Federated provisioning (Why and how)

Federation Boot Camp,
Vienna Feb. 05-06, 2018

Peter Gietz, DAASI International
Peter.gietz@daasi.de

Problem Statement

- Provisioning is the process of providing identity information to a target system (here an application secured by a SAML Service Provider)
- In SAML2 based federations provisioning is done „just in time“
  - User logs in at the IdP and IdP sends an assertion to SP that the user has correctly authenticated
  - IdP can also add assertions about user attributes (i.e. identity information)
- There are good reasons why this is not enough, we also need a „just in case“ (Peter Schober) provisioning

DAASI International GmbH

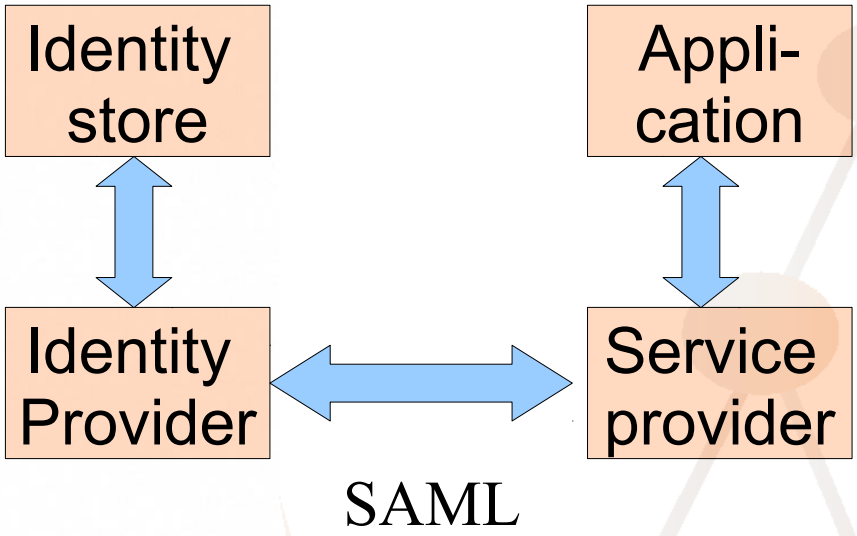
Use cases for „just in case“ (de-)provisioning

- The SP protected application needs to know membership information before actual first log-in, e.g.
  - E-Learning system needs to plan courses and needs membership information from Identity store.
  - Teachers may want to know how many people will join her course
- With just in time, the application will never know, if a user has left the organization
  - Application will have information about people it shouldn‘t have any information about any more (data privacy legislation!)
  - E-Learning system cannot plan courses correctly, because it has a wrong number of potential course members

DAASI International GmbH

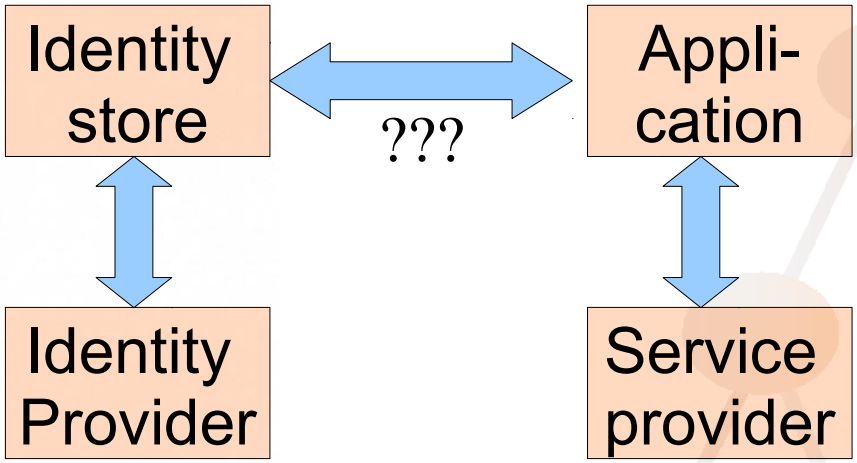
General Architecture

Just in time



General Architecture

Just in case



Provisioning strategies 1/2

- Push versus pull
  - Push:
    - Identity store sends information to the application
  - Pull:
    - application requests information from identity store

DAASI International GmbH

Provisioning strategies 2/2

- Mass versus one-by-one
  - Mass:
    - Identity store provides all identities at t once
    - Application has to do a diff with its own information and add, modify or delete identities respectively
  - One-by-one (only push):
    - When a new identity is added, modified or deleted in identity store it will tell this to the Application
    - Application adds, modifies or deletes the respective identity

DAASI International GmbH

Solution 0,5: The SAML way

- The SP could regularly check all identities it has with the IdP
  - SAML attribute query protocol
    - “Hey give me the attributes of subject 1234“
    - If attributes come, the user is still existent and the application can check, if data have changed
    - Otherwise the IdP sends „I don‘t know subject 1234“ and application can delete user
- This only works for the de-provisioning use case
- It is quite costly if application has a lot of users in terms of time and bandwidth

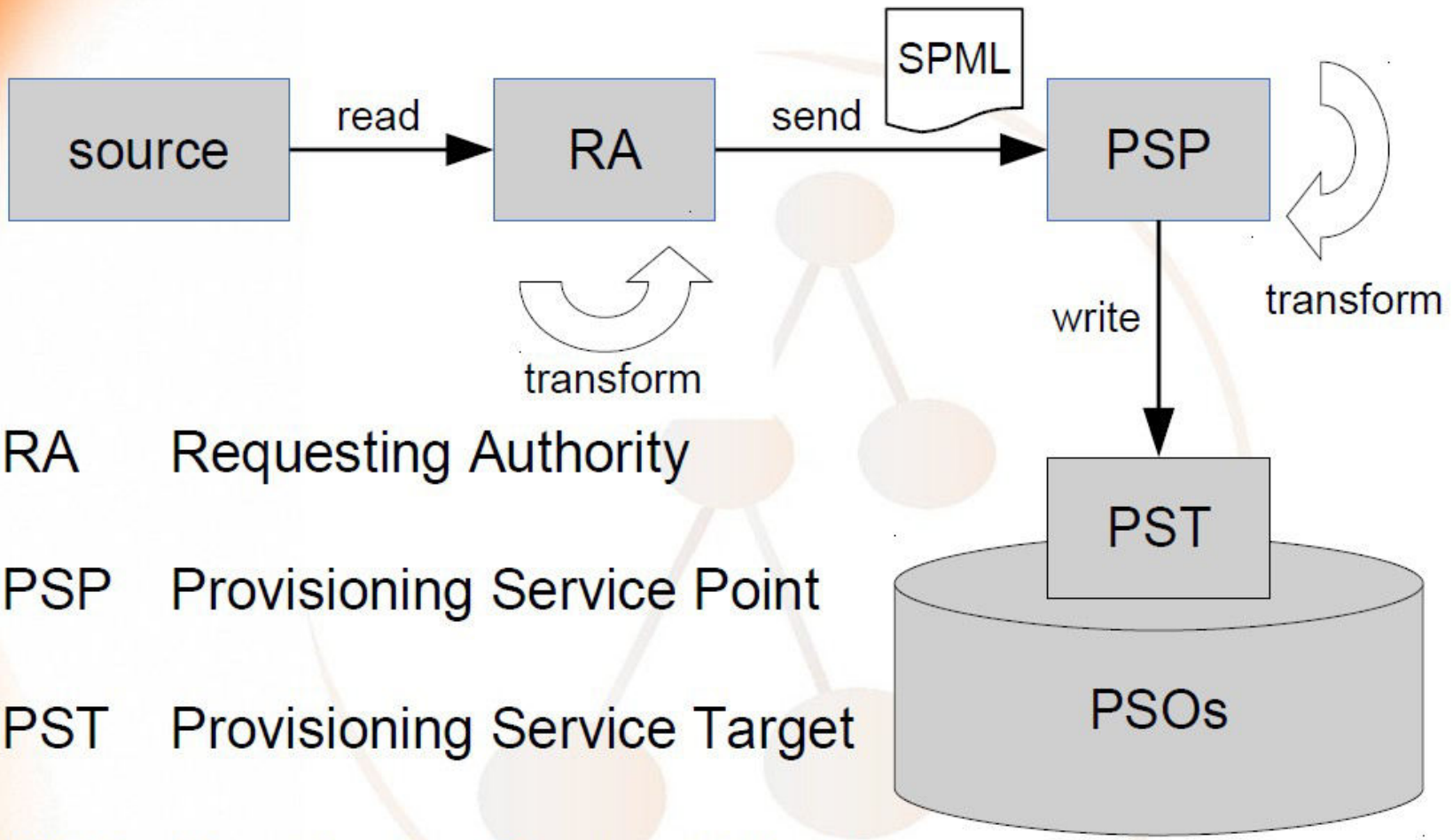
DAASI International GmbH

Solution 1: SPML

- Service Provisioning Markup Language
- As SAML an XML-based OASIS standard (v2 from 2006)
- Rich protocol that allows for provisioning and de-provisioning
- Supports different data schemas: standard DSML (OASIS again) or custom schemas
- Extensible standard with core and extensions
- Uptake was not too impressive
  - Implemented in a lot of provisioning systems
  - But not a lot at the target side
- I still think it is a well thought through protocol and it was successfully implemented within organizations

DAASI International GmbH

SPML



RA Requesting Authority

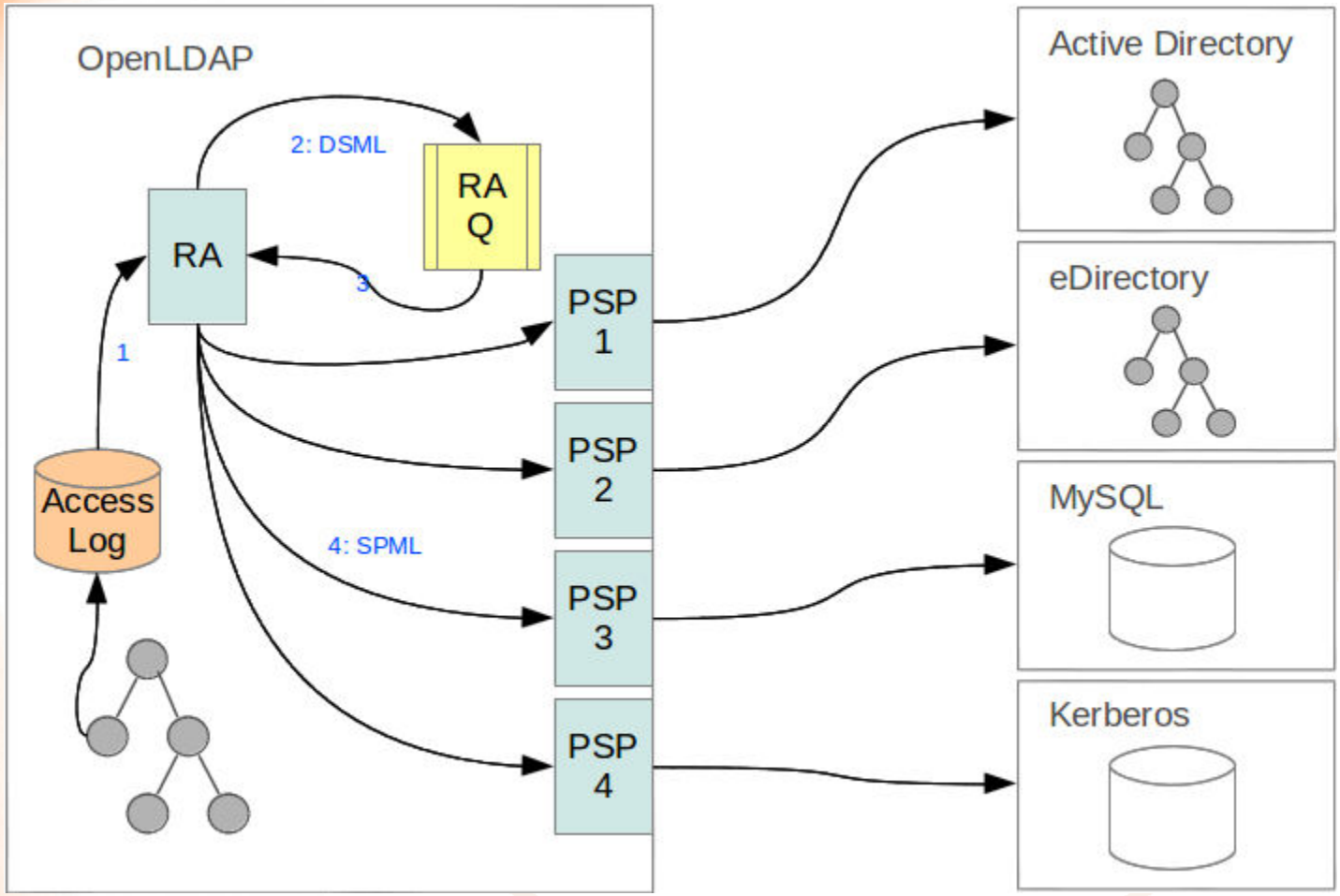
PSP Provisioning Service Point

PST Provisioning Service Target

PSO Provisioning Service Object

DAASI International

SPML-Provisionierung



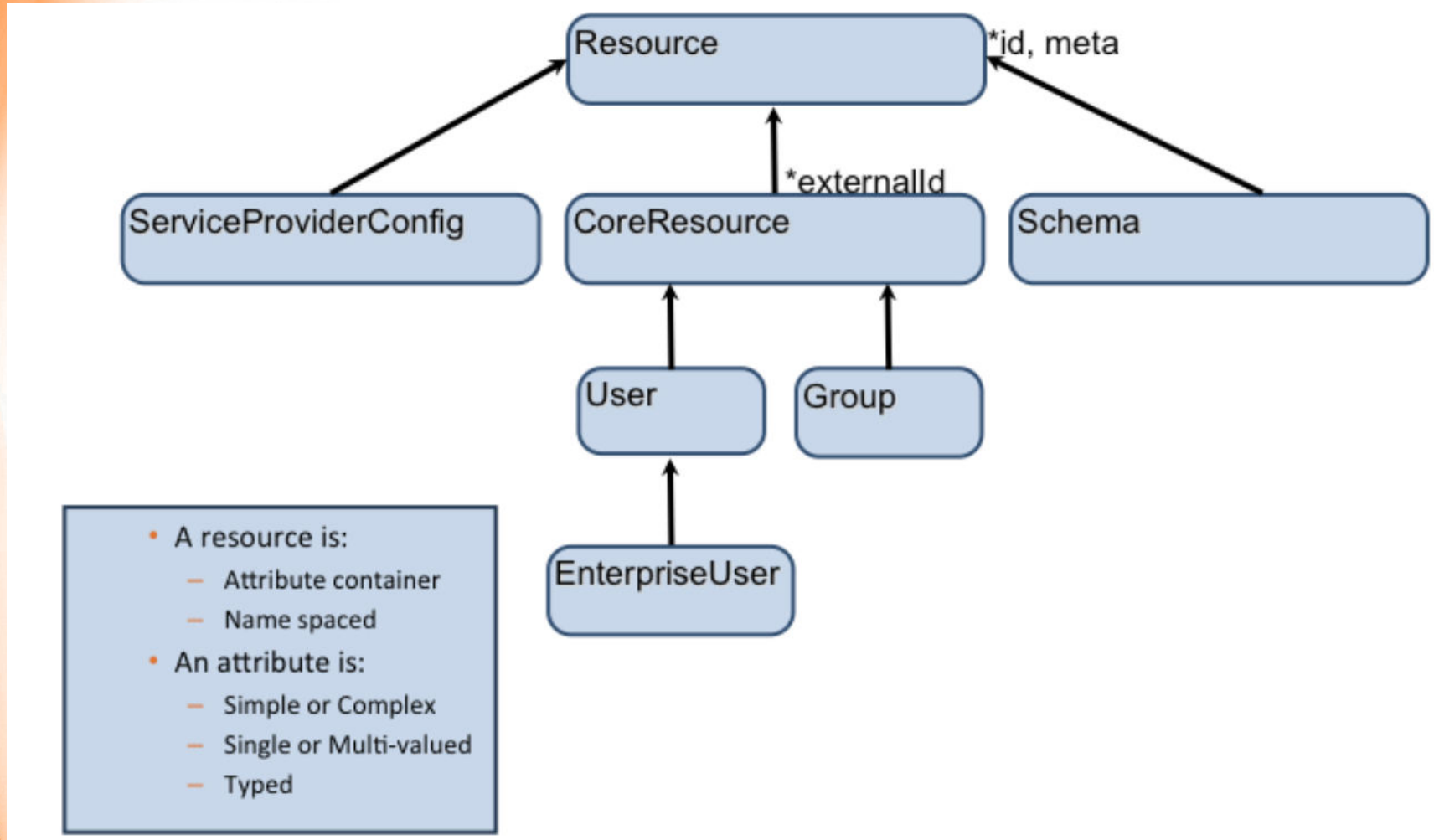
DAASI International GmbH

Solution 2: SCIM

- System for Cross-domain Identity Management
  - Originally it stood for Simple Cloud Identity Management
  - Sort of the “LDAP in the cloud“
- Version 1.0 (OWF Dezember 2011), Version 1.1 (OWF Juli 2012)
- Version 2.0 (IETF September 2015, RFC7642, RFC7643 and RFC7644)
- It is much simpler than SPML
  - Fixed schema for user and group
- Uses JSON instead of XML
- Synchronous REST HTTP for CRUD instead of SOAP
- OAuth or mutual X.509 for security
- Protocol Binding for SAML, LDAP

DAASI International GmbH

Solution 2: SCIM



Thanks for your attention!

- Any questions?

DAASI International GmbH

- SCIM: http://datatracker.ietf.org/wg/scim
- https://www.terena.org/mail-archives/tf-emc2/pdfat8UjFqw99.pdf