
RIA-653549

# INDIGO – DataCloud

## Enabling Collaboration in an Identity-rich World

**Andrea Ceccanti (INFN)**
Paul Millar (DESY)
Bas Wegh (KIT)
Patrick Fuhrmann (DESY)

indigo-aai-tf@lists.indigo-datacloud.org

# INDIGO Datacloud

- An H2020 project approved in January 2015 in the EINFRA-1-2014 call
  - 11 M€
  - 30 Months (Apr. 2015 -> Sept. 2017)
- **Who:** 26 partners from 11 European countries
- **What:** develop an **open source** platform for computing and data targeted at **multi-disciplinary scientific communities**
- **Where:** provisioned over hybrid (public and private) e-infrastructures

INDIGO - DataCloud

# INDIGO objectives

- Provide **seamless access** to data and computing provisioned over private, public or hybrid e-infrastuctures
- Leverage and extend current Cloud technologies, **fill the gaps,** provide tools and services to support scientists, software developers, resource providers and e-infrastructures for the **efficient exploitation of computing, data and network technologies**:

**Better software for better science**

INDIGO - DataCloud

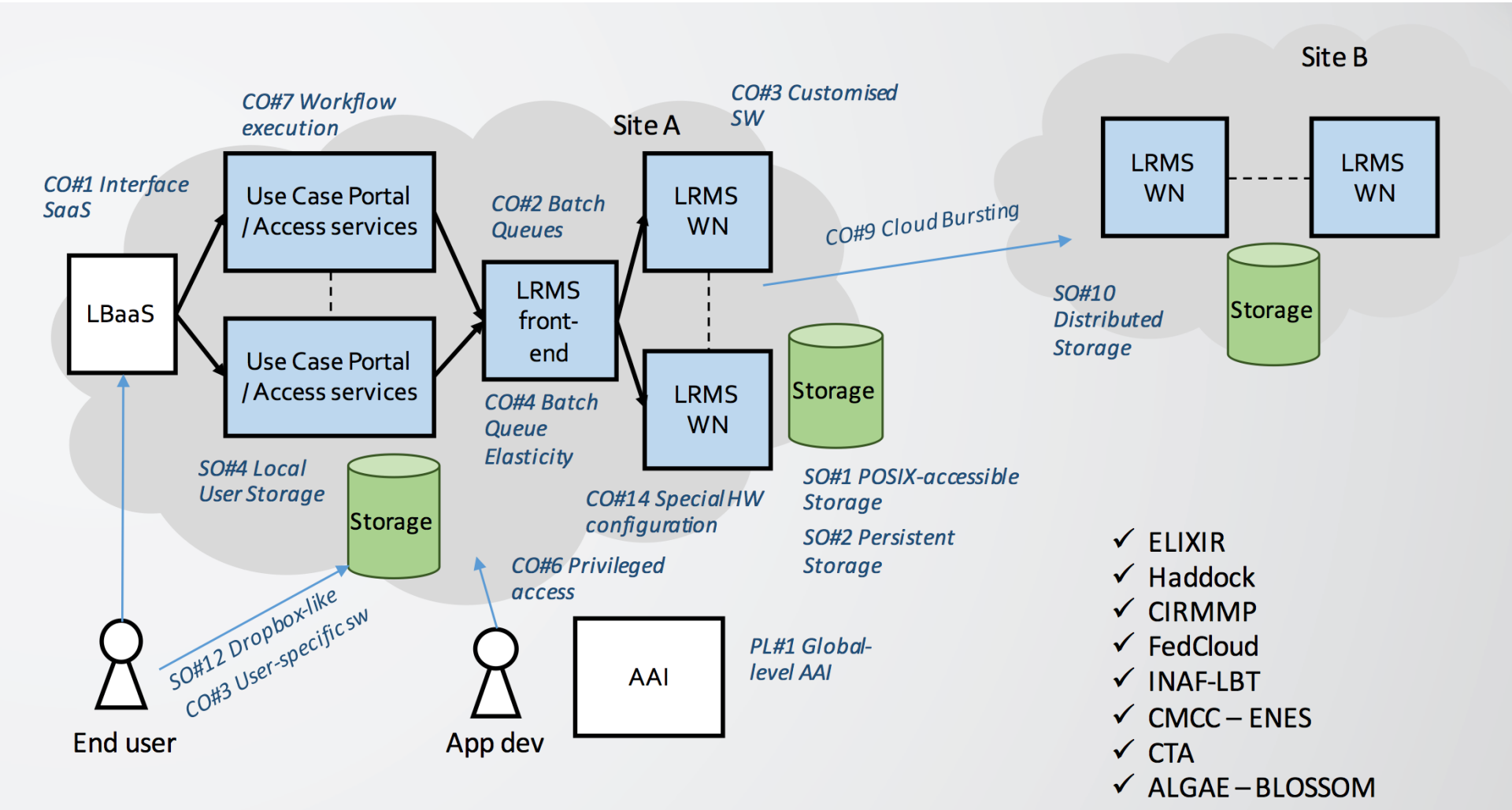
# The INDIGO approach

- Based on Open Source solutions
  - widely supported by big communities
- whenever possible exploit general solutions instead of specific tools/services
  - increased sustainability
- ensure that the framework offered to final users, as well as to developers, will have a **low learning curve**
  - ease of adoption and integration

INDIGO - DataCloud

# Example use case scenario

## Computational Portal "as a service"



INDIGO - DataCloud

# Example use case scenario

## Computational Portal "as a service"

- A scientific community has an application (or a set of them) that should be accessed through a portal, and:
  - Requires a dynamically instantiated batch queue as a back-end
  - Exhibits unpredictable workload
  - Supports multiple access profiles
  - Should be deployable through Cloud providers, with features such as redundancy and elasticity
  - May require cloud-bursting to other infrastructures
  - Should support access to external reference data and data local to the application, which must be accessible in a distributed way

INDIGO - DataCloud

# The AAI problem

- Heterogeneous authentication/authorization mechanisms
  - but we need common AAI ground, persistent identifiers, ease of integration in services
- To be effective, security should not impact usability
  - Federated identities support is lacking or very limited
- We need AAI solutions that allow our users to manage authentication and authorization on **dynamically** provisioned resources in a secure way
  - Without being security experts!

INDIGO - DataCloud

# INDIGO AAI: approach

- How can we have common AuthN and AuthZ primitives that “just work” across several distributed infrastructures?
- Which tools should we provide to our users so that they have complete control on how AuthN and AuthZ is configured and performed on the resources (assembled from distributed providers) they will use for their research?
- How do we avoid reinventing the wheel? How do we exploit what is already available, leverage existing standards and ensure that what we develop is sustainable?

# Authentication



**Slide courtesy of Paul Millar**

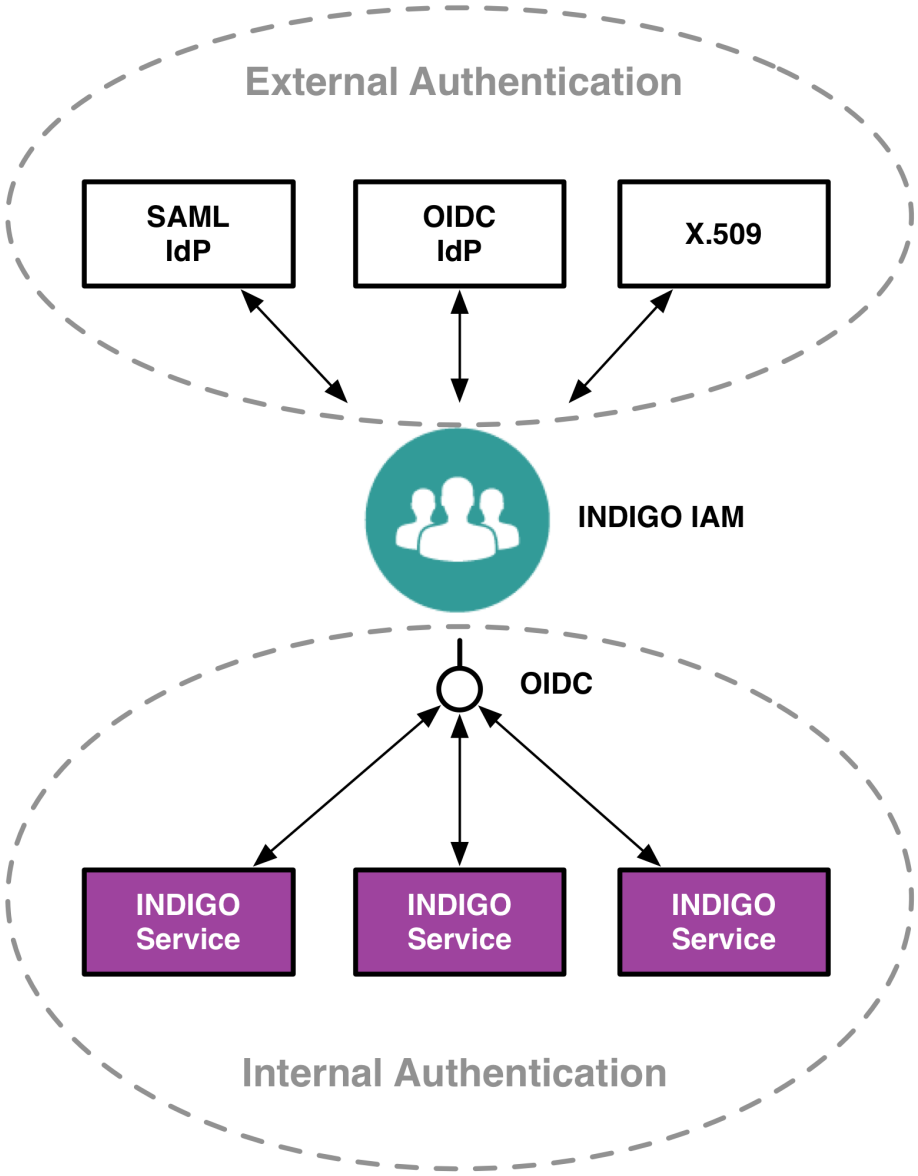
INDIGO - DataCloud

# Identity layer challenges

- Support multiple AuthN mechanisms
  - SAML, OpenID-Connect, X.509
- Harmonise Identities
  - One INDIGO identity linked to multiple user authN mechanisms
  - Persistent INDIGO identity identifier
- Link group membership and other attributes to INDIGO identity
  - similarly to what VOMS does with VO attributes and X.509 certificates, but in a way that is orthogonal to the AuthN mechanism used

# Identity in INDIGO

- The INDIGO identity layer speaks **OpenID-connect**
- The INDIGO **Identity and Access Management Service** is an OIDC provider
  - Authenticates users with supported AuthN mechanism
    - SAML, X.509, OIDC
  - Provides to RP access to identity information through standard OIDC interfaces
- Can be seen as a first credential translation step




# Why OpenID connect


- Standard and widely adopted in industry
  - Don’t reinvent the wheel
- Reduced integration complexity in relying services
- Lots of things we need are covered and standardized
  - Dynamic Registration of clients/relying parties
  - Token revocation
  - Discovery
  - Session management
  - Distributed/Aggregated claims
- Mobile-friendly

# INDIGO AuthN flow

**INDIGO Service**

Access service

Marcus wants to access some service at INDIGO service

**Home IdP**

**Indigo IAM**

# INDIGO AuthN flow

**INDIGO Service**

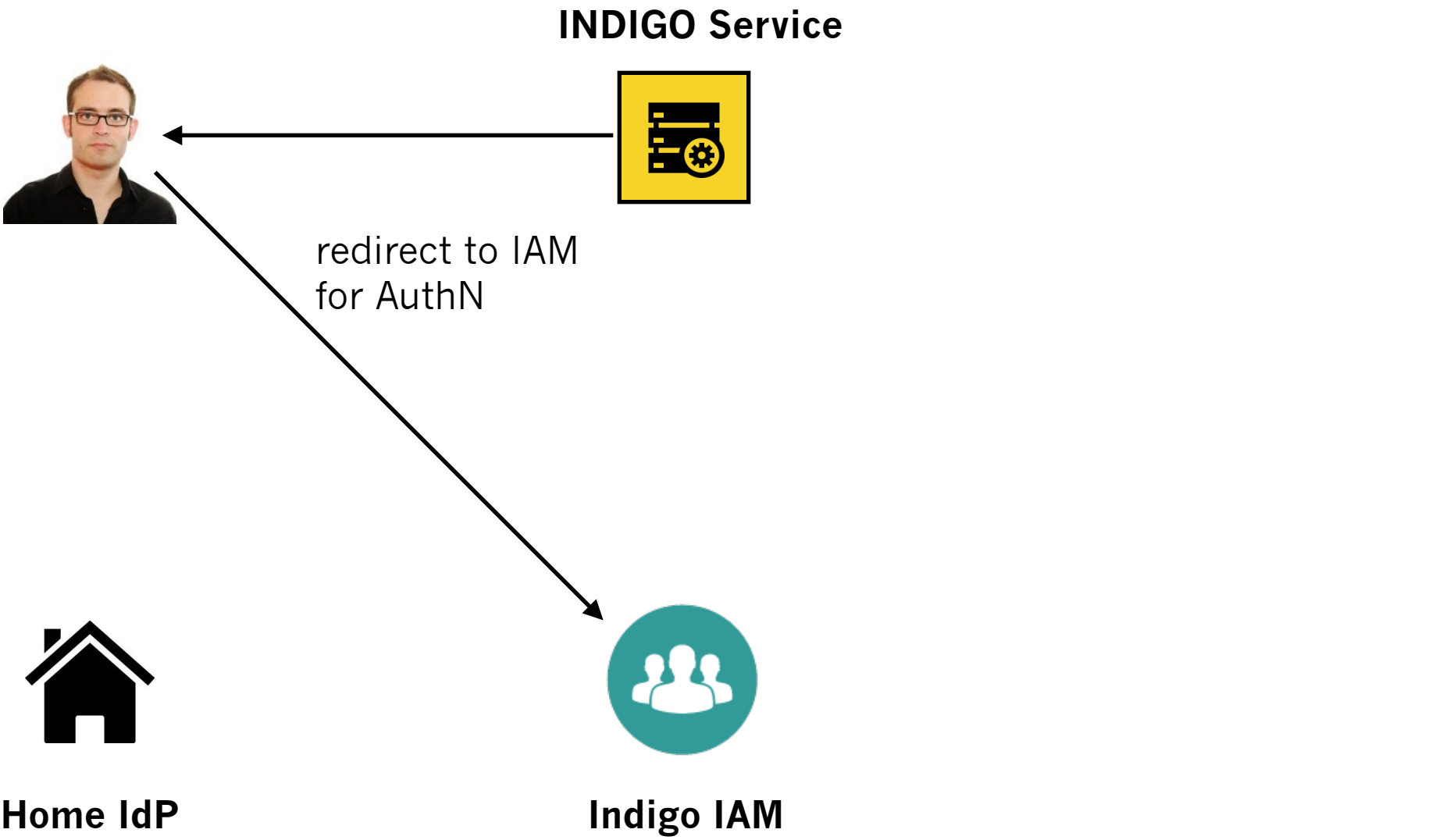
Access service

INDIGO Services sees that Marcus is not authenticated, and redirects him to INDIGO IAM for authentication

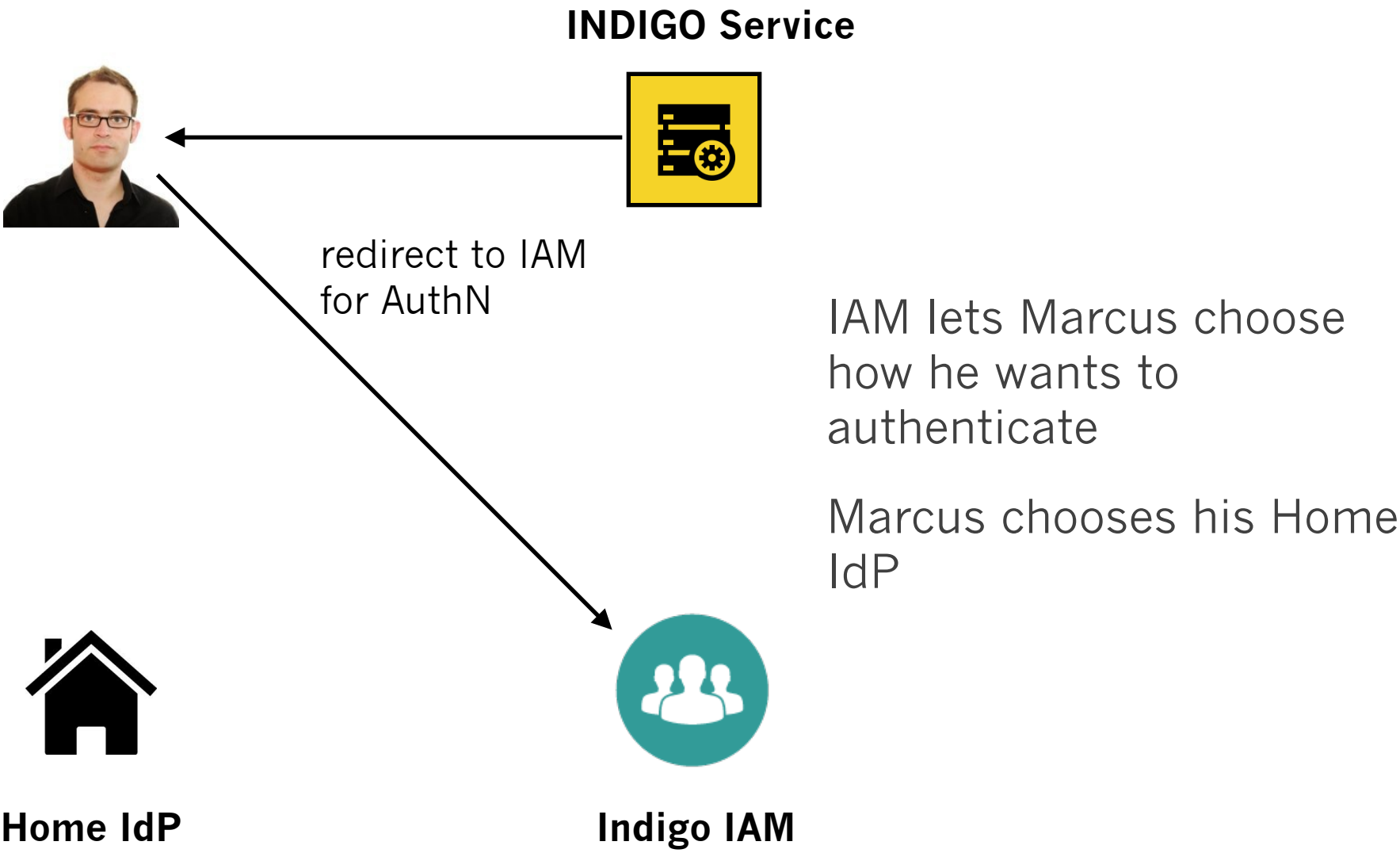
**Home IdP**

**Indigo IAM**

# INDIGO AuthN flow



# INDIGO AuthN flow



# INDIGO AuthN flow

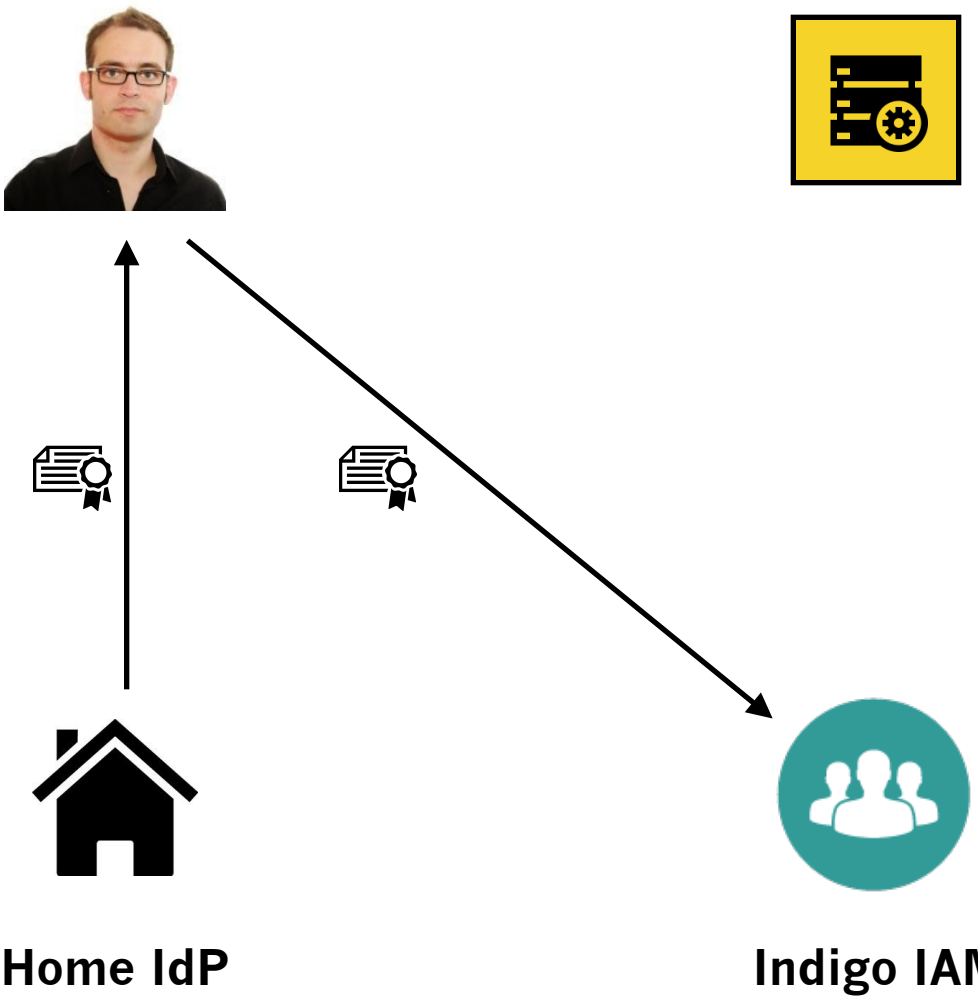
**INDIGO Service**

redirect to home
IdP for AuthN

**Home IdP**

**Indigo IAM**

# INDIGO AuthN flow




Home IdP authenticates Marcus and sends back an AuthN assertion

# INDIGO AuthN flow

**INDIGO Service**

IAM validates assertion.
Marcus is now
authenticated at IAM.

**Home IdP**

**Indigo IAM**

# INDIGO AuthN flow

**INDIGO Service**

send back to IS
OIDC authz code

**Home IdP**

**Indigo IAM**

# INDIGO AuthN flow

**INDIGO Service**

exchange
authZ code
for OIDC ID-token
access token

**Home IdP**

**Indigo IAM**

# INDIGO AuthN flow

INDIGO - DataCloud

**INDIGO Service**

IS validates ID-Token.
Marcus is now authenticated at IS

**Home IdP**

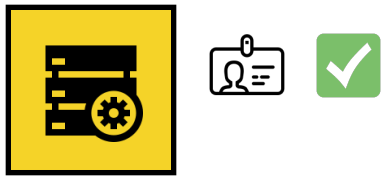
**Indigo IAM**

# INDIGO AuthN flow

INDIGO - DataCloud

**Marcus**

**INDIGO Service**



IS requests additional profile information about Marcus from IAM user info endpoint

**Home IdP**

**Indigo IAM**

INDIGO - DataCloud

# Authorization


**Slide courtesy of Paul Millar**

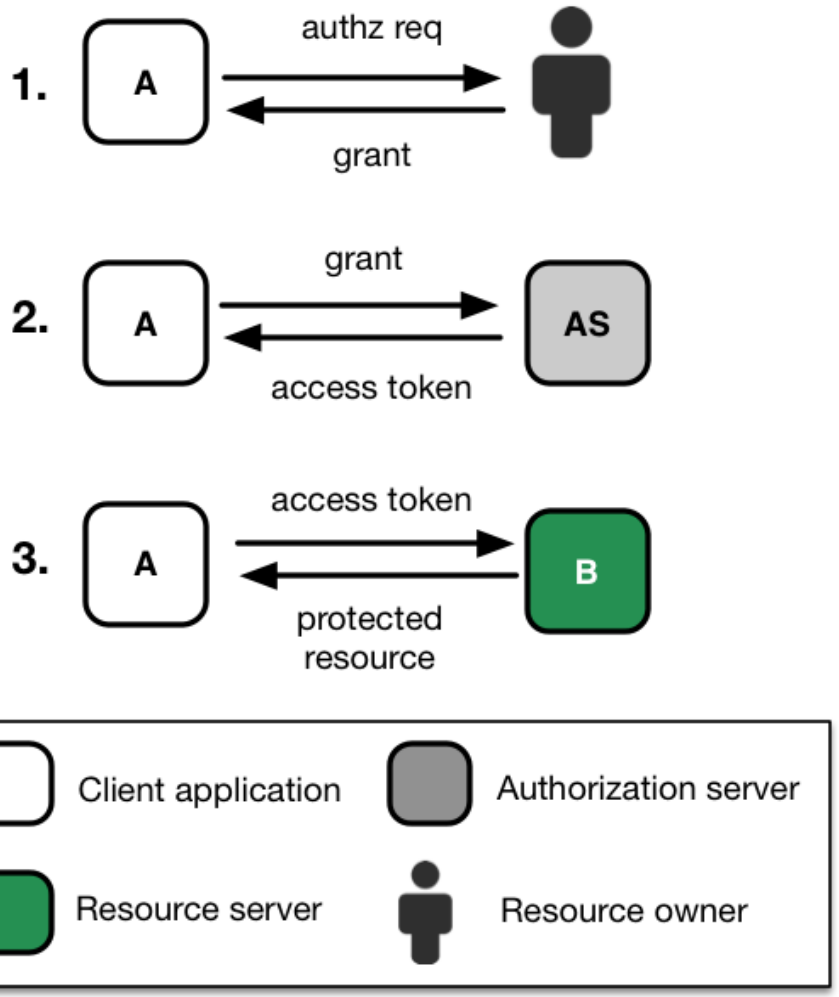
INDIGO - DataCloud

# Authorization challenges

- Support controlled delegation of privileges by design
- Provide tools to support cross-organizational user and privilege management
  - Enrollment flows and group management
  - User information provisioning
- Provide tools to **dynamically** define, propagate, compose and enforce authorization policies at various levels of the INDIGO stack based on identity attributes
  - Uniform and consistent authZ over resources assembled from multiple, heterogeneous providers

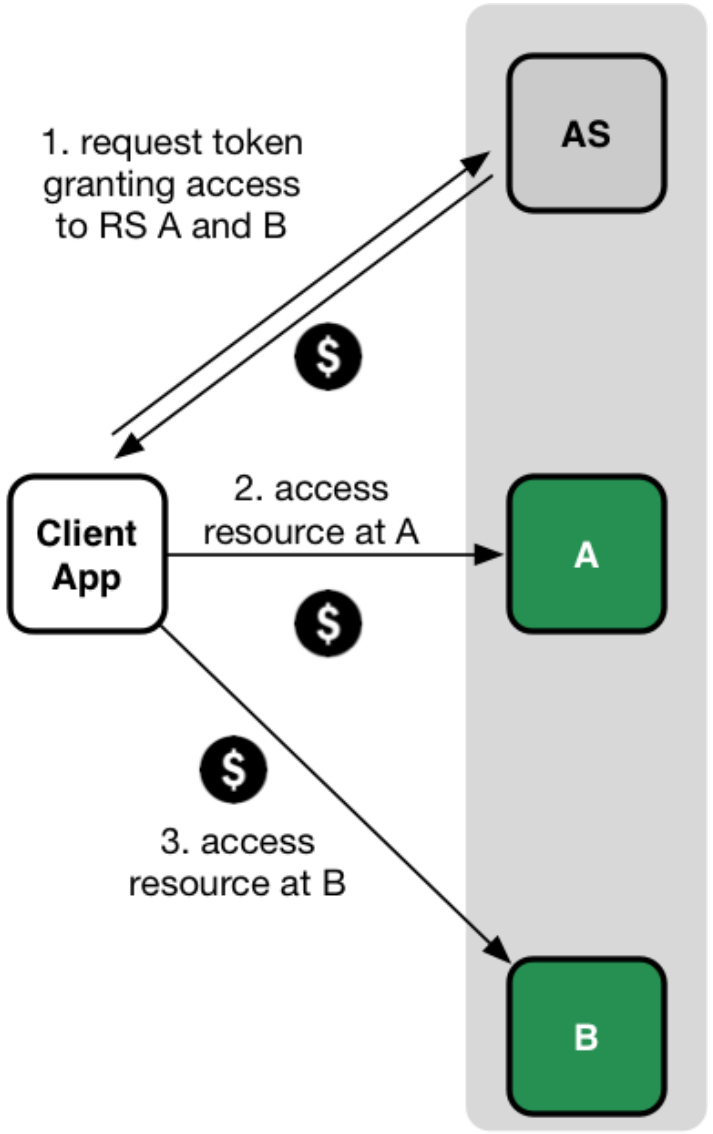
# AuthZ in INDIGO: OAuth 2

- A standard framework for delegated authorization to access HTTP protected resources
- Decouples AuthZ from AuthN
- Natural solution for delegated authorization in HTTP services



# Authorization in INDIGO

- INDIGO services are HTTP APIs protected by an OAuth Authorization Service
- In order to access resources, a client needs an access token
- OAuth scopes used to
  - target the token to specific APIs/ services
  - provide hints for finer grained authZ
- Identity layer provides other attributes as base for AuthZ decisions



INDIGO - DataCloud

# Scope-based authorization

- Each service registers the supported scopes when it registers at the authorization server (AS)
- The AS maintains policies that determine which client is authorized to request a given scope
- The request for a given scope is authorized by the user through the OAuth consent mechanism
  - but is possible to define trusted, whitelisted client services for which user consent is not requested
- Authorization is enforced at the target service considering scopes and other relevant information

# Scope-based authorization

Access SG

**SG**

**Science Gateway**

**Indigo IAM**

**Registered Scopes**

`js:submit_job`
`js:cancel_job`

`ss:read`
`ss:write`

**Job Scheduler**

**Storage Service**

# Scope-based authorization

Indigo IAM

**Registered Scopes**

`js:submit_job`
`js:cancel_job`

`ss:read`
`ss:write`

Redirect
for AuthN
& consent

**SG**

**Science Gateway**

**Job Scheduler**

**Storage Service**

# Scope-based authorization

Redirect
for AuthN
& consen

**SG**

**Science Gateway**

**Indigo IAM**

**Registered Scopes**

```
js:submit_job
js:cancel_job

ss:read
ss:write
```

**Authorization requested**

**Scientific Gateway would like to:**

**know your identity**
**submit jobs on JS on your behalf**
**read files from SS on your behalf**
**write files on SS on your behalf**

**Cancel** **Accept**

**Storage Service**

# Scope-based authorization

INDIGO - DataCloud

**Indigo IAM**

**Registered Scopes**

`js:submit_job`
`js:cancel__job`

`ss:read`
`ss:write`

Returned
id token &
access token

**Job Scheduler**

**SG**

**Science Gateway**

**Storage Service**

# Scope-based authorization

**Indigo IAM**

**Registered Scopes**

`js:submit_job`
`js:cancel_job`

`ss:read`
`ss:write`

Submit job

$

**Job Scheduler**

**SG**

**Science Gateway**

**Storage Service**

# Scope-based authorization

**Indigo IAM**

**Registered Scopes**

`js:submit_job`
`js:cancel_job`

`ss:read`
`ss:write`

Submit job

$

**Job Scheduler**

**Job scheduling is authorized by the js:submit_job scope**

**SG**

**Science Gateway**

**Storage Service**

# Scope-based authorization

**Indigo IAM**

**Registered Scopes**

`js:submit_job`
`js:cancel_job`

`ss:read`
`ss:write`

**Job Scheduler**

Read job output data

**SG**

**Science Gateway**

**Storage Service**

# Scope-based authorization

**Indigo IAM**

**Registered Scopes**

`js:submit_job`
`js:cancel_job`

`ss:read`
`ss:write`

**Job Scheduler**

**Data access is authorized by the ss:read scope**

Read job output data

$

**SG**

**Science Gateway**

**Storage Service**

INDIGO - DataCloud

# Fine-grained authorization

- OAuth scope-based authz provides a first coarse grained authorization step
- Finer-grained authorization can be implemented at services on top of this step taking into account
  - User identity attributes
  - Service authorization policies
  - Collaboration/VO policies
- Consistent authorization across services is enabled by callouts to the Argus authorization service

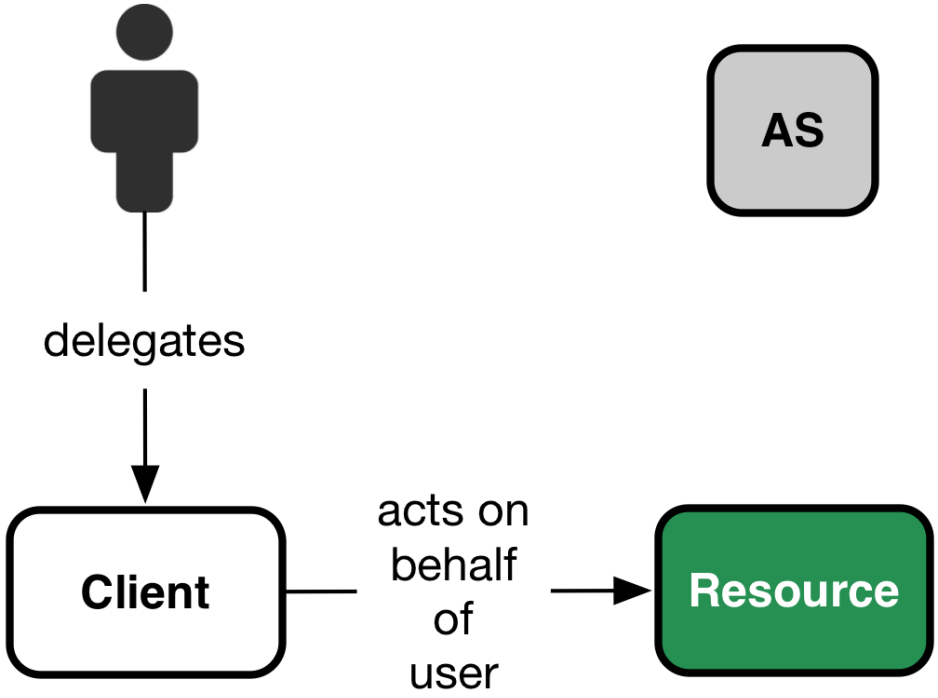
# Fine-grained authorization

INDIGO - DataCloud

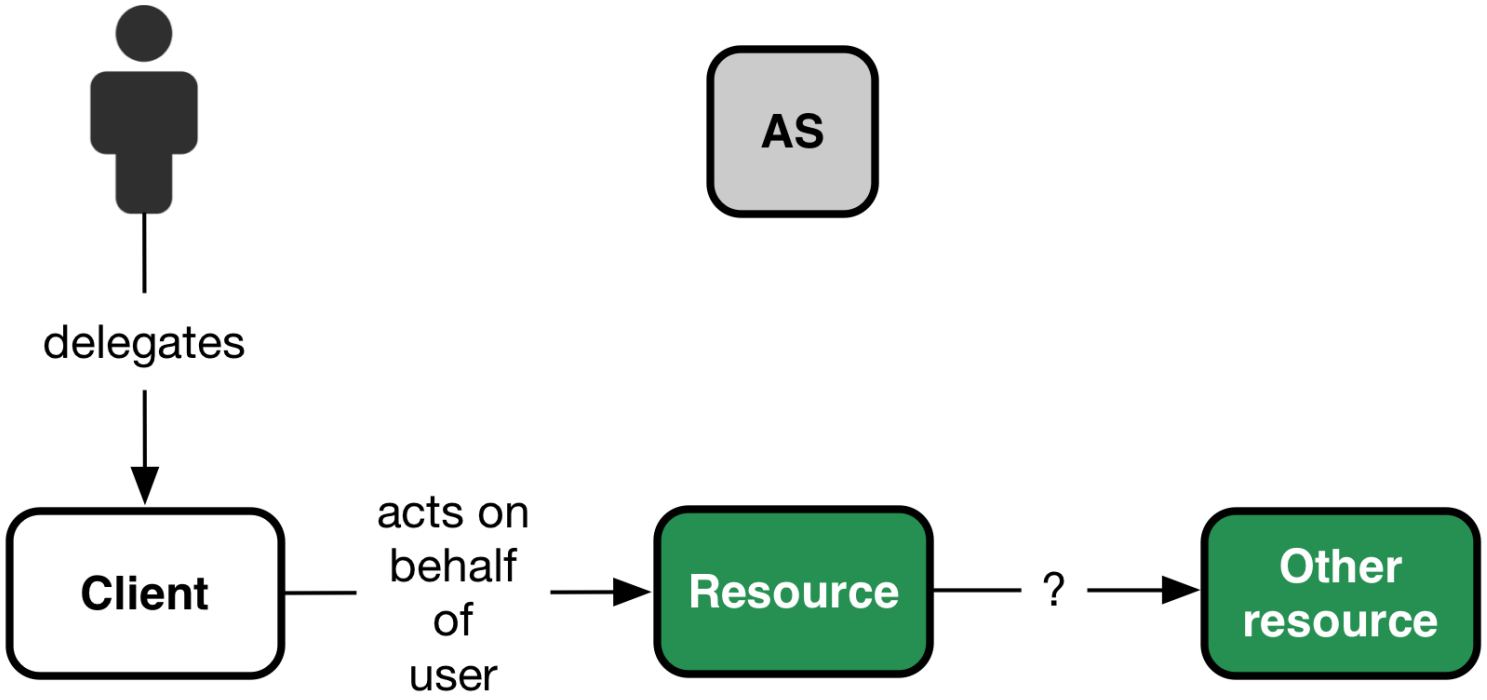
# Delegation

INDIGO - DataCloud

# OAuth delegation

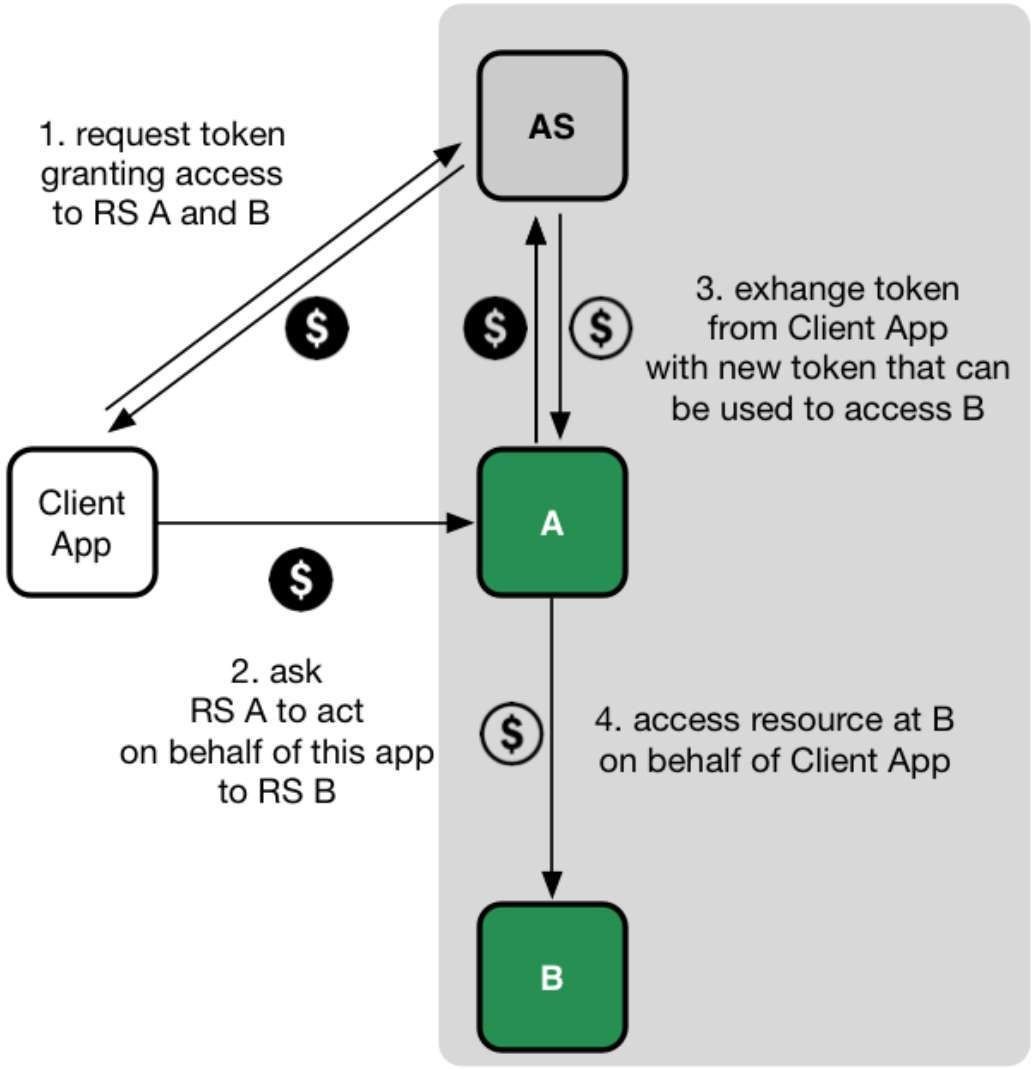


# OAuth chained delegation?



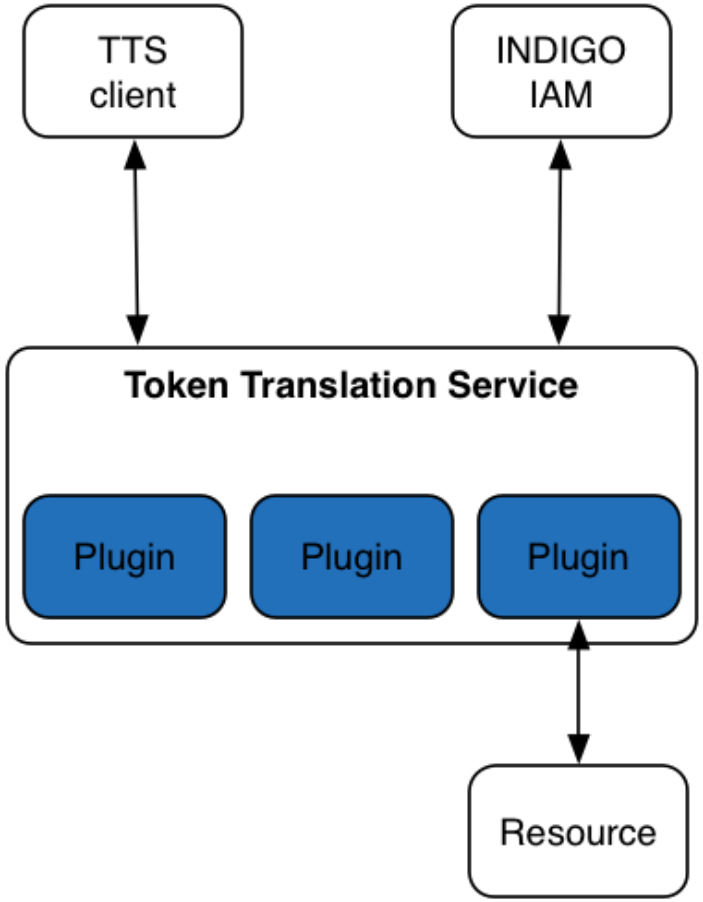
# OAuth token exchange

- OAuth flow to implement chained delegation among services
- Under standardization
- Supports impersonation and delegation

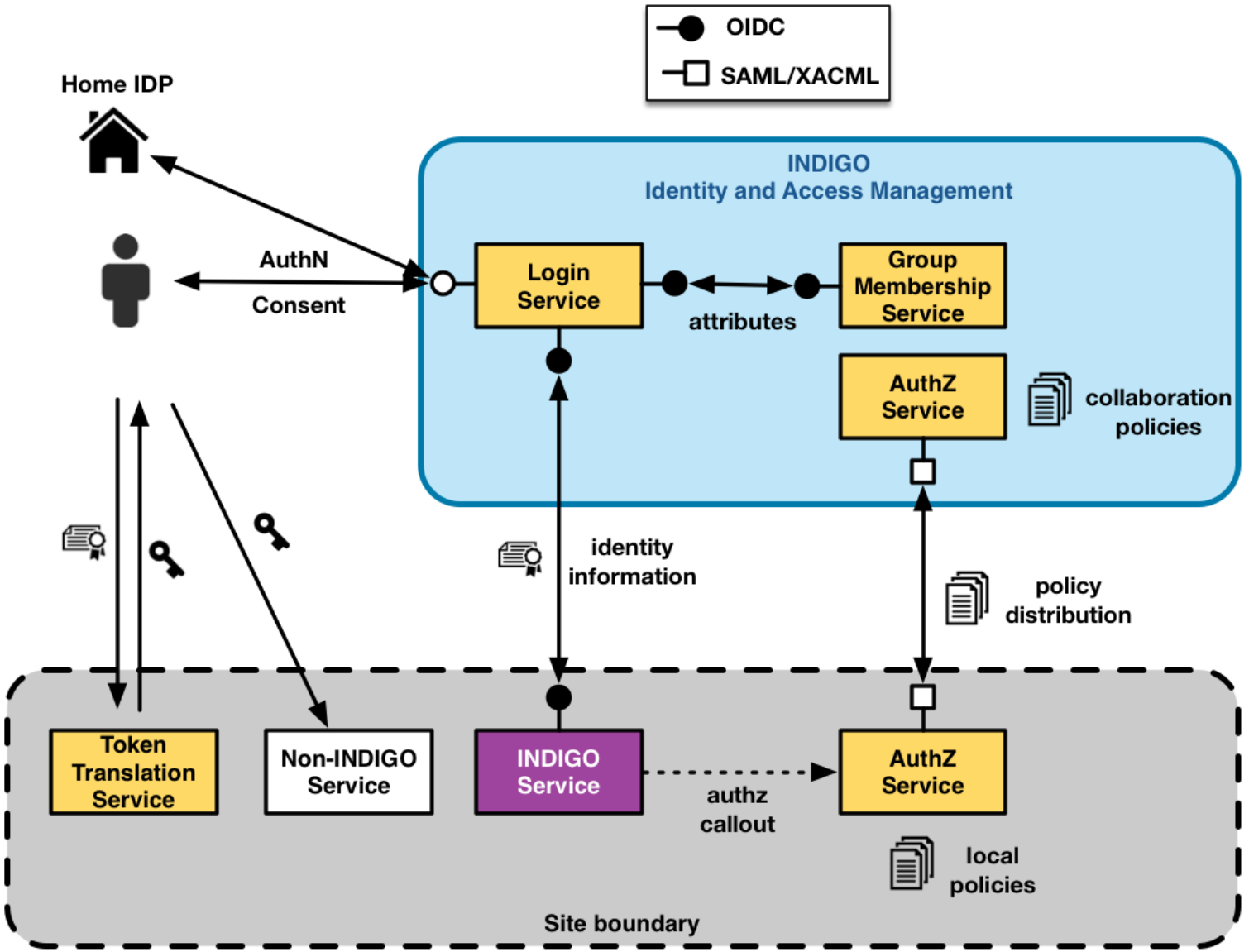


# Integration issues

- What about integration with services that do not speak OpenID-connect?
- The INDIGO **Token Translation Service** (TTS)
  - maps INDIGO identity & attributes to external service credentials
  - provides an extensible plugin-based architecture, and will initially support translations to
    - ssh keypair
    - S3 keys
    - X.509 certificate



# INDIGO AAI architecture



INDIGO - DataCloud

# Timeline

- Architectural and design deliverables done
  - See https://www.indigo-datacloud.eu/documents-deliverables for all INDIGO deliverables
- Development activities started
- First official INDIGO release expected end of July, 2016
  - but we will start make available services as soon as they are ready enough to be tested

INDIGO - DataCloud

# Thanks!

# Questions?

indigo-aai-tf@lists.indigo-datacloud.eu