------ Excerpt ------

A Roadmap for SPML

------ Excerpt ------

A Roadmap for SPML

Rami Elron, Senior System Architect, Security Business Unit, BMC Software

Over the last couple of years, we have witnessed a steady and significant increase in the importance of provisioning. The term known by very few not too long ago is by now an important piece in the enterprise IT jigsaw, and provisioning software proves to be an indispensable tool for supporting and monitoring daily business operations in the burgeoning marketplace. Having said that, the utilization of provisioning applications in organizations is merely a first step towards realizing the enormous potential provisioning promises. Its importance is mostly evidenced in managing internal company data, but we approach the point where provisioning services are required on a much broader scale. The significance attributed to global digital identity, web services and secured access to such services is a testament to this trend and an additional proof that the key success factor for a provisioning vision lies in two things: standards and interoperability.

Provisioning – so far

Provisioning may be roughly described as the definition, management and automation of processes, which govern the lifecycle of computer-stored identities. These identities consist generally of users, but various implementations include resource identities as well. Implementation of provisioning solutions varies from one company to another, but most share a common objective – to address issues concerning administrative overhead and security risks inherent with management of multiple account repositories. Hiring a new employee often necessitates the creation of multiple user accounts in various applications and operating systems, according to the relevant job description. During the course of employment, account properties need to be changed occasionally to accommodate new job descriptions and new responsibilities. Upon job termination, it is required to immediately revoke (and possibly remove totally) relevant user accounts to block further access to restricted company data and resources.

These needs and more have spawned a long list of provisioning vendors and solutions. However, lacking hitherto any standard definition and clear scope, provisioning has evolved to encompass today many aspects of the 'Identity Management' space, including password management, access management, and more. Add to that a proprietary service API for each vendor's offering and it comes to no surprise that even minimal interoperability between implementations is questionable at best.

A Vision for Provisioning

Today provisioning is generally associated with 'User provisioning' or 'Account Provisioning'. In the future though, it is not inconceivable that provisioning will be used in the broader context of 'Service Provisioning'. One could likely envision an information world with service requestors, service providers and service brokers (constituting a

hierarchy of service) – all identifiable, and exploiting a common, standard, effective, secure and scalable framework to exchange provisioning requests.

SPML stands for Service Provisioning Markup Language – a specification created by the OASIS Provisioning Services Technical Committee (PSTC). The SPML initiative was born to answer an urging business need – to offer a standard, expressive way to convey and exchange provisioning data and operations between communicating parties. In light of the previous paragraphs though, it is much more than a mere dialect. The specification is designed to play a vital part in stimulating adoption of provisioning best practices. Thus, it comes to no surprise that SPML should be regarded as nothing less than a standard framework to manage provisioning services in the future marketplace. This becomes more apparent as one considers the impact provisioning has had on businesses so far.

Current provisioning implementations often do not have much in common, even from the standpoint of objectives. Whereas one company uses provisioning software to enable central management of identity-based data, another might require such a solution for development purposes or even auditing. Much of this stem of the fact that provisioning is neither a trivial term nor an easy concept to grasp and it suffers from proliferation of 'meanings'.

The current SPML specification is the first step in a path to establish a comprehensive provisioning framework for global interoperable services. The SPML committee members made every effort to ensure that even in its first version, the specification supports the following:

- Ability to build effective solutions for client practical needs
- Flexibility to incorporate changes necessary to accommodate changing market requirements
- Optimal utilization of existing and proven standards wherever possible
- Freedom for service providers to offer extended functionality without breaching the standard specification
- Interoperability among service providers

Interoperability may mean different things to different people, but in the context of a provision-esque language specification it probably translates best to letting requestors submit requests to a service point with minimal (or no) concern with the particular provider's service implementation. One approach for a solution is to standardize on an appropriate set of verbs that mirror real world provisioning requests. This approach has much appeal since it theoretically allows a requestor to use a single format to request a given service from multiple service providers. A different approach is to restrict provisioning verbs to a set of 'core' operations that act on a standard schema where specific object classes and attributes mirror common provisioning scenario objects. So in essence one approach favors the mirroring of provisioning 'verbs' whilst another favors

mirroring of provisioning objects. Alas, both 'pure' approaches do not lend themselves easily to accommodate every imaginable provisioning scenario - at least not without necessitating some sort of extension mechanism to be included in the model. Such an extension could allow any provider to still offer requestors with access to non-standardized functionality. Of course – there is no interoperability in this solution – bringing us back to square one. But there IS a solution. By combining the best of the aforementioned approaches and introducing some necessary mapping functionality it is quite possible to cover all the aspects needed to solve the interoperability challenge. These pieces include (1) a data schema specifying PSTC-approved provisioning object classes and relevant attributes; (2) a core set of generic operations (e.g. add, modify, delete), which can be used independently, or sequenced to form complex workflows acting on the data schema objects; (3) an ability to let providers extend both data schema and operation 'verbs' to offer clients with non-standard proprietary provisioning services; (4) sequencing functionality to link several 'core' commands together to form complex commands, without resorting to proprietary, non-interoperable verbs; (5) discovery mechanism to enable providers to expose the details of their service options so requestors can learn about them and utilize them accordingly; (6) mapping specifications to link proprietary verbs with standard verbs, proprietary objects/attributes to standard objects/attributes.

The PSTC decided to pursue the vision for interoperable provisioning in a phased approach, starting with the implementation of a core operations model in which provisioning operations are manifested by applying generic operations on a standard data schema.

The committee has devised a work plan focusing on the achievement of the following objectives:

- Reach accepted definition of "provisioning"; preparation of a mission statement for the provisioning service markup language (SPML)
- Scope provisioning solution to be addressed by the specification
- Define roadmap for achieving the goals set in the vision statement
- Define deliverables of first phase major milestone – SPML version 1.0
    - Identify business scenarios where provisioning play a recognizable role
    - Identify roles and operations in provisioning scenarios
    - Typify provisioning scenarios
    - Identify services pertinent to provisioning scenarios
    - Define logic model for provisioning solution
    - Identify and review relevant existing standards
    - Define functions to support elementary provisioning services

- Define protocol for submittal and reception of provisioning requests/responses
- Review protocol
- Build prototype for feasibility/validity testing and demonstration
- Submit version 1.0 for approval

- Initiate work on roadmap phase 2

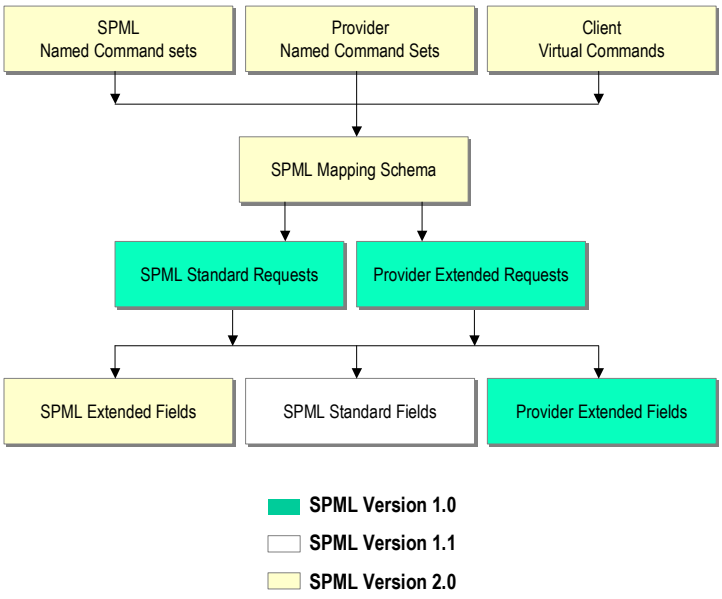

Figure 1: SPML functional blocks

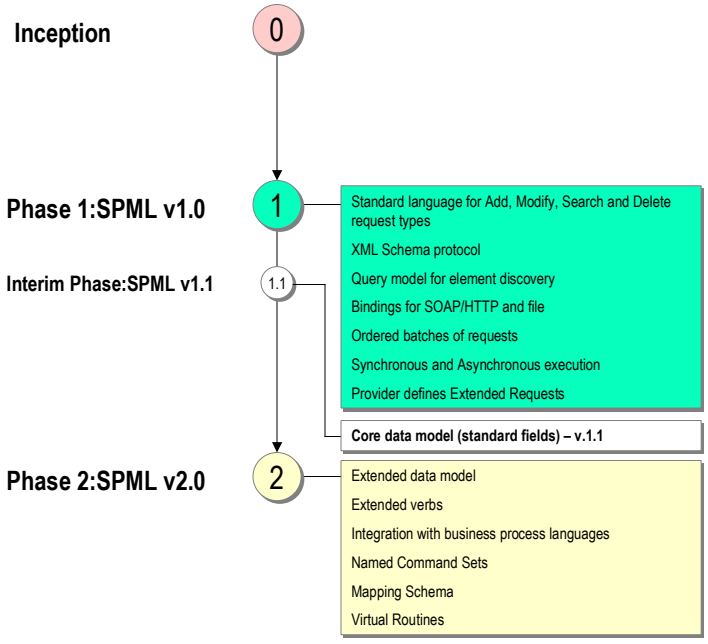

Figure 2: SPML roadmap

The functional specification of SPML v1.0 lays the foundation for realizing the provisioning vision by defining a provisioning lingua franca. This offers clients and service providers with the means to express provisioning directives and pertinent information in a standard fashion, using straightforward verbs. While it borrows various aspects from proven standards, SPML is a genuine protocol sporting features intentionally designed to support provisioning scenarios as befits a provisioning-oriented standard. While planned to be addressed fully in the next version, interoperability is maintained to an extent thanks to the core standard schema included in the SPML specification.

The SPML version 1.0 specification features the following:

- XML schema-based protocol for exchanging provisioning messages between a client and a service provider.
- Core provisioning operations for realization of elementary 'micro-provisioning' data services – addition, modification, deletion and search
- Query model providing a service-requestor (client) with an ability to discover details about provisioning data and operations he is authorized to request with respect to a given service provider

- Ability for a service provider to define and implement Extended Requests – specially constructed verbs for operations not covered by the SPML specification, still conforming to standard rules.
- Definition of batch requests – collection of operations requested to be handled as a single entity of operation
- Support for synchronous and asynchronous requests
- File and SOAP/HTTP bindings
- Core data model for maintaining an accepted level of interoperability

The SPML version 2.0 specification will improve on its predecessor in two main areas – the data model and verb model. Focusing chiefly on interoperability, the version 2.0 specification will feature a broader core data model for enabling a higher degree of interoperability than possible before. To abet this, SPML 2.0 will specify two mapping protocols – one for data and one for verbs. The data mapping protocol will enable service providers to support standard SPML field arguments while implementing a different proprietary data model. The verb mapping protocol shall enable a proprietary request verb to be used in lieu of a corresponding standard request. Coupled with a new functionality to support naming of command sets, it will be possible to create complex 'macro-like' commands consisting of (ergo – mapped to) ordinary 'core' commands. The core set of commands defined in v1.0 will be enhanced as well.

In addition, SPML v2.0 will be revised to accommodate new emerging standards available by then (e.g. BPEL).

The functional specification of SPML version 2.0 will address the following:

- **Extended and expanded data model scheme – new and revised fields**
  SPML 2.0 will build on the framework presented in SPML 1.0 and 1.1 while adding new functionality to support creation, management, discovery and mapping of "field namespaces" – named definition sets of fields relating to a particular market segment. Such functionality could allow separate parties to agree on a standard 'field set' and refer to it in a standard manner, without necessitating the involvement of a standards body to incorporate the specific data into a common schema. The role of SPML is to specify a standard for the creation, format, publication, discovery and usage of such data.
- **Extended scheme for standard verbs**
  SPML 1.0 provides a powerful way to express provisioning requests. However, its interoperability chiefly depends on the adoption of a standard schema specifying common and accepted objectclasses. It is reasonable to assume that in order to facilitate scalability and still ensure interoperability, some sort of federation or

distributed scheme should be used. SPML will address these cardinal issues in two aspects. From the data perspective, SPML 2.0 will specify a standard model for schema hierarchies and relations. From the operation side, SPML 2.0 will specify an extended set of standard verbs (in addition to the existing 'core' add, modify, search, delete), which will be in fact mapped sequences (see *named command sets)* of core verbs with 'pre-configured' attributes, albeit with options to let an implementer add (but not remove!) additional commands to the sequence. This way, there will be a common understanding of what basically constitutes a command, while still enabling extensions by service providers.

- **Support for field and verb mapping**
- **Support for named command sets**
  SPML 2.0 will support the creation of core verb sequencing – i.e. the chaining of add, modify, delete, search verbs acting on specified objectclasses, and using specified attributes – under a given name. This named request will be referenced in the schema and could be referred similarly to ordinary core verbs.
- **Expanded discovery mechanism for provider capabilities**
  SPML 2.0 will expand on the discovery specification of version 1.0 to enable requesting authorities to discover functional enhancements introduced in any new version of SPML in addition to the capabilities already supported before.
- **Accommodation of new relevant standards**

More On SPML Data And Verb Mapping

What is SPML data and verb mapping? Simply put, it is a means for enabling translation between any two SPML-supporting providers' proprietary (i.e. extended) *provisioning dialects* (PD) - using the standard SPML schema as a common denominator 'bridge'.

As the SPML schema is still in its infancy and evolving, it obviously does not address yet every possible provisioning scenario, and it is arguable whether ANY schema could ultimately provide such capability along with unanimous concurrence of all provisioning services to adopt it AS IS. In reality, service providers and vendors alike will each continue to strive to offer a smorgasbord of new features along with enhanced functionality that requires the addition of new, non-standard objectclasses or the use of extended requests, thus impeding the vision of interoperability.

There are compelling reasons to implement mapping. First and foremost is interoperability. Within the SPML provisioning vision, interoperability ultimately translates to two things:

1. Letting any *SPML-compliant requestor* (SR) use a single un-modified PD to request provisioning services from any *SPML-compliant provider* (SP) (e.g. submitting a single format *revoke* request for an account.).
2. Letting any given SP construct a single un-modified PD to communicate with any *SR* (e.g. responding uniformly to a *revoke* request submitted by non-uniform PDs).

Why is all this needed? Here are just a few examples:

1. An SR needs to access multiple SPs for a similar service, however each of the SPs has a different implementation of objectclasses and verbs. The SR does not want to implement multiple request dialects.
2. An SR wishes to replace the SP, sans modifying PD code.
3. An SR employs a complex request model and favors submittal of high-level requests that mirror this model over submittal of simpler 'atomic' requests. Moreover, the SR wants to receive responses correlating to such requests.
4. An SR wishes to use the services of a specific SP, but not at the expense of using proprietary requests and losing interoperability.
5. An SP wishes to implement extended functionality that is not covered in the SPML standard spec – but without resorting to non-interoperable Extended Requests.
6. An SP wishes to use a different term for specific objectclasses and verbs without breaching compliance to SPML standards.

To facilitate such capabilities, translation services are required, providing some sort of mapping between specific PD elements and SPML standard schema elements.

The following is a non-exhaustive list of questions that arise with respect to dialects:

1. Who is entitled to define a PD? Is every SR entitled to do so or is the 'creation' of a PD a prerogative given to specific parties? If yes – who is involved here? Who authorizes the PD? Who maintains the PD?
2. What kinds of verbs and data are allowed to be mapped? Are there exceptions, or is everything map-able?
3. How (if at all) is dialect mapping related to the SPML support for internationality?
4. Who provides the dictionary for a PD?
5. How is the dictionary constructed? What does it include? Who is responsible for the dictionary spec?
6. Who is responsible for performing the mapping (the Mapper) – is it the SR, the SP or a 3rd party specialized service?
7. How is the Mapper referenced in a provisioning message?
8. How does one discover Mapper services?
9. How are Mapper services described and accessed?
10. What is required from SPs to support PDs? What is required from SRs?

Obviously, finding an optimal solution to all those questions is not trivial, yet not necessarily a staggering undertaking either. Given the aforementioned issues, it is reasonable to regard the SPML schema not as THE provisioning sole dialect but as a core foundation for constructing provisioning dialects, furnishing service providers with the proper building blocks to express any complex sequence of requests and responses deemed necessary.

Such a mechanism would allow providers to construct their own custom dialect based on the SPML core building blocks. By having providers support a common standard mapping scheme, it is possible for customers to converse with any SPML-compliant provider.

* * *