A solution for Access Delegation based on SAML

Ciro Formisano
Ermanno Travaglino
Isabel Matranga

Access Delegation in distributed environments

SAML 2.0 Condition to Delegate

Implementation

Future plans

European Grid Infrastructure

Access Delegation in distributed environments

SAML 2.0 Condition to Delegate

Implementation

Future plans

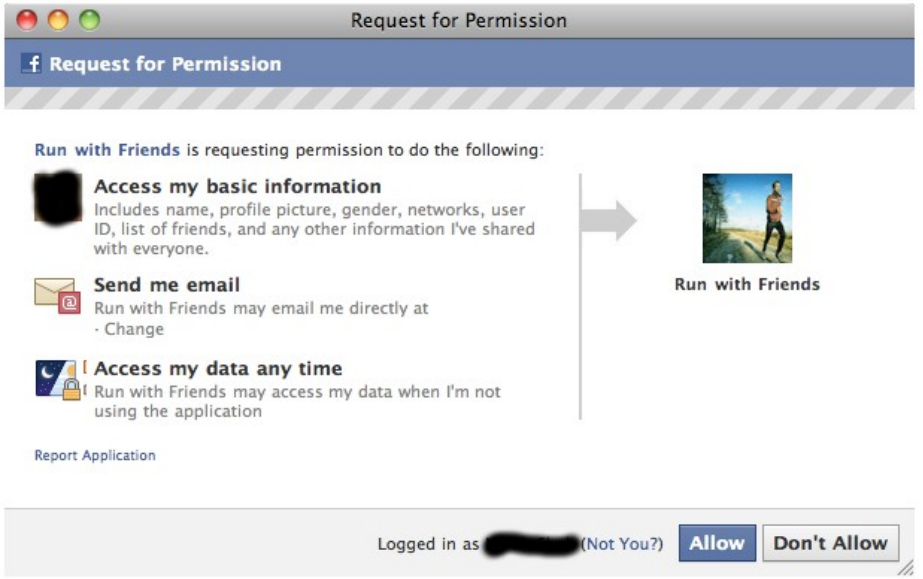
ENGINEERING

Access Delegation

- *Access Delegation* is the process by which an entity provides another entity with a subset of its privileges
  - Who provides the privileges is the *delegator*
  - Who is provided with the privileges is the *delegate*
  - Delegation is recursive: i.e. a delegate can provide another delegate with a subset of his/her/its privileges
- Delegator and delegate can be users or processes
  - In most cases the first delegator is a human user (but it is not mandatory)
- Delegator and delegate must be known by the involved identity domain(s)
- Delegation is time limited

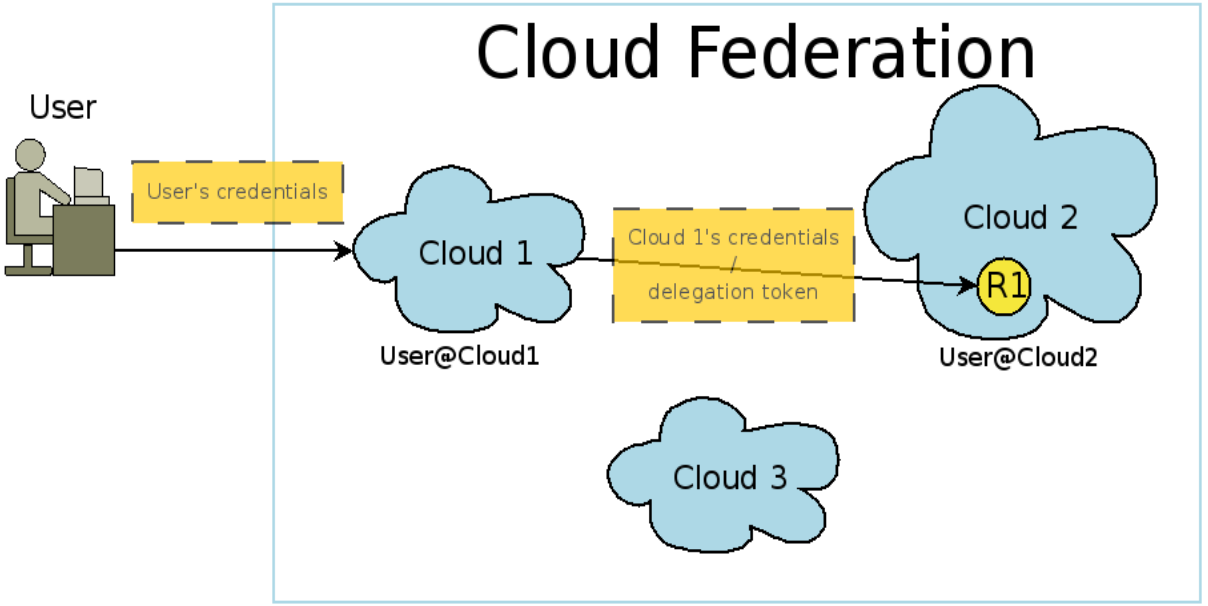
Use cases

- The delegate is a user
  - Someone provides someone with a power of attorney for some private affairs
  - The boss enables an employer to access certain data on his behalf
- The delegate is a process
  - A batch grid job started by a process authenticated by a proxy certificate
  - A Facebook user enables an application to access some data with his/her privileges



Delegation in a Cloud Federation

- The User has identities on the two Clouds (User@Cloud1 and User@Cloud2 who owns R1)
- Cloud 2 trusts Cloud 1 through its credentials (e.g. X509)
- User@Cloud2 delegates Cloud 1 to manage R1
- From this moment User@Cloud1 can safely manage his/her objects on Cloud 2 as they were on a single cloud
- Behind the scene Cloud 1, if needed, accesses Cloud 2 to manage data on behalf of User



Basic information to delegate

- The basic information needed for a delegation process, is:
  - The *delegation actors* (involved delegators, delegates, actions, roles and resources)
  - Delegation expiring time
- These pieces of information must be checked by the Identity and Access Management System:
  - The delegation is not expired
  - The caller must be be authenticated (through his/her/its credentials) and his/her identity must match with the identifier of the delegate
  - The delegator is associated with the delegated roles and actions on delegated resources and is able to delegate them

Access Delegation in distributed environments

SAML 2.0 Condition to Delegate

Implementation

Future plans

SAML 2.0 Condition to Delegate

- OASIS Specification fully compliant with SAML 2.0
  - defines *DelegationRestrictionType, a subtype* of *Condition* containing a sequence of *Delegate* elements defining the delegation chain
  - The *Delegate* element includes attributes defining the delegation instant and the confirmation method
  - The identifier of the delegate is included in a child element of *Delegate*
- The produced Token is a valid SAML 2.0 Assertion
  - Contains all the mandatory elements
  - The Condition element of *DelegationRestrictionType* is ignored by Service Providers not compliant with the specification

European Grid Infrastructure

Access Delegation in distributed environments

SAML 2.0 Condition to Delegate

Implementation

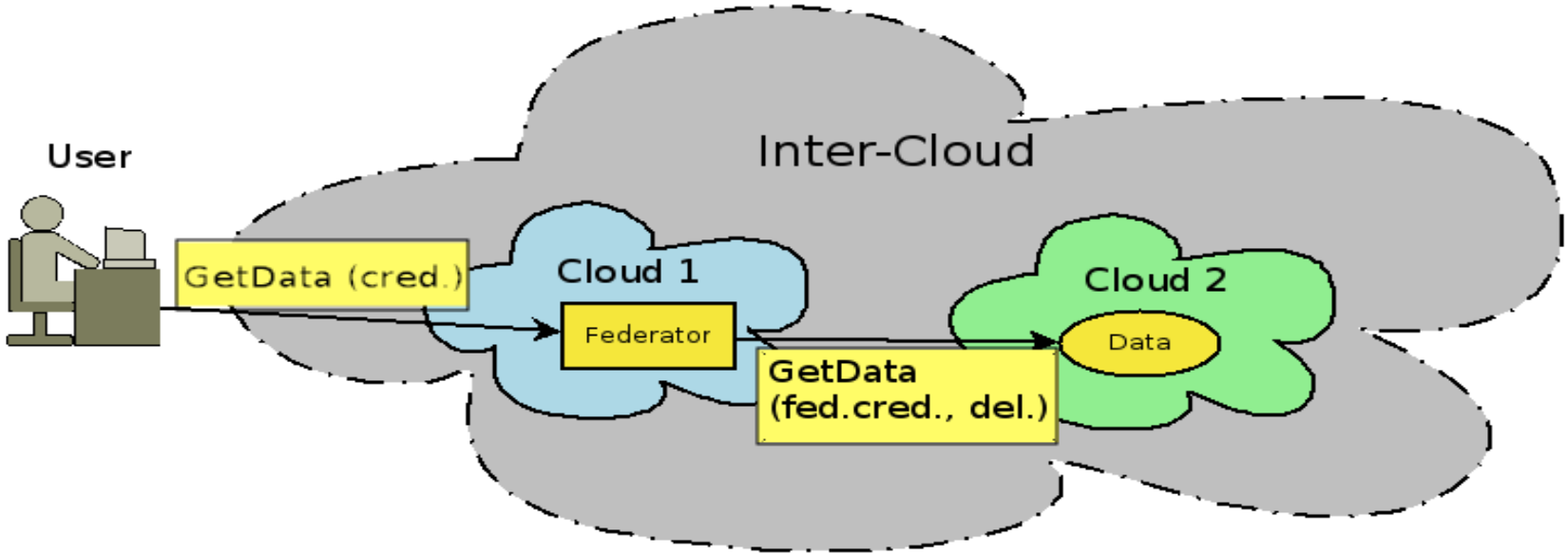
Future plans

ENGINEERING

Complete Information set for delegation

- SAML 2.0 Condition to Delegate specification defines two pieces of information:
  - Delegation chain
  - Delegation instant
- A basic delegation process need further pieces of information, specified in the rest of the Assertion, in particular:
  - The Assertion subject defines the last delegator (also present in the chain)
  - Assertion attributes define delegated actions, roles and resources
  - *NotOnOrAfter* and *NotBefore* elements define the Assertion lifetime

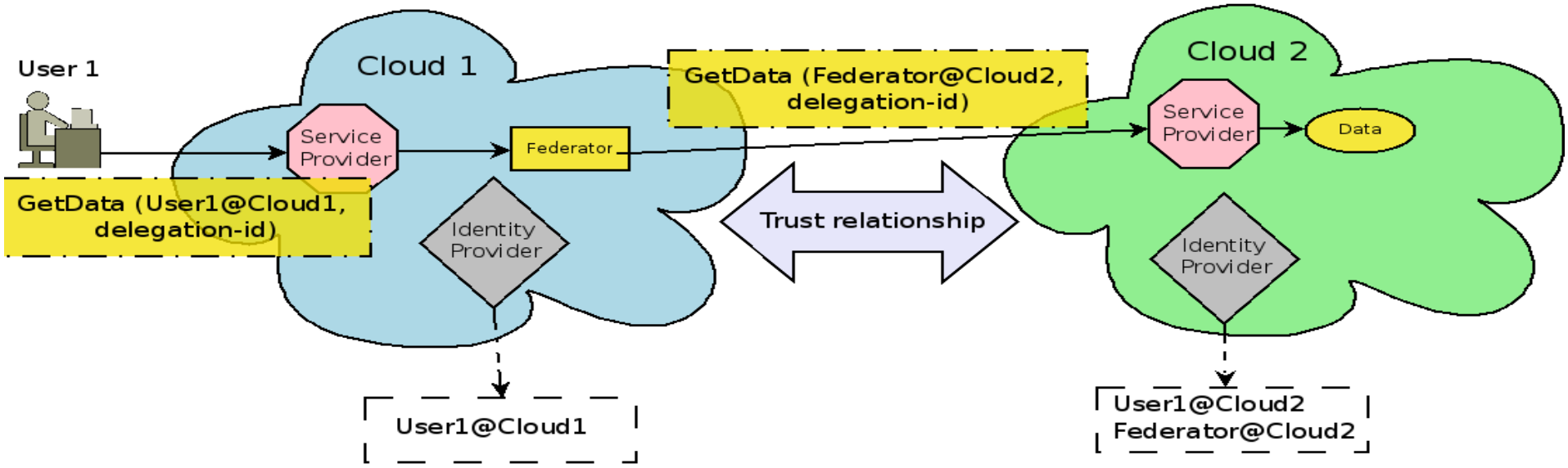
Use Case: data federation on a inter-cloud



Resources inside the federation should be accessed regardless their actual location

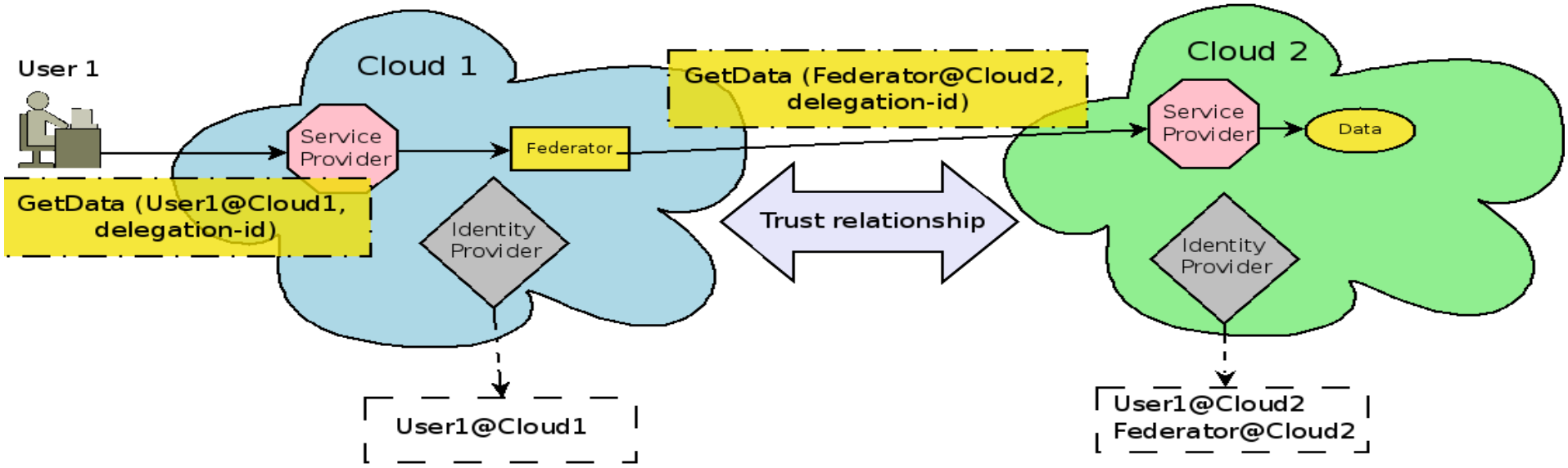
- A process called *Federator* dispatches every user request to the cloud where resources are
- The Federator Process is *delegated* by every user of the Cloud to access their data in the Federation

Resource Federation Process (1)



- The two clouds must trust each other
- User 1 owns an identity on Cloud 1 and an identity on Cloud 2
- The Federator owns an identity on Cloud 2
- User 1 provides the Federator with a delegation id to access his/her data on Cloud 2

Resource Federation Process (2)



- User 1 logs into Cloud 1 to request Data
- Cloud 1 discovers that requested data are on Cloud 2 and asks the Federator to retrieve them
- The Federator uses the delegation id to get the data with its identity

Developed components

- Shibboleth provides a plugin compliant with SAML 2.0 Condition to delegate
  - Shibboleth plugin is limited to uPortal use case
  - Shibboleth plugin does not enable direct Token management
- Implemented an extension to Shibboleth Service Provider and Identity Provider
  - Manages SAML Token Profile
  - Provides SAML 2.0 Condition to delegate at Token level

European Grid Infrastructure

Access Delegation in distributed environments

SAML 2.0 Condition to Delegate

Implementation

Future plans

ENGINEERING

Future plans

Access Delegation Mechanisms will be part of a Federated Identity and Access Management Framework

- SAML Identity Federation will be integrated with SAML Delegation
  - SAML Identity Federation Module provides Shibboleth based web browser SSO and web services federation
  - will enable delegated access with identities valid in the whole Federation (no user1@Cloud1 and user1@Cloud2)
- A complete XACML based Federated Authorization framework
  - Multi-layer authorization, enabling to define and enforce attribute based authorization policies at federation layer and domain layer

Kiitos!

Ciro Formisano

<ciro.formisano@eng.it>