

DESTINATION EARTH CORE SERVICE PLATFORM



Destination Earth

Funded by
the European Union



Implemented by



IAM **INTEGRATION AND FEDERATION SCENARIOS**

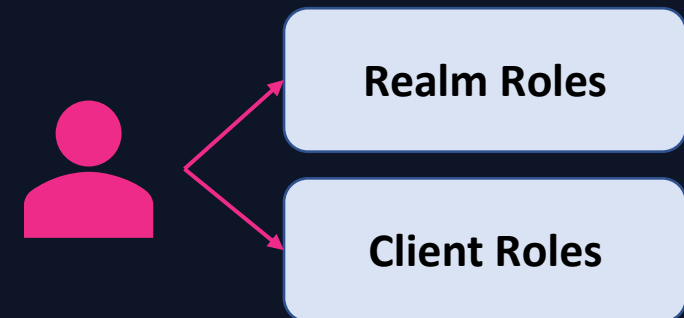
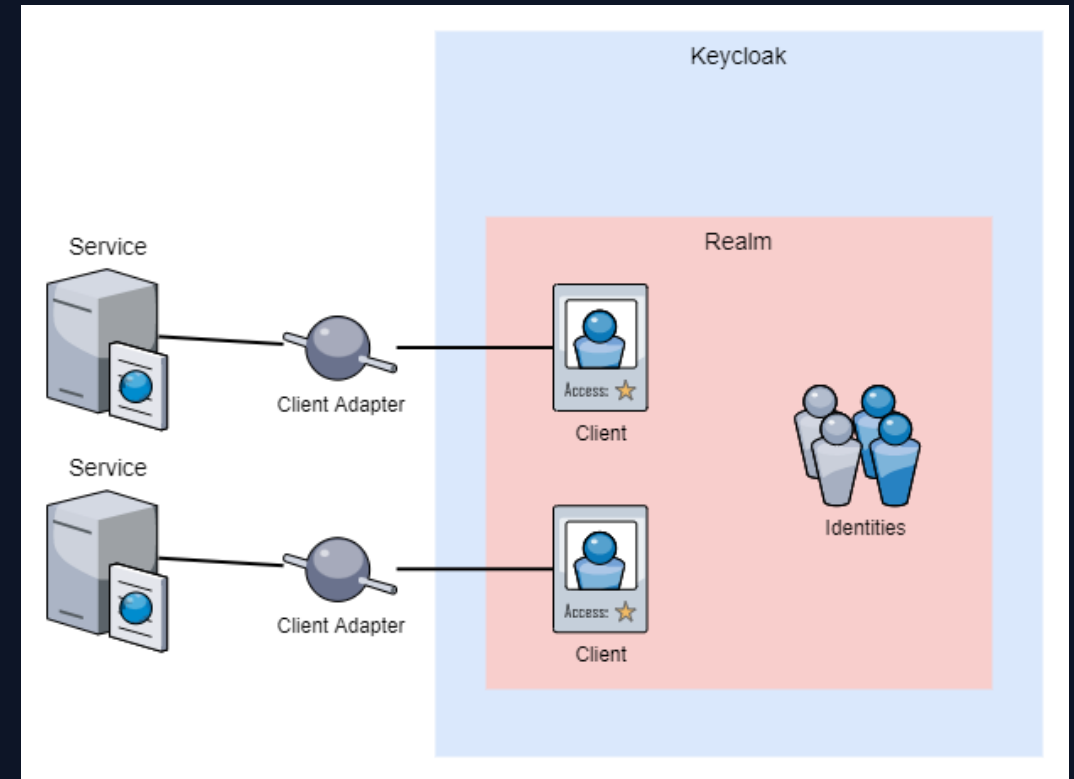


TOPICS

- **Keycloak concepts**
 - **IAM Actors**
 - **IAM Integration**
 - Self assessment
 - Procedure overview
 - Survey
 - **IAM Federation**
 - Self assessment
 - Federated Identity Provider overview
 - Federated Service overview
 - **Focus**
 - Authorization Code Flow
 - Access Token Structure – Header
 - Token Structure – Payload
 - Token Structure – Signature
 - API protection with Authorization Code Flow
 - Client Management
 - Client-role Management
 - Client-role assignment
- 
- An abstract network diagram consisting of numerous small grey dots connected by thin, light-grey lines. The dots are scattered across the right half of the slide, with a higher density towards the bottom right corner, creating a web-like structure that suggests connectivity and complexity.

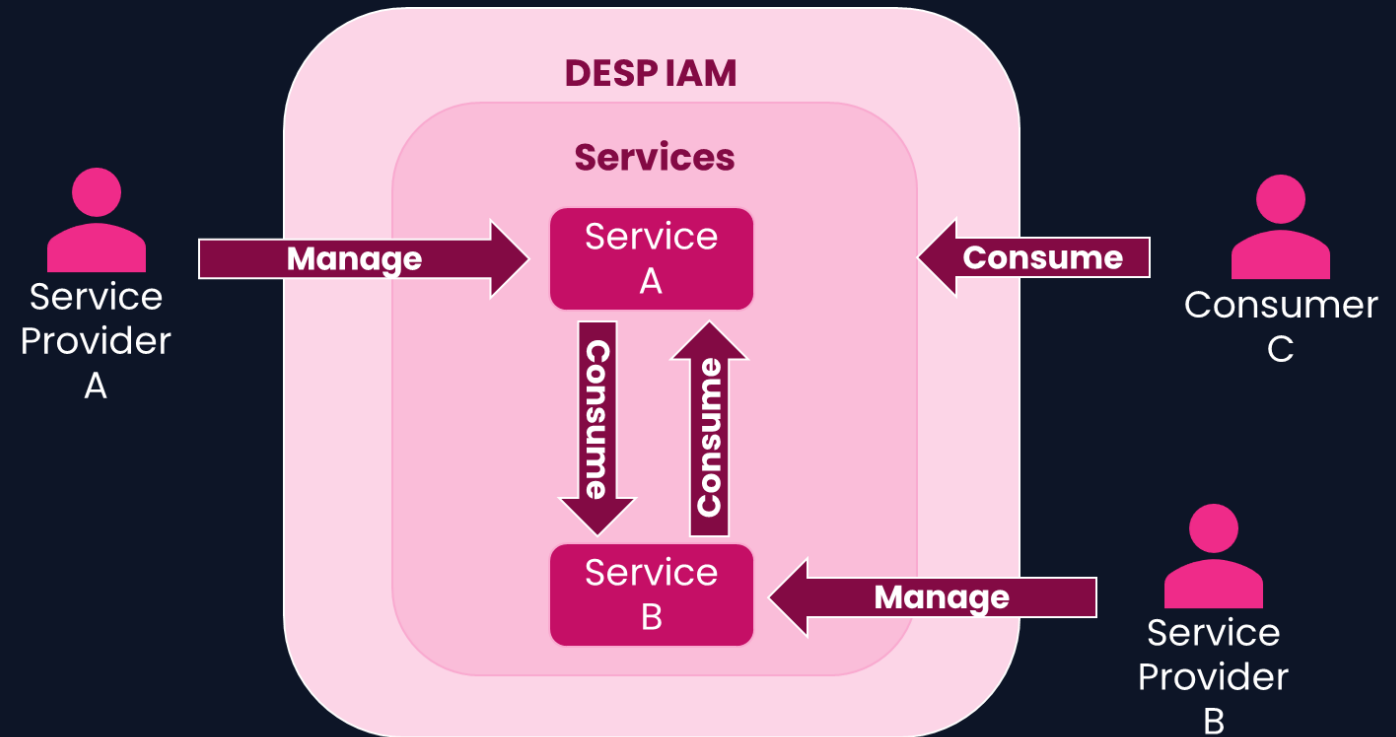
KEYCLOAK CONCEPTS

- Token-based Authorization and Authentication
- **Standard Protocols:** OpenID Connect and SAML
- **Service Provider:** Resource Provider (Dhus/TF Software)
- **Client:** Connection element between Service and Keycloak
- **Client Adapter:** Service Middleware to interact with Keycloak via the Client
- **Identities:** User/Service Accounts
- **Realm:** Logic Domain that groups Service Clients and Identities
- **Realm Roles:** Roles which implies specific behaviours over all the DHS Components.
- **Client Roles:** Role defined in the Client-domain that implies specific behaviours over a specific Service.



DESP IAM ACTORS

- **Service Provider:** Owns Software Instance/Resources to be consumed by Consumers (Users)
- **Service Consumer:** The User that consumes Services
- A Service Provider can be a Service Consumer of another DESP Service



IAM INTEGRATION

SELF ASSESSMENT

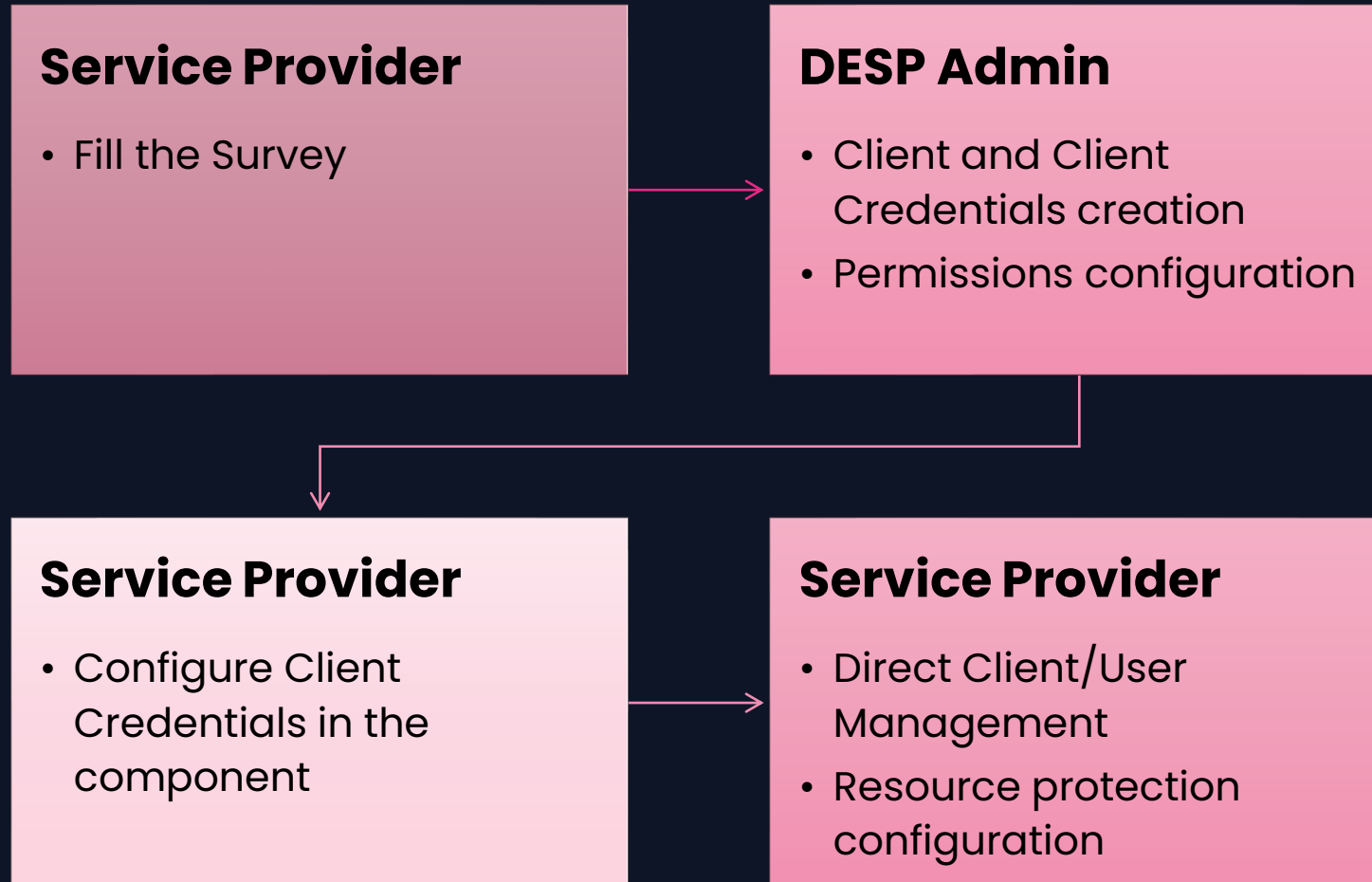
What type of Application do I want to protect?

Application Type	Client Type	Auth and Authz Flow
<ul style="list-style-type: none">• Single Page App• Client-side App without backend• Mobile App	<ul style="list-style-type: none">• Public Client	Authorization Code
<ul style="list-style-type: none">• API	<ul style="list-style-type: none">• Confidential Client (Bearer) <u>and</u>• Public Client	Authorization Code
<ul style="list-style-type: none">• Browser Based Application with frontend and/or backend	<ul style="list-style-type: none">• Confidential Client	Authorization Code

- Pre-Requirements:

- Deployable Software Component
- Identified Keycloak/OIDC/OAuth provider Integration method (library, proxy)

IAM INTEGRATION PROCEDURE OVERVIEW



IAM INTEGRATION SURVEY

Field	Description
Name	Human-Readable Name to assign to the Client
Description	Application brief description
Client Authentication Type	<p>1. Public Client: This when your application is entirely public and it cannot safely store client secret. Eg. Single Page Javascript application without any backend which does not have any way of protecting client secret.</p> <p>2. Confidential Client: This is when the client application can store secret safely on the application backend side (secret will not be available in the javascript files served)</p>
Home URL	Default URL to use when the auth server needs to redirect or link back to the client.
Valid Redirect URIs	Valid URI pattern a browser can redirect to after a successful login. Simple wildcards are allowed such as ' http://example.com/ '
Valid Post Logout redirect URIs	Valid URI pattern a browser can redirect to after a successful logout. Simple wildcards are allowed such as ' http://example.com/ '.
Web Origins	List of allowed CORS origins.
List of users to be Client Admins	Email, First Name, Last Name

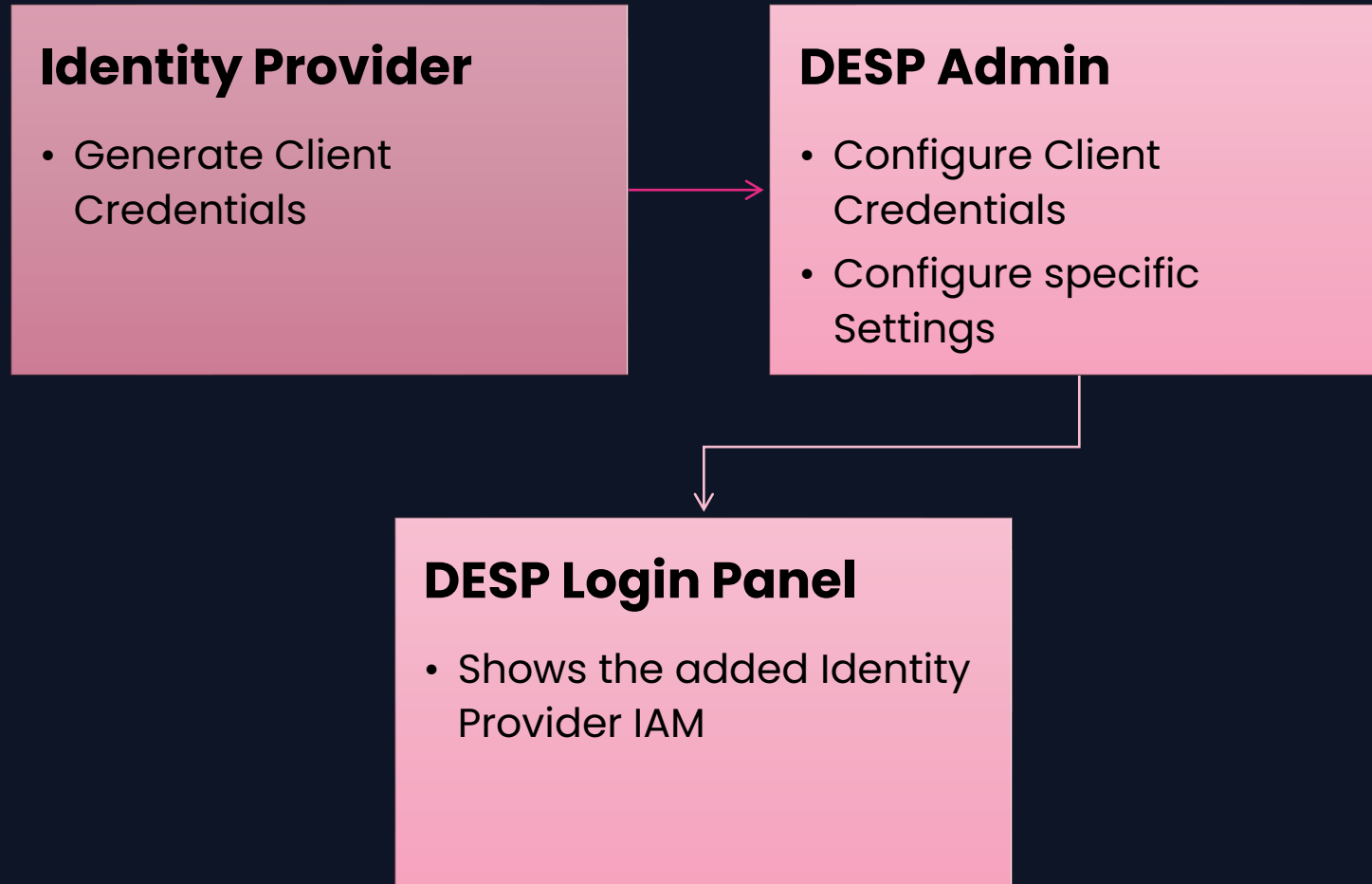
IAM FEDERATION

SELF ASSESSMENT

- Pre-Requirements:
 - Existing Operational Service
 - User Identities managed by an OIDC/SAML compatible IAM
- What type of Federation I need?
 - I want to give DESP Platform Access to my existing Users -> **Federated Identity Provider**
 - I want to give access to my existing Service to the DESP Users:
 - Do I need an Accounting service?
 - YES -> **Federated Service**
 - NO -> **External Federated Service**

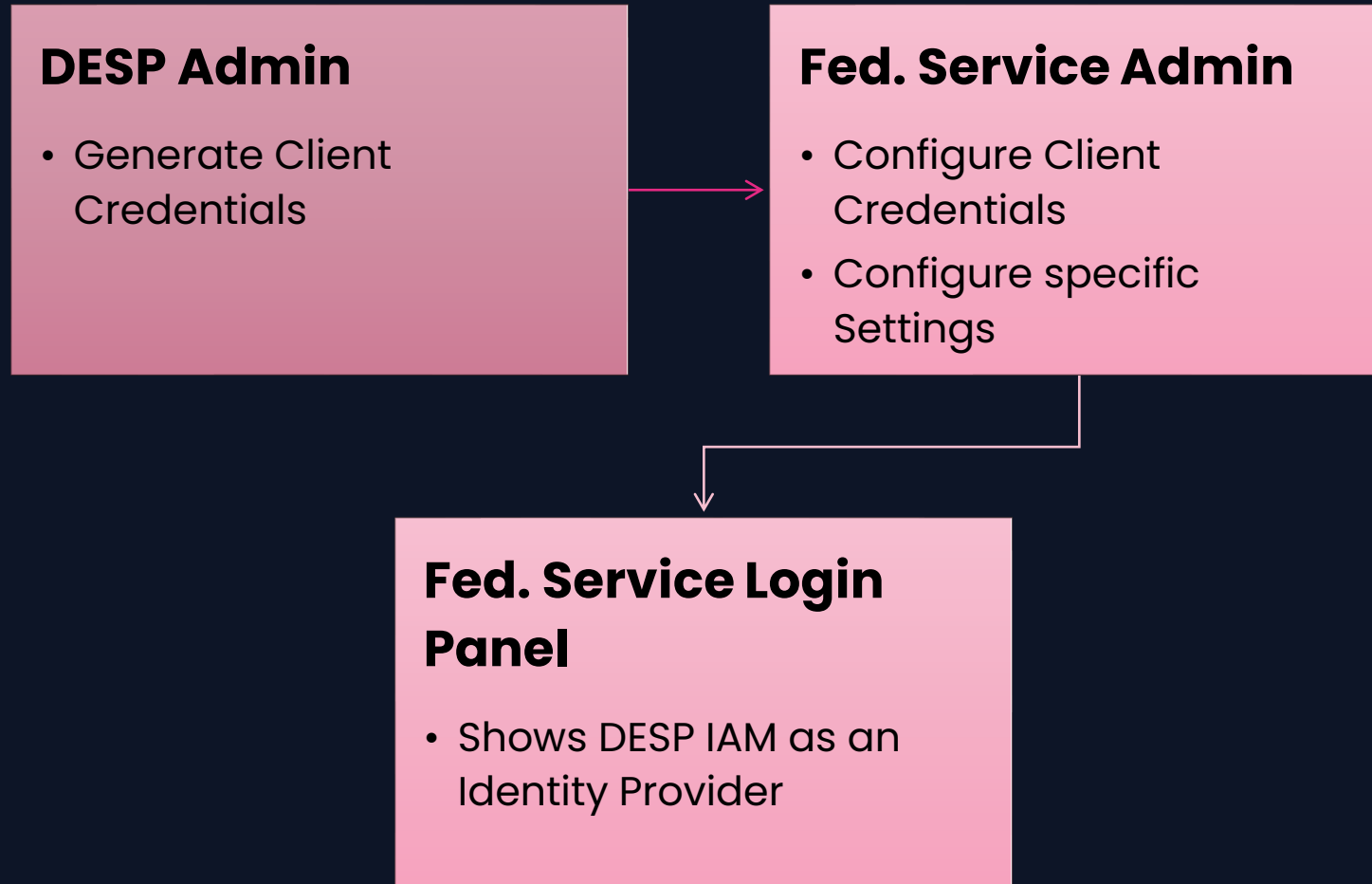
IAM FEDERATION

FEDERATED IDENTITY PROVIDER OVERVIEW

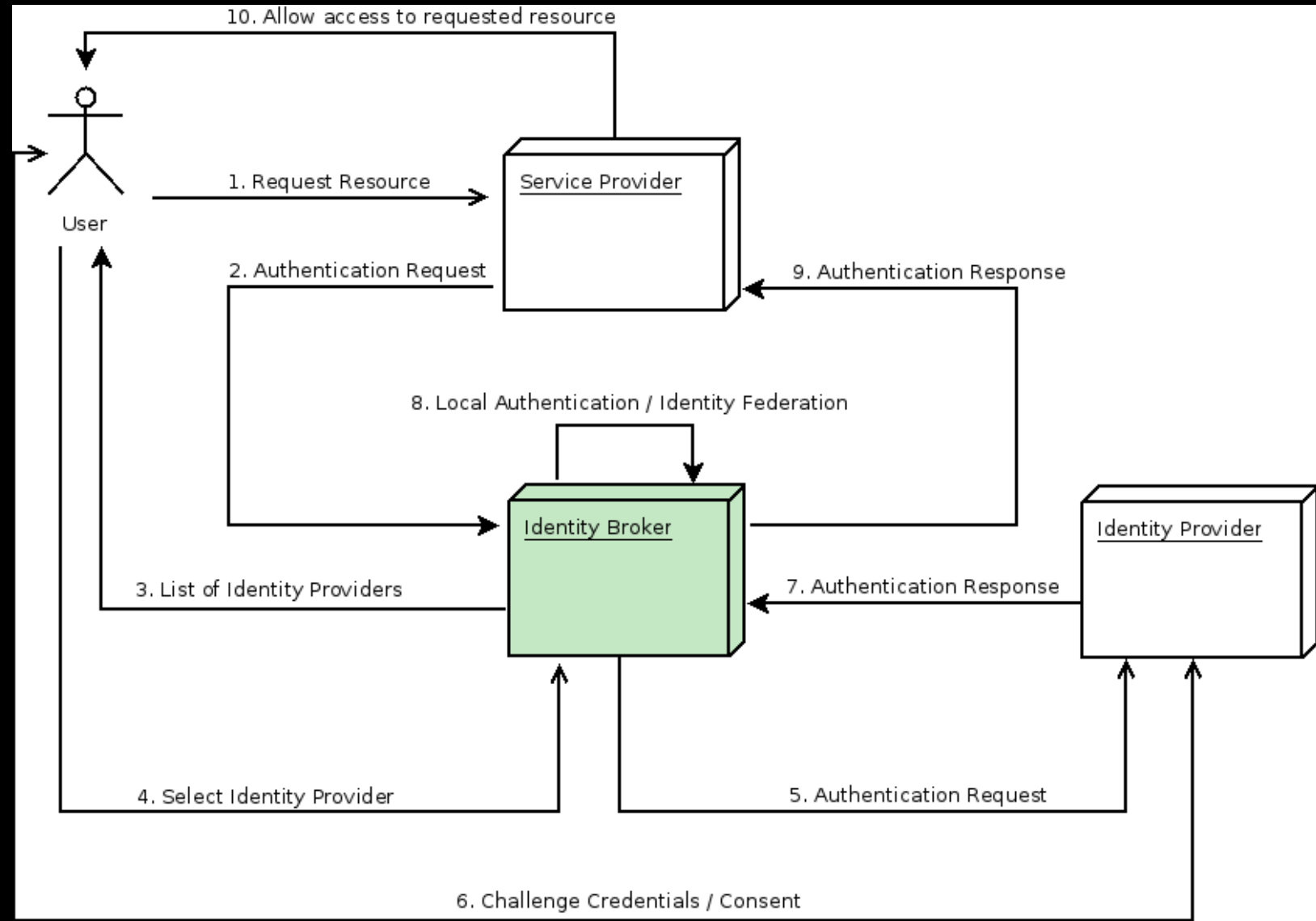


IAM FEDERATION

FEDERATED SERVICE OVERVIEW



FOCUS Authorization Code Flow



FOCUS

Authorization Code Flow

- Credential Validation happens through the IAM:
 - User is redirected to the IAM to input the Credential
- Credential Validation Output:
 - **Identity Token**: proofs the User has been authenticated
 - **Access Token**: contains User access rules
 - **Refresh Token**: used to refresh the Access Token without new login by the User

FOCUS

Access Token Structure - Header

A token consists of three Base64-URL-encoded strings separated by dots:

- **Header** -> specifies the token type and the signing algorithm

```
{
```

```
  "alg": "RS256",
```

```
  "typ": "JWT",
```

```
  "kid": "zgeOJ-OVBIEwd6M5nSDjftW7uXjJkxq17..."
```

```
}
```

FOCUS

Token Structure - Payload

- **Payload** -> specifies the token issuer, the token expiration, and claims, which are pieces of information regarding the user. Please note that the claims follow the “least information” principle.
- Email field in the Tokens (ID and Access) will be available only after a proven need and after the GDPR compliance evaluation.
- The User realm-roles can be available as a claim in the Access Token after proper evaluation.

```
{
  "exp": 1704381460,
  "iat": 1704381160,
  "auth_time": 1704381160,
  ...
  "iss": "https://iam.ivv.desp.space/realms/desp",
  "sub": "ad7874e0-f21b-4801-bf2a-8ef698d726e2",
  "typ": "Bearer",
  "azp": "danpen-cl",
  ...
  "client_roles": [
    "dp-admin"
  ],
  ...
  "email": "daniele.pensa@serco.com"
}
```

FOCUS

Token Structure - Signature

- **Signature** -> certifying that the message hasn't been corrupted in transit. It is produced by combining the encoded header, the encoded payload, a secret, the algorithm defined in the header, and signing it. It contains the relevant elements to validate the token through the issuer (IAM).

```
{  
  "e": "AQAB",  
  "kty": "RSA",  
  "n":  
  "o4K9K882C_nuRMeFs4zc7VZxZFJj0OLBQRqIL3AkbH8rUnUb3KJAFQQ2IWABsfkSTZjm..."  
}
```


FOCUS

API protection with Authorization Code Flow

- Phases:

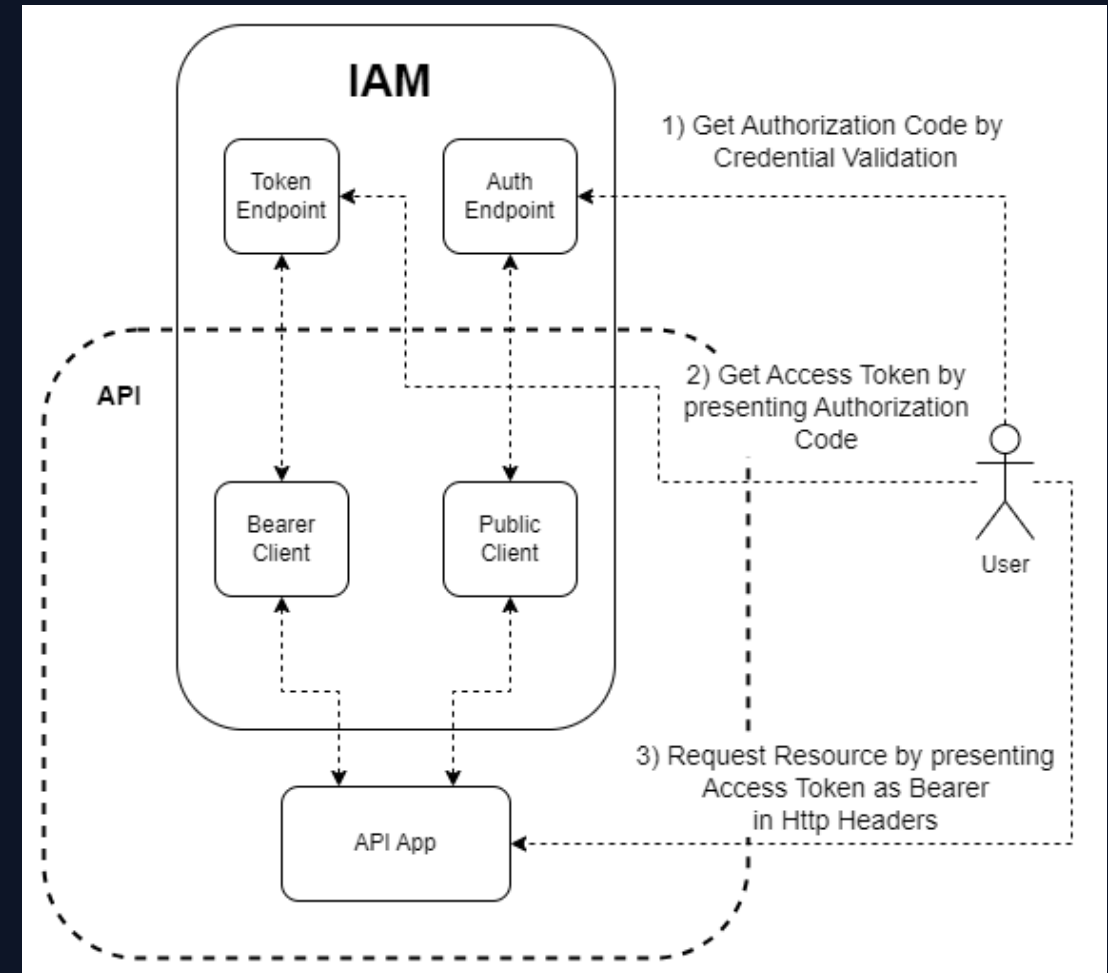
1. Obtain Authorization Code:

- Credential are exchanged only with the IAM
- The protocol uses Public Client parameters (client_id, redirect_uri)

2. Obtain Access Token

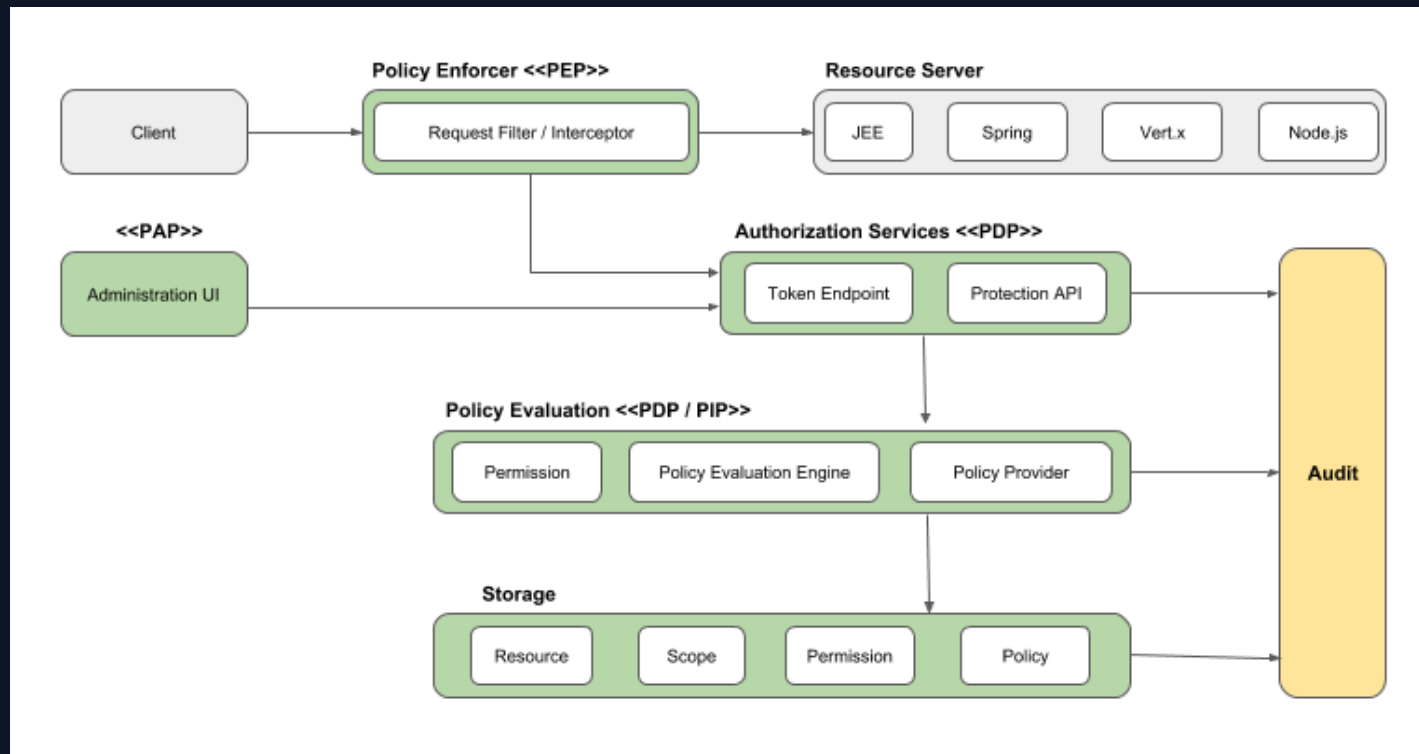
3. Request Resource

- Public and Bearer Clients are managed by service Provider
- The flow is fully scriptable (tested in bash and python)



FOCUS



CLIENT MANAGEMENT



**image from official Keycloak website*

- Protected Resources configuration
- Client-Level Roles
- Client-Level Authorization Policies
- User-Client Roles/Policies Mapping

FOCUS CLIENT MANAGEMENT



desp ▾

Manage

Clients

Client scopes

Users

Groups

Configure


Clients

Clients are applications and services that can request authentication of a user. [Learn more](#)


Clients list

Initial access token

Client registration



Search for client



1-1 ▾ < >

Client ID	Name	Type	Description	Home URL
test-client-a	test-client-a	OpenID Connect	Test OpenID client-a for Demo	http://localhost:8700 ⋮

1-1 ▾ < >

FOCUS CLIENT-ROLE MANAGEMENT

The screenshot displays the Keycloak Admin Console interface. On the left is a dark sidebar with navigation options: Manage, Clients (highlighted), Client scopes, Users, Groups, and Configure. Above the sidebar is a header with a menu icon, the Keycloak logo, and the user 'Francesco Pezzullo'. The main content area shows the 'test-client-a' details page, specifically the 'Roles' tab. It includes a search bar for roles, a 'Create role' button, and a table listing existing roles.

test-client-a OpenID Connect Enabled Action

Clients are applications and services that can request authentication of a user.

Settings Keys Credentials **Roles** Client scopes Sessions Advanced

Search role by name → Create role 1-2 < >

Role name	Composite	Description
test_client_role	False	–
uma_protection	False	–

1-2 < >

FOCUS CLIENT-ROLE ASSIGNMENT

The screenshot shows a web application interface with a dark sidebar and a light main content area. A modal window is open in the center, titled "Assign roles to daniele.pensa@serco.com". The modal has a close button (X) in the top right corner. Inside the modal, there are two search filters: "Filter by clients" with a dropdown arrow and "Search by role name" with a search icon and a right arrow button. Below these filters is a table with two columns: "Name" and "Description". The table contains two rows of data, both with a "test-client-a" client name. The first row has a checked checkbox and the role name "test_client_role". The second row has an unchecked checkbox and the role name "uma_protection". At the bottom of the modal, there are two buttons: "Assign" (in blue) and "Cancel" (in light blue). The background interface shows a sidebar with a menu icon, a logo, and a user profile "Francesco Pezzullo". The main content area has a breadcrumb "Users > User details" and a search bar.

Assign roles to daniele.pensa@serco.com

Filter by clients Search by role name

<input type="checkbox"/>	Name	Description
<input checked="" type="checkbox"/>	test-client-a test_client_role	
<input type="checkbox"/>	test-client-a uma_protection	

Assign Cancel



DESTINATION EARTH CORE SERVICE PLATFORM

For info: daniele.pensa@serco.com

serco

ThalesAlenia
a Thales / Leonardo company
Space

CGI

expri^{via}

deimos
elecnor group

MEEO⁸⁰

 **aliaspace**

 **OVHcloud**