for such application specific provisioning purposes seems promising but needs more research when used in the context of identity federations.
References

[1] Bargh, M.S., Scalability of trust and metadata exchange in confederations, GigaPort3 deliverable EDS-P14, work in progress, 2010
[7] Hulsebosch, Bob, Use Cases and Requirements for an e-Infrastructure, GigaPort3 deliverable D1.1.1, June 2010
[10] Liberty Alliance Project, Liberty ID-SIS Employee Profile, Kellomäki, Sampo and Lockhart, Rob (eds.), Version 1.1, 2005