INDIGO - DataCloud

RIA-653549

INDIGO – DataCloud

An introduction to INDIGO AAI

Andrea Ceccanti (INFN)

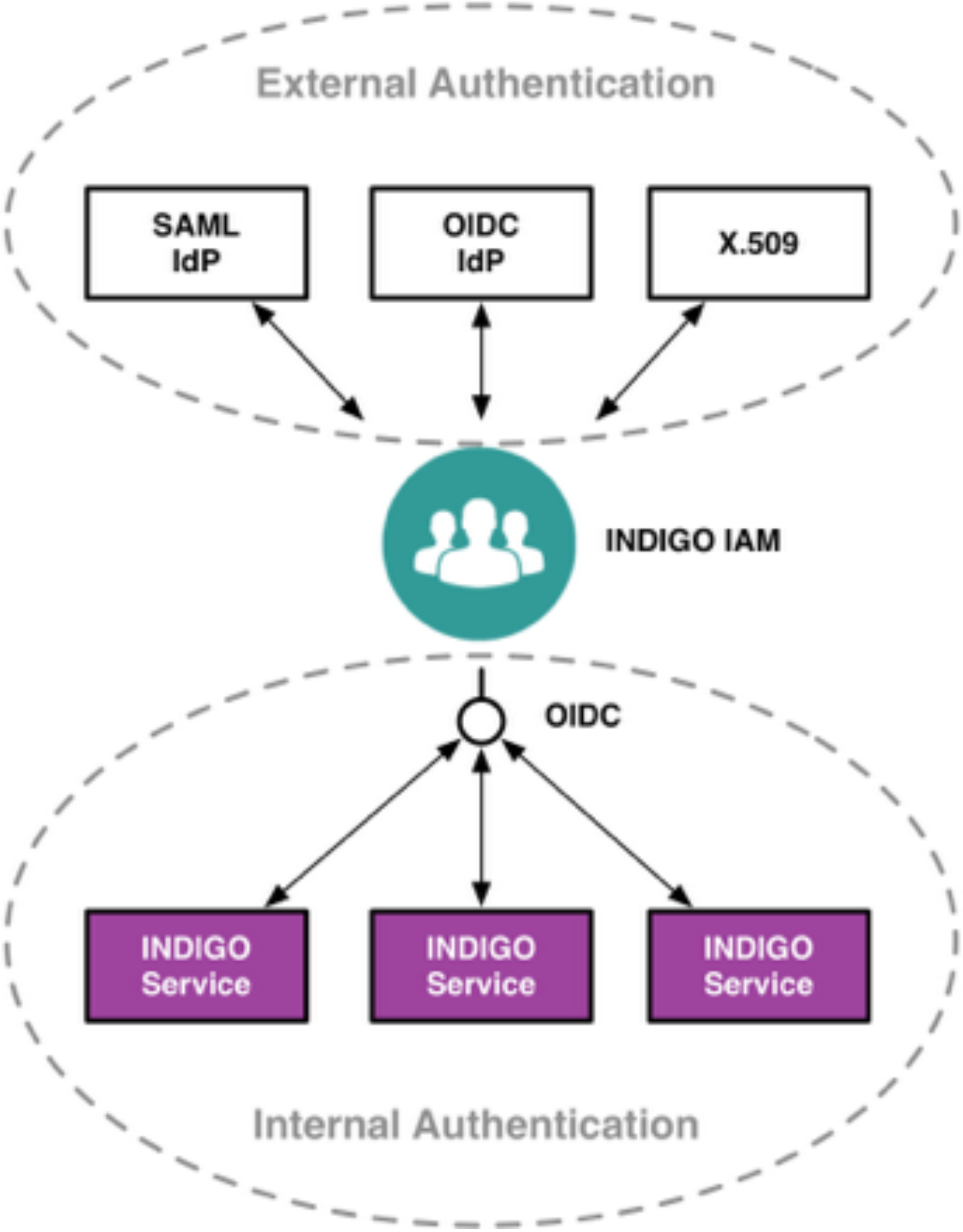
andrea.ceccanti@cnaf.infn.it

INDIGO AAI: main challenges

- Authentication
  - Support for **federated AuthN & social logins**
- Identity Harmonisation
  - Link multiple accounts to a single INDIGO identity, providing a **persistent identifier** orthogonal to AuthN mechanism
- Authorization
  - **Orthogonal to AuthN**, attribute-based, dynamic
  - Consistent across heterogeneous infrastructures
- Delegation
  - Provide the **ability for services to act on behalf of a user**
  - Support **offline access for long-running applications**
- Provisioning
  - provision/de-provision identities to services/relying resources
- Token translation
  - **enable integration with services relying on heterogeneous AuthN mechanisms**

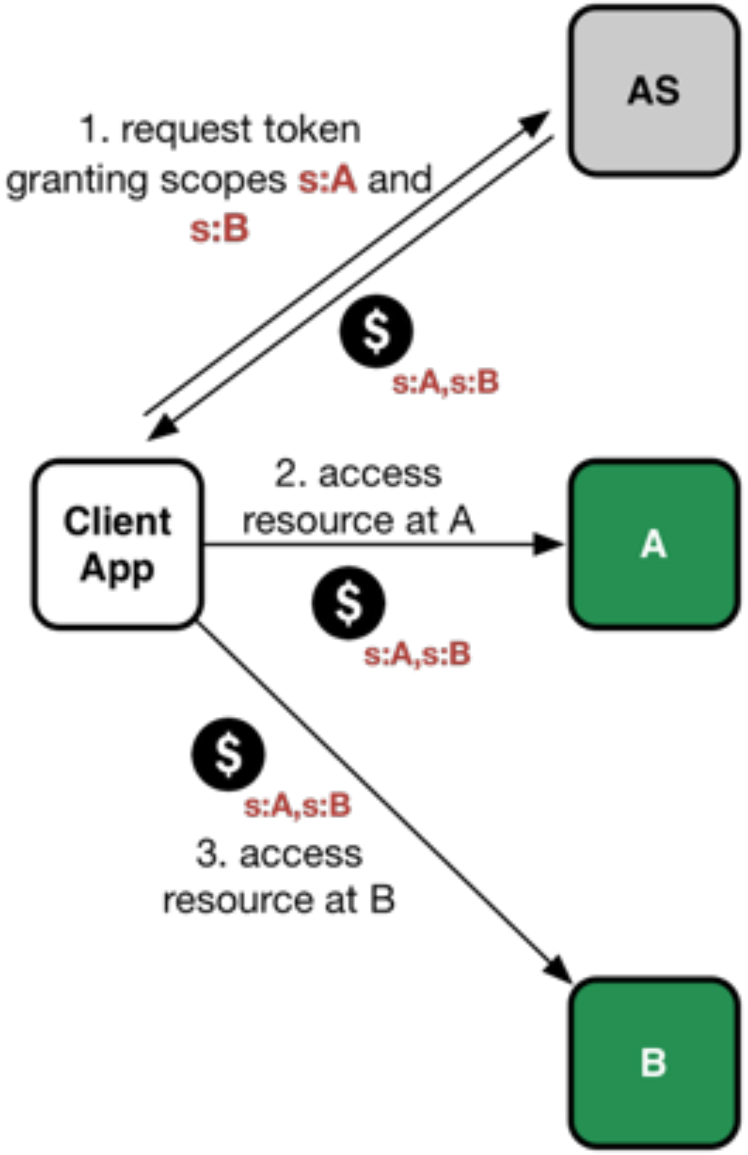
Identity in INDIGO

- 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 persistent identifier and links other attributes (e.g., group membership) to the INDIGO identity
- Provides to RP access to identity information through standard OIDC interfaces



OAuth2 AuthZ in INDIGO

- INDIGO services are HTTP APIs protected by an **OAuth** Authorization Service (AS)
- 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
  - e.g., group membership attributes

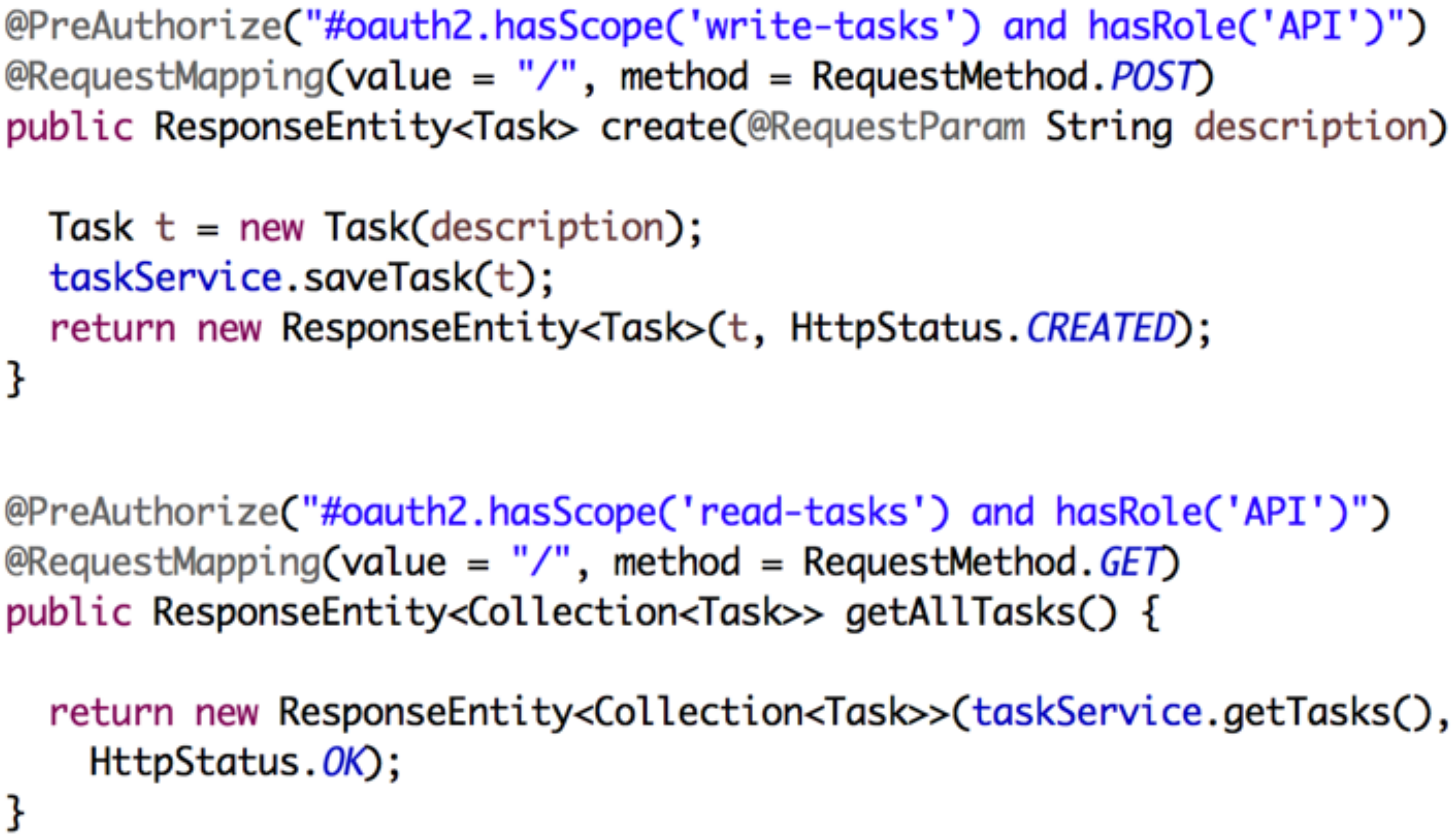


Scope-based authorization

- 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

INDIGO - DataCloud

Scope-based authz: example



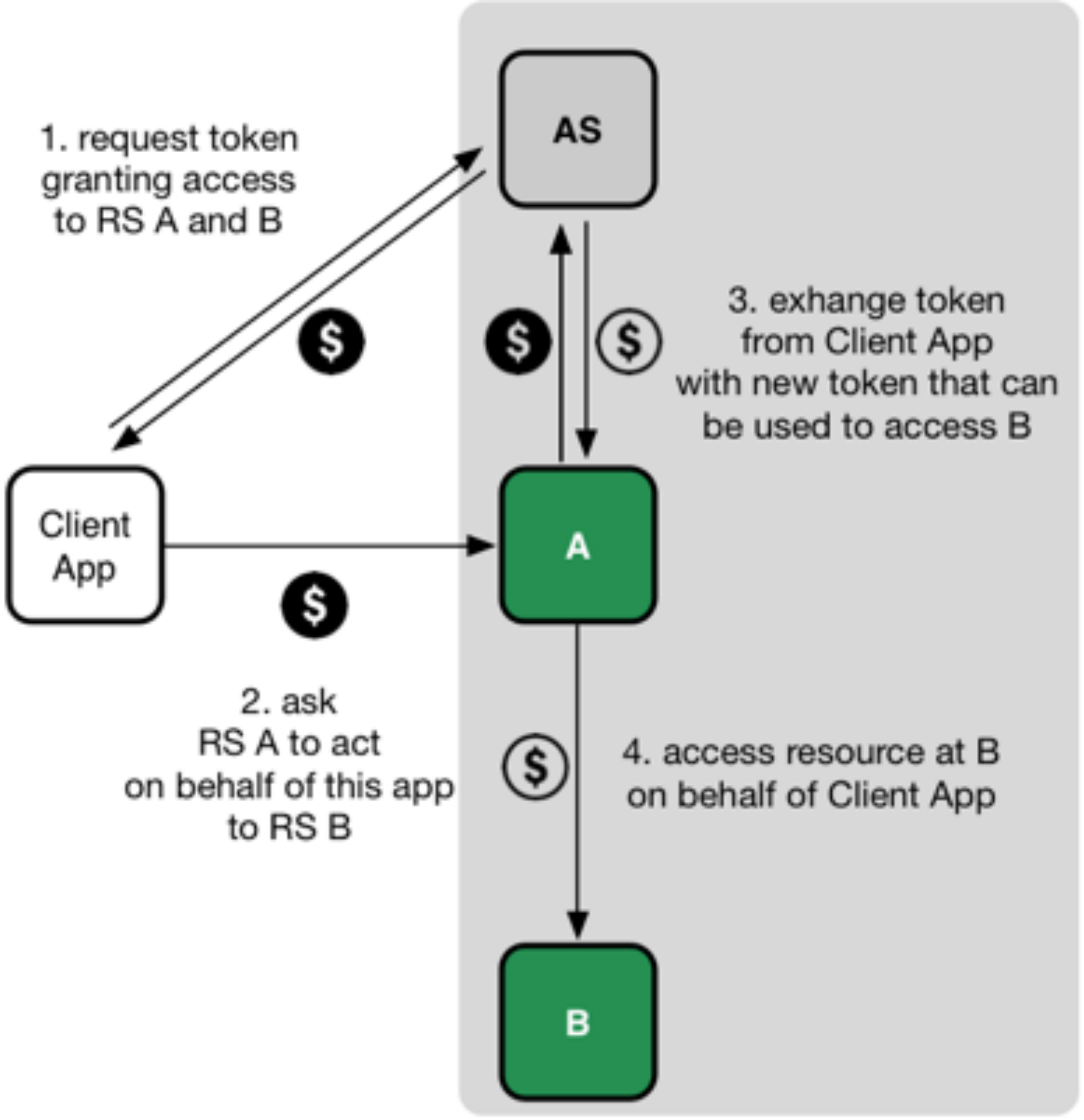
```
@PreAuthorize("#oauth2.hasScope('write-tasks') and hasRole('API')")
@RequestMapping(value = "/", method = RequestMethod.POST)
public ResponseEntity<Task> create(@RequestParam String description)

  Task t = new Task(description);
  taskService.saveTask(t);
  return new ResponseEntity<Task>(t, HttpStatus.CREATED);
}


@PreAuthorize("#oauth2.hasScope('read-tasks') and hasRole('API')")
@RequestMapping(value = "/", method = RequestMethod.GET)
public ResponseEntity<Collection<Task>> getAllTasks() {

  return new ResponseEntity<Collection<Task>>(taskService.getTasks(),
    HttpStatus.OK);
}
```

OAuth token exchange

- OAuth flow to implement chained delegation among services

- Under standardization

- Supports impersonation and delegation


AS
1. request token granting access to RS A and B
$
$
$
3. exhange token from Client App with new token that can be used to access B
Client App
A
$
2. ask RS A to act on behalf of this app to RS B
$
4. access resource at B on behalf of Client App
B

INDIGO - DataCloud

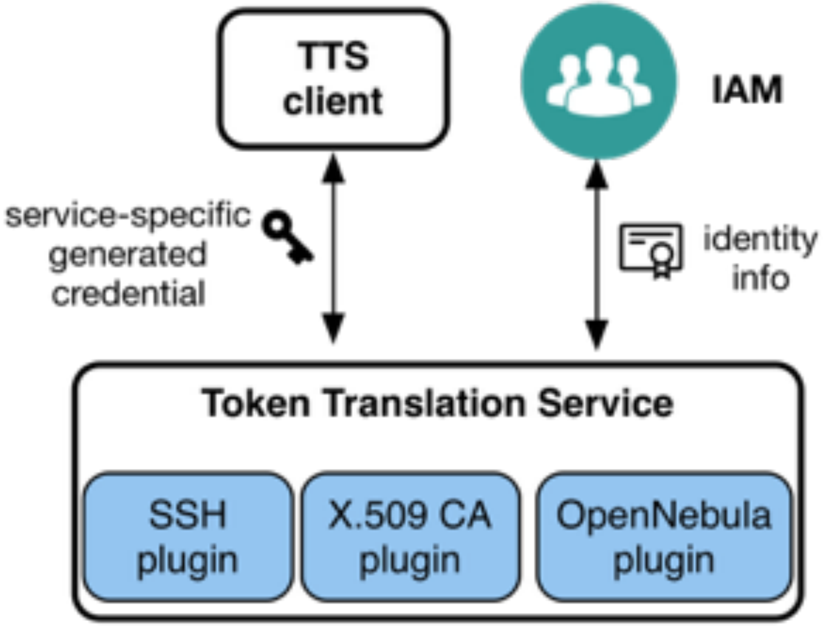
Provisioning

SCIM

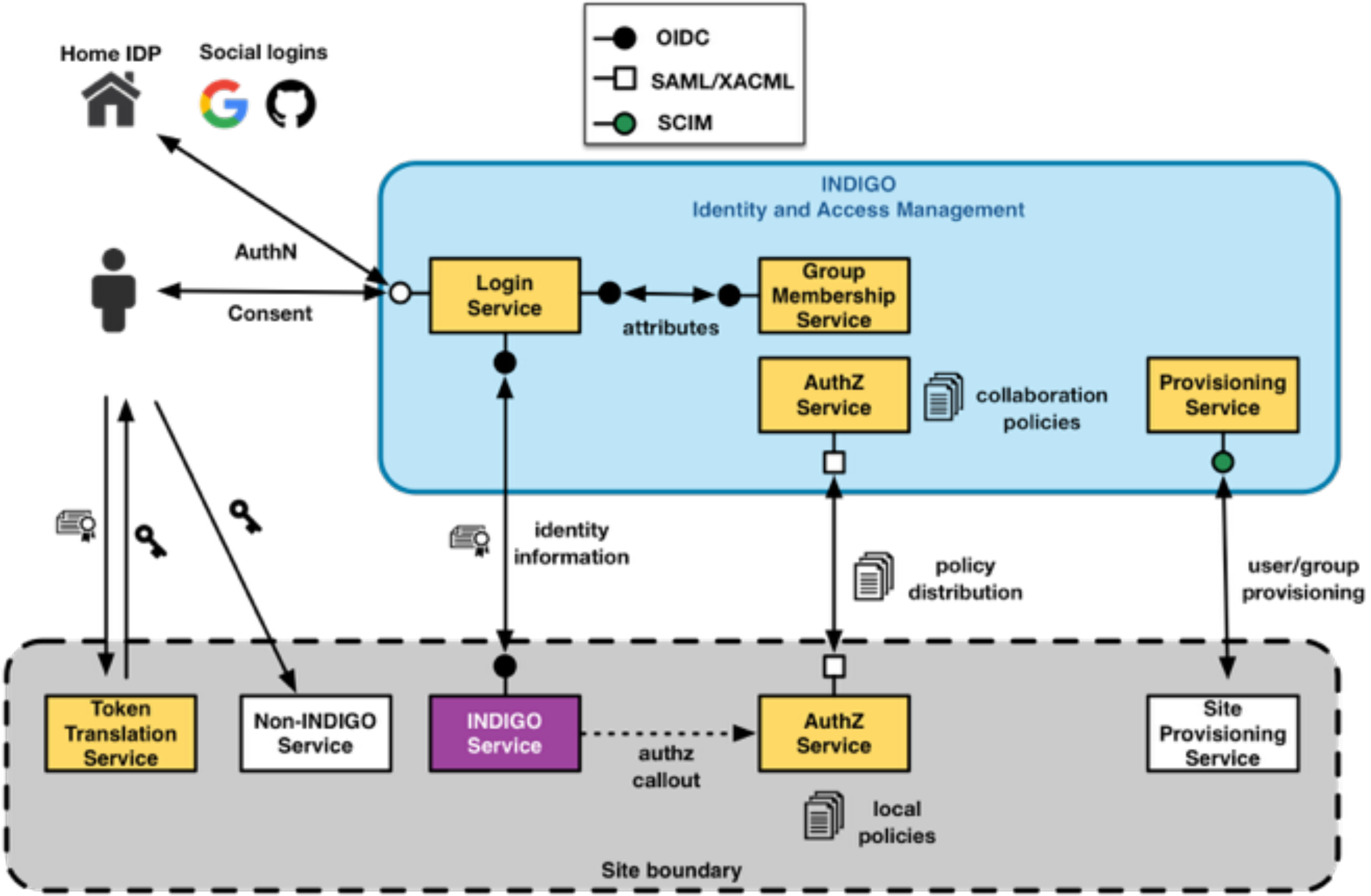
- A distributed infrastructure demands and interoperable way of propagating identity and group information to all involved resources
- INDIGO AAI relies on the standard **S**ystem for **C**ross-domain **I**dentity **M**anagement (SCIM) v. 2.0
- Indigo IAM SCIM APIs provide means to propagate identity and group information to relying services, to implement, for instance, dynamic account creation and other resource lifecycle management at various levels of the INDIGO infrastructure depending on events related to user identity status.

Token translation

- 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 currently support the generation of
    - ssh keypairs
    - X.509 certificates
    - Opennebula username/password credentials



INDIGO AAI architecture



OAuth/OIDC Grant types

Grant types

=

Flows

=

Ways for an application to get tokens

OAuth/OIDC grant types

- Authorization code
  - server-side apps acting on behalf of a user
- Resource owner password credentials
  - trusted apps, CLIs acting on behalf of a user
- Client credentials
  - server-side apps acting on behalf of themselves
- Refresh token
  - Used to refresh access tokens

We focus on these!

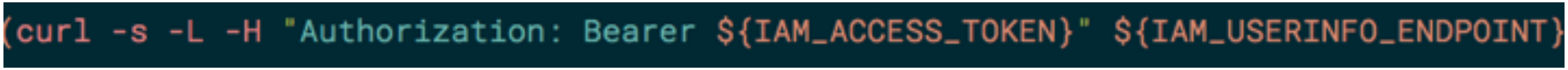
- Implicit
  - mobile/javascript apps acting on behalf of a user

OpenID-Connect

- Standard identity and single-sign on layer **built on top** of OAuth2
- Provides information about the authenticated users to client applications via a signed JSON Web Token (JWT) and a dedicated information query endpoint
  - the /userinfo endpoint
- The modern single-sign on technology
  - Widely supported in industry
    - Google, Microsoft, Oracle, PingIdentity...

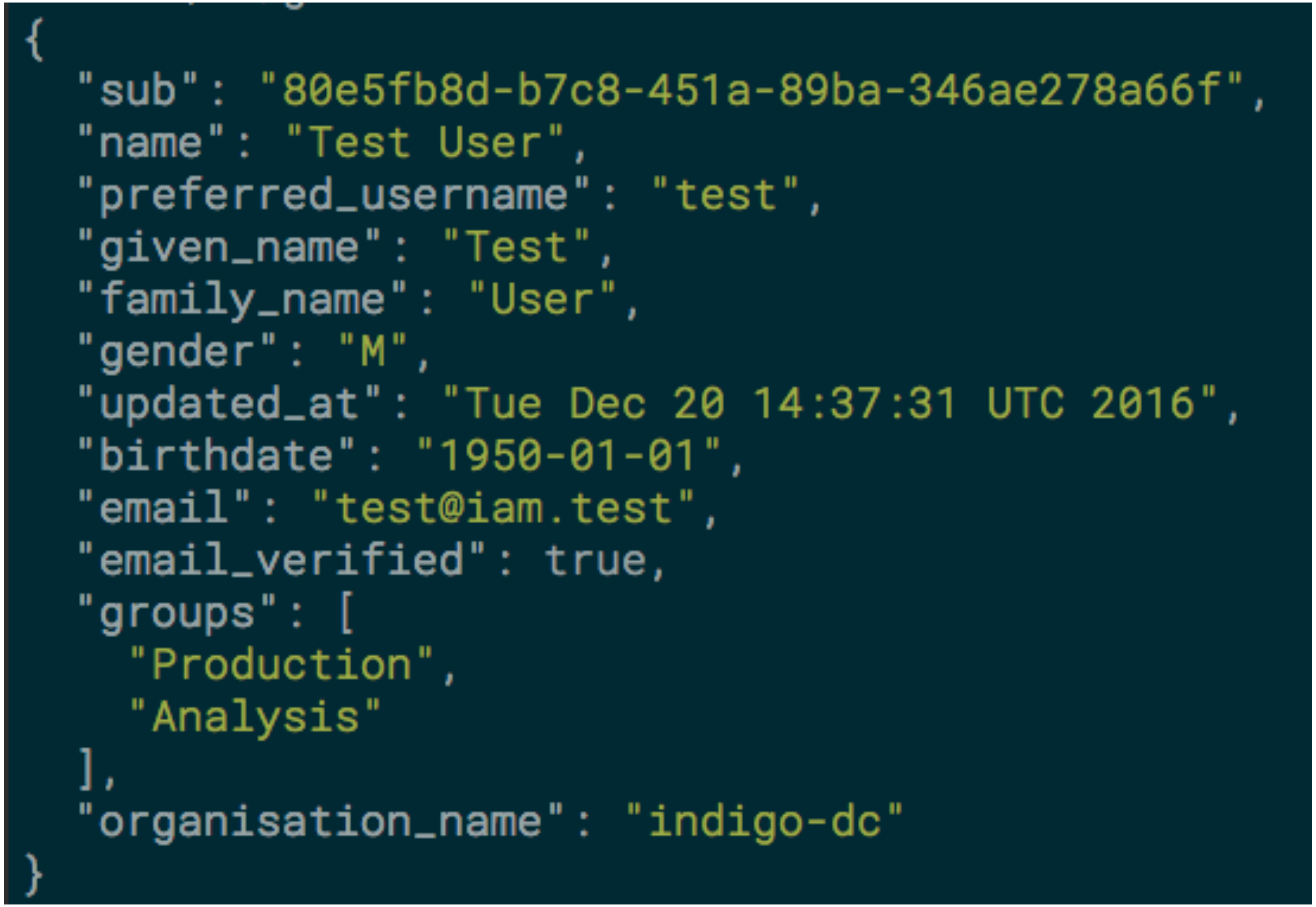
The OIDC userinfo endpoint

- A client/protected resource can get more information about the user identity by querying the /userinfo endpoint at the Authorization Server (i.e. the IAM)
- This information can be used to drive authorization decisions
- The call to the /userinfo endpoint is authorized by the access_token



```
(curl -s -L -H "Authorization: Bearer ${IAM_ACCESS_TOKEN}" ${IAM_USERINFO_ENDPOINT}
```

IAM userinfo output

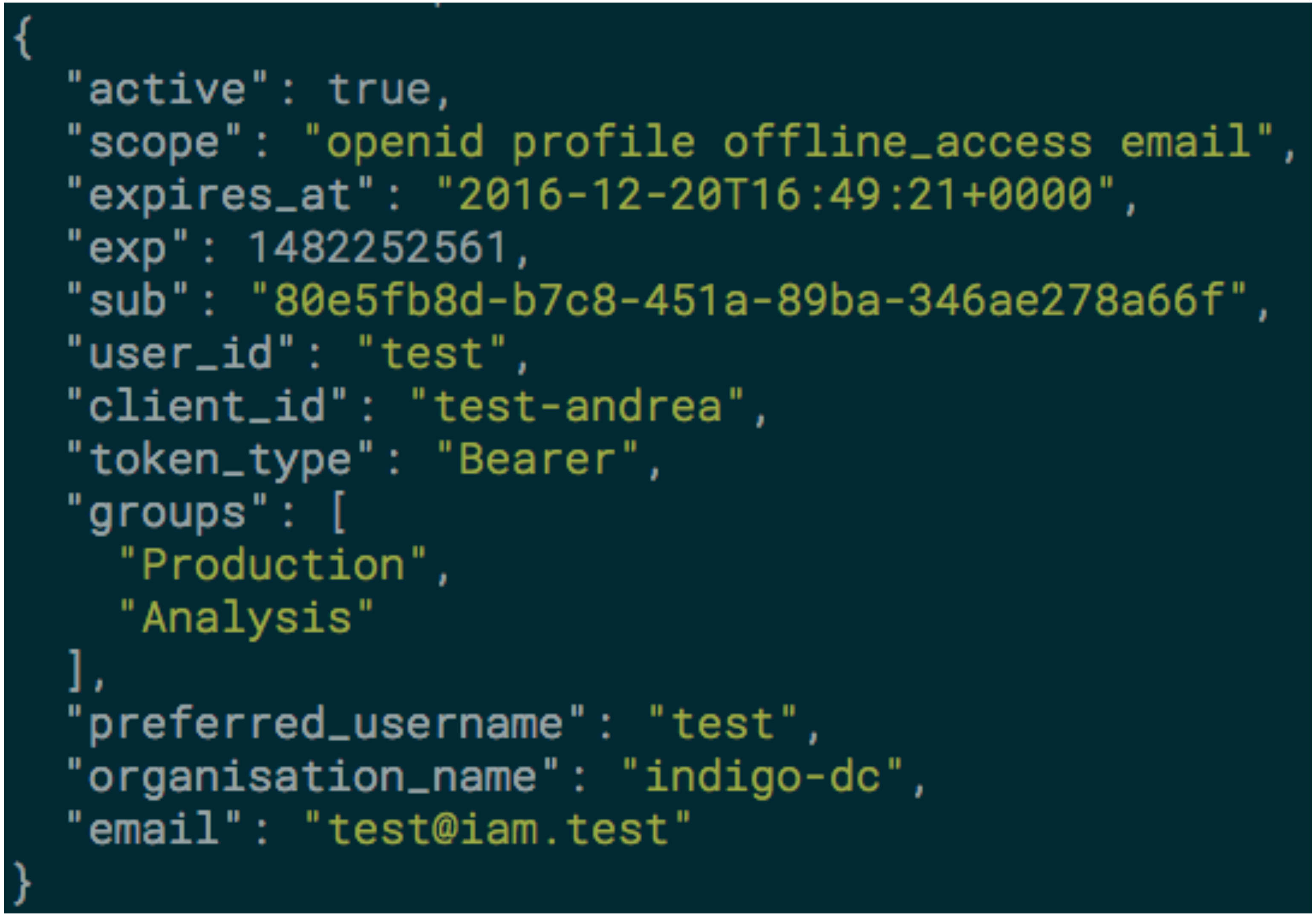


```
{
  "sub": "80e5fb8d-b7c8-451a-89ba-346ae278a66f",
  "name": "Test User",
  "preferred_username": "test",
  "given_name": "Test",
  "family_name": "User",
  "gender": "M",
  "updated_at": "Tue Dec 20 14:37:31 UTC 2016",
  "birthdate": "1950-01-01",
  "email": "test@iam.test",
  "email_verified": true,
  "groups": [
    "Production",
    "Analysis"
  ],
  "organisation_name": "indigo-dc"
}
```

Token introspection

- A client/protected resource can get more information about an access token by querying the /introspect endpoint at the Authorization Server (i.e. the IAM)
- The introspection endpoint allows
  - to handle the access token as an opaque token
  - to check token validity
  - to get additional information that is scoped to specific client application
- Calls to the introspection endpoint are authenticated via client credentials
- The introspection endpoint implements RFC 7662

Valid token introspect output

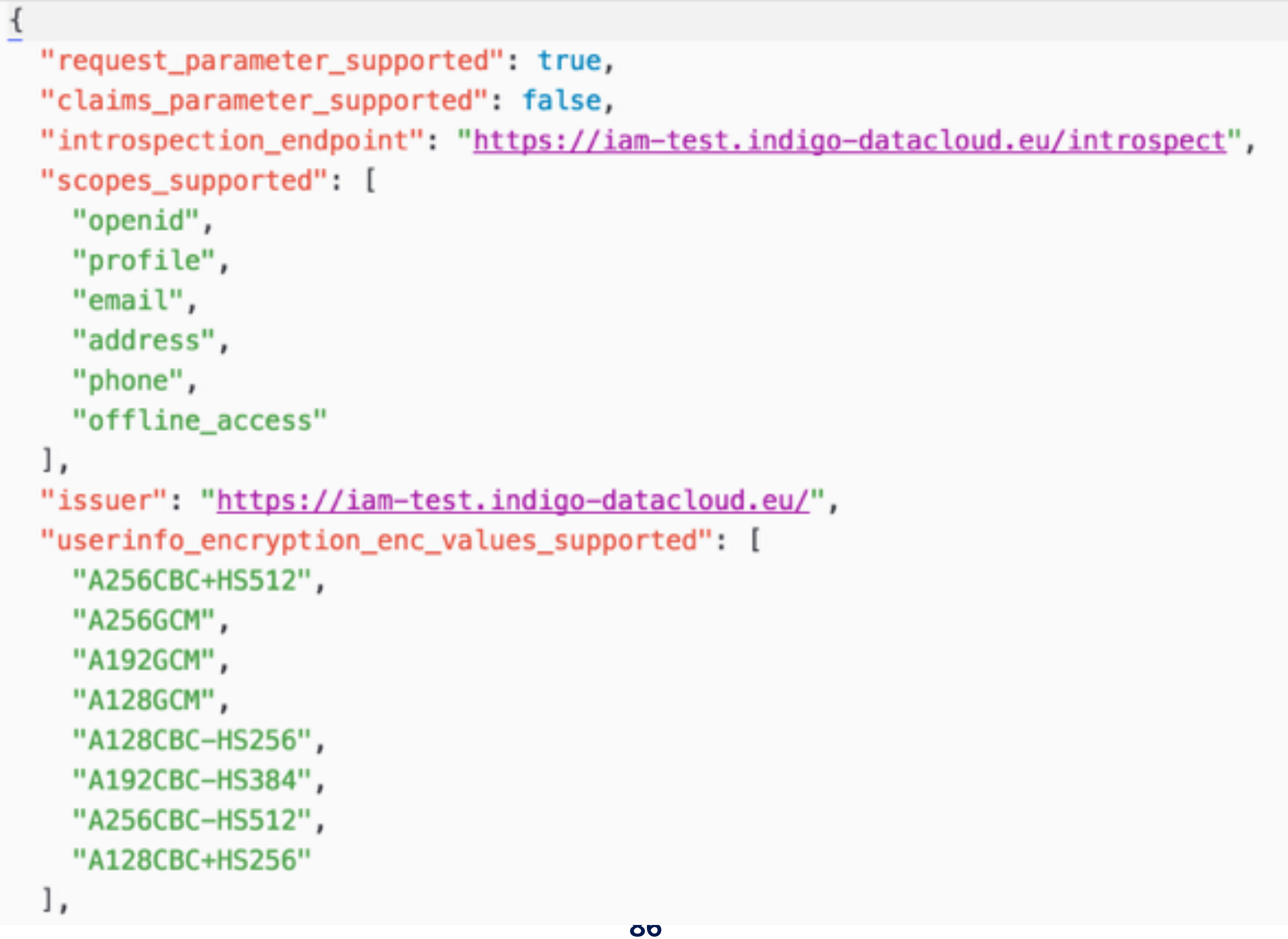


```
{
  "active": true,
  "scope": "openid profile offline_access email",
  "expires_at": "2016-12-20T16:49:21+0000",
  "exp": 1482252561,
  "sub": "80e5fb8d-b7c8-451a-89ba-346ae278a66f",
  "user_id": "test",
  "client_id": "test-andrea",
  "token_type": "Bearer",
  "groups": [
    "Production",
    "Analysis"
  ],
  "preferred_username": "test",
  "organisation_name": "indigo-dc",
  "email": "test@iam.test"
}
```

The OIDC Discovery endpoint

- The IAM provides a discovery endpoint that can be used to query information about the OpenID provider configuration
- Useful to automatically configure client libraries
- Compliant with https://openid.net/specs/openid-connect-discovery-1_0.html
- https://iam-test.indigo-datacloud.eu/.well-known/openid-configuration

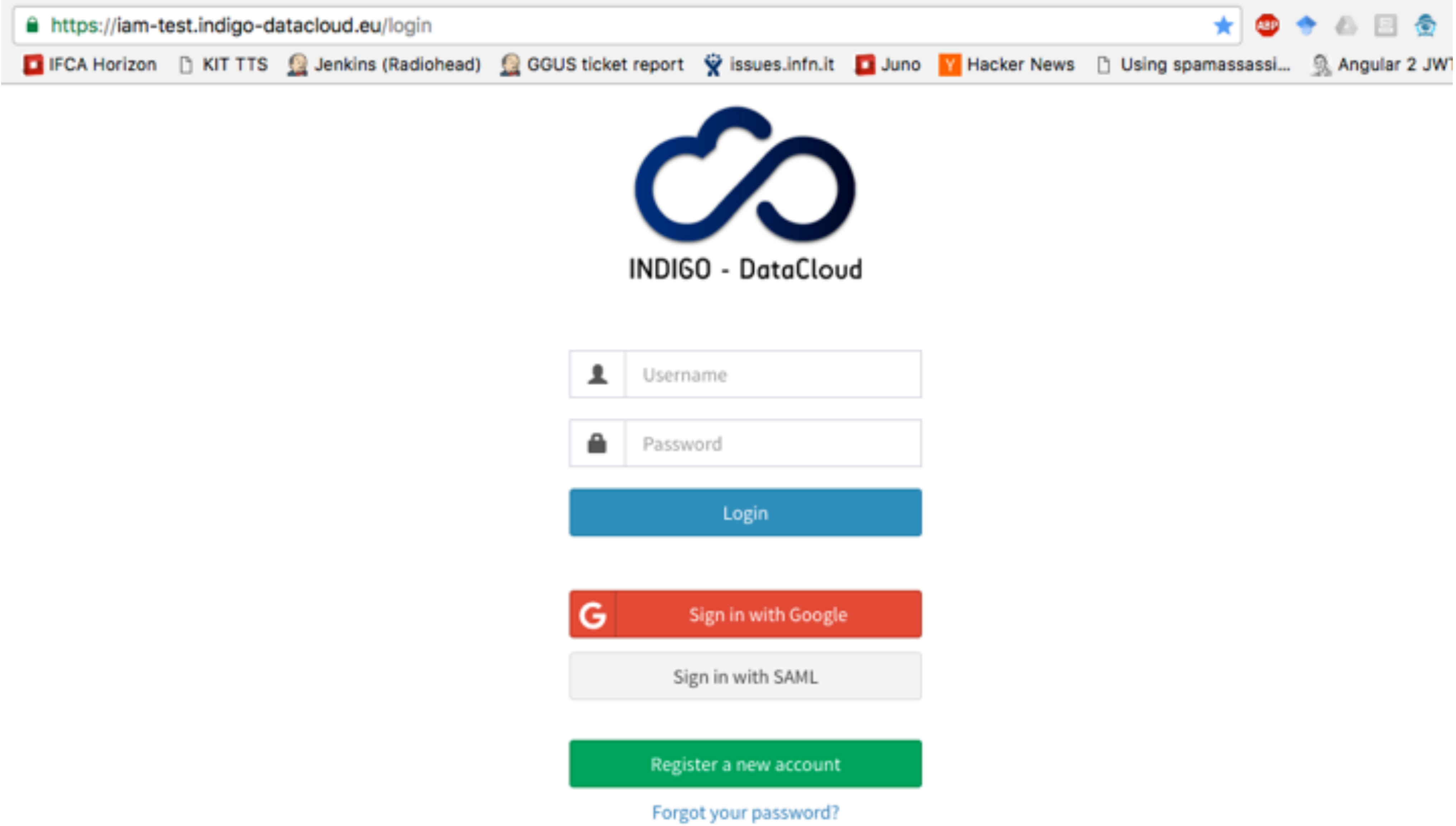
Discovery endpoint output



```
{
  "request_parameter_supported": true,
  "claims_parameter_supported": false,
  "introspection_endpoint": "https://iam-test.indigo-datacloud.eu/introspect",
  "scopes_supported": [
    "openid",
    "profile",
    "email",
    "address",
    "phone",
    "offline_access"
  ],
  "issuer": "https://iam-test.indigo-datacloud.eu/",
  "userinfo_encryption_enc_values_supported": [
    "A256CBC+HS512",
    "A256GCM",
    "A192GCM",
    "A128GCM",
    "A128CBC-HS256",
    "A192CBC-HS384",
    "A256CBC-HS512",
    "A128CBC+HS256"
  ],
```

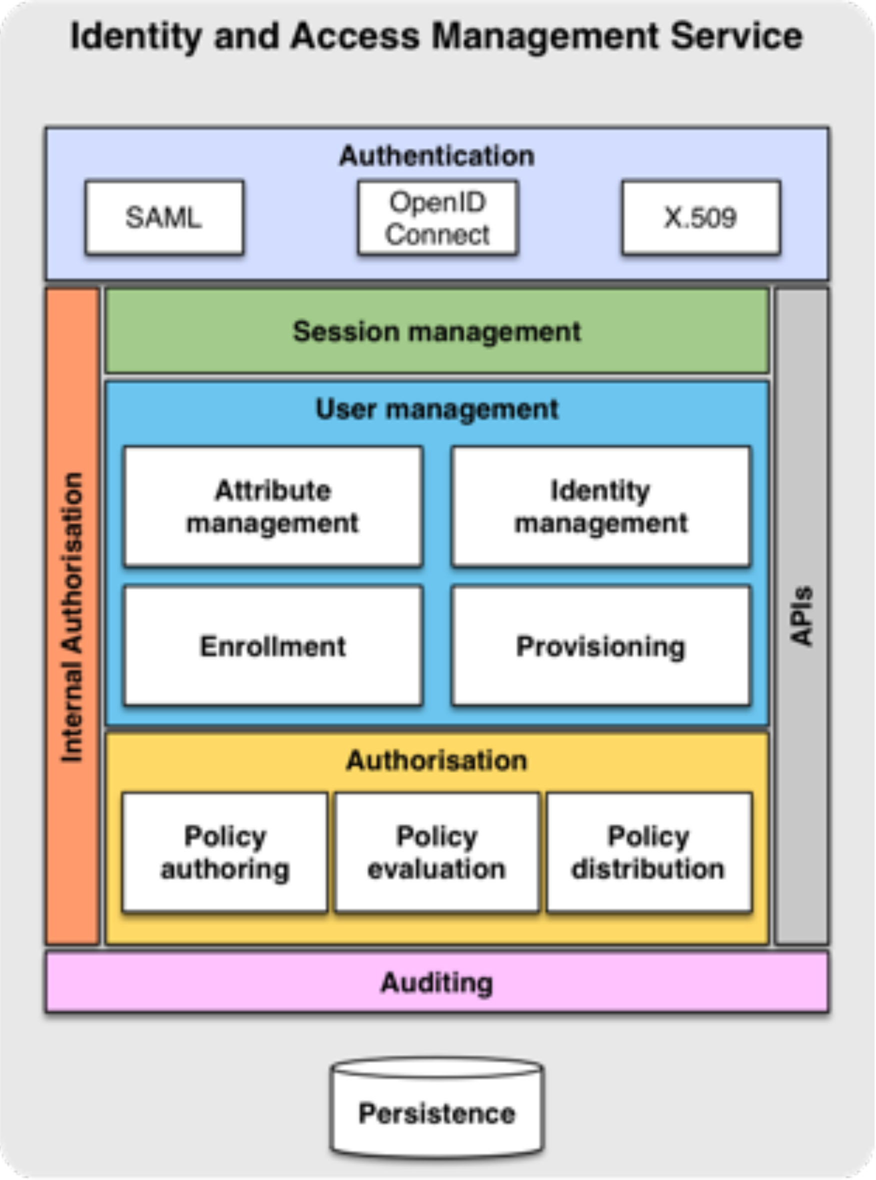
INDIGO - DataCloud

The INDIGO IAM service


https://iam-test.indigo-datacloud.eu/login
IFCA Horizon
KIT TTS
Jenkins (Radiohead)
GGUS ticket report
issues.infn.it
Juno
Hacker News
Using spamassassi...
Angular 2 JW
INDIGO - DataCloud
Username
Password
Login
Sign in with Google
Sign in with SAML
Register a new account
Forgot your password?

IAM: Goal of the service

- Provide a central service that deals with
  - User authentication (and exposes this info to RPs via standard OIDC interfaces)
  - Identity harmonization (link heterogeneous AuthN mechanisms to a single identity)
  - VO membership management
  - Registration and enrollment
  - Provisioning of VO structure and membership information to services
  - Management, distribution and enforcement of authorization policies



IAM: implementation details

- INDIGO IAM is a Spring Boot application that
  - extends and improves the certified MitreID Connect OpenID Connect implementation, and acts as a OpenID provider to relying services
  - provides
    - a registration service
    - a simple VO/collaboration management solution
    - a certified OIDC provider/OAuth authorization server
    - provisioning APIs based on SCIM standard
  - as a standalone, integrated solution

IAM deployment model

- An instance of the IAM service is deployed for each VO/ community/collaboration (following the VOMS model)
- VO/collaboration administration rights can be granted to any member of the IAM managed VO/collaboration
  - decoupling service operation from VO administration
- Administrators can
  - Manage membership requests
  - Add/remove/suspend users and grant them administrator rights
  - Organize users into groups
  - Manage client applications

Contacts and resources

- The AAI-TF wiki:
  - https://project.indigo-datacloud.eu/projects/aai-taskforce/wiki/Wiki
- The AAI-TF mailing list:
  - https://lists.indigo-datacloud.eu/sympa/lists/info/indigo-aai-tf
- The AAI-TF slack room:
  - https://indigo-aai.slack.com/
- Useful tutorial scripts:
  - https://github.com/andreaceccanti/indigo-aai-tutorial

INDIGO - DataCloud

Thanks!

Questions?

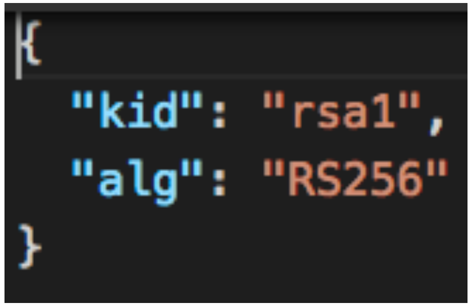
indigo-aai-tf@lists.indigo-datacloud.eu

Json Web Tokens (JWT)

- From RFC 7519:
  - **JSON Web Token** (JWT) is a compact, URL-safe means of representing claims to be transferred between two parties.
  - The claims in a JWT are **encoded as a JSON object** that is used as the payload of a JSON Web Signature (JWS) structure or as the plaintext of a JSON Web Encryption (JWE) structure, **enabling the claims to be digitally signed or integrity protected with a Message Authentication Code (MAC) and/or encrypted.**

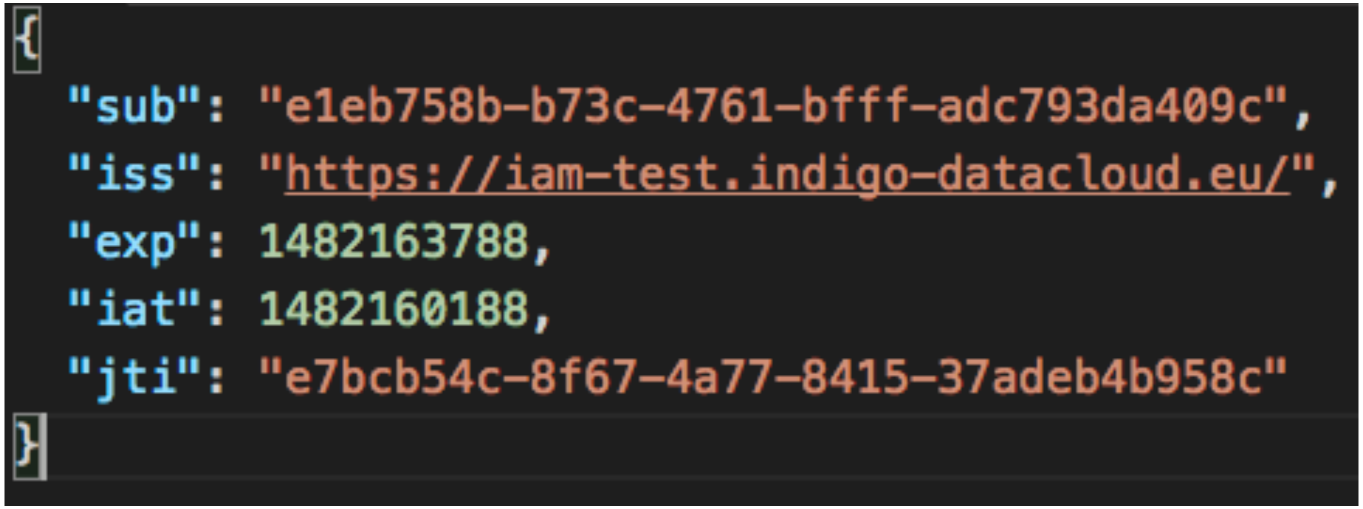
JWT: header+ body + signature

header



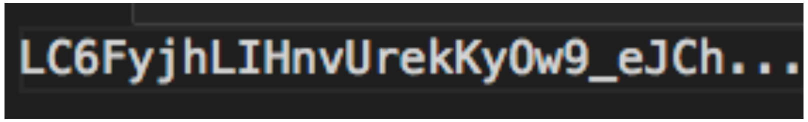
```
{
  "kid": "rsa1",
  "alg": "RS256"
}
```

body



```
{
  "sub": "e1eb758b-b73c-4761-bfff-adc793da409c",
  "iss": "https://iam-test.indigo-datacloud.eu/",
  "exp": 1482163788,
  "iat": 1482160188,
  "jti": "e7bcb54c-8f67-4a77-8415-37adeb4b958c"
}
```

signature



```
LC6FyjhLIHnvUrekKyOw9_eJCh...
```

JWT: compact serialized form

base64(header).base64(body).base64(signature)

↓

eyJraWQiOiJyc2ExIiwiYWxnIjoiUlMyNTYifQ.eyJzdWIiOiJlMWViNzU4Yi1iNzN
jLTQ3NjEtYmZmZi1hZGM3OTNkYTQwOWMiLCJpc3MiOiJodHRwczpcL1wva
WFtLXRlc3QuaW5kaWdvLWRhdGFjbG91ZC5ldVwvIiwiZXhwIjoxNDgyMTYz
Nzg4LCJpYXQiOjE0ODIxNjAxODgsImp0aSI6ImU3YmNiNTRjLThmNjctNGE
3Ny04NDE1LTM3YWRlYjRiOTU4YyJ9.LC6FyjhLIHnvUrekKyOw9_eJCh4yXU
qPJKDBmwSEZuCrCg-
uOXYkIdYobEWXKv_sTOJ6aTI1YwFu3Ayonb3MEycixD0xDaD5S0fxD2Dmm
2zFt5CPbo-Qwi_trNI67VP6-
Gzb16JINTRQxd2midAeRLAudqdebrgRRxf0DDfovQI