Login Service Design Document

EOEPCA.SDD.xxx

TVUK System Team

Version 0.1, dd/mm/yyyy:

Login Service Design Document

EO Exploitation Platform Common Architecture

Login Service Design Document

EOEPCA.SDD.xxx

COMMENTS and ISSUES	**PDF**
If you would like to raise comments or issues on this document, please do so by raising an Issue at the following URL https://github.com/EOEPCA/um-login-service/issues.	This document is available in PDF format here.
EUROPEAN SPACE AGENCY CONTRACT REPORT The work described in this report was done under ESA contract. Responsibility for the contents resides in the author or organisation that prepared it.	**TELESPAZIO VEGA UK Ltd** 350 Capability Green, Luton, Bedfordshire, LU1 3LU, United Kingdom. Tel: +44 (0)1582 399000 www.telespazio-vega.com

AMENDMENT HISTORY

This document shall be amended by releasing a new edition of the document in its entirety. The Amendment Record Sheet below records the history and issue status of this document.

Table 1. Amendment Record Sheet

ISSUE	DATE	REASON
0.1	dd/mm/yyyy	Initial in-progress draft

Chapter 1. Introduction

1.1. Purpose and Scope

This document presents the Login Service Design for the Common Architecture.

1.2. Structure of the Document

Section 2 - Overview

Provides an overview of the Login Service component, within the context of the wider Common Architecture design.

Section 3 - Building Block Design

Provides the building block design of the Login Service component.

1.3. Reference Documents

The following is a list of Reference Documents with a direct bearing on the content of this document.

Reference	Document Details	Version
[EOEPCA-UC]	EOEPCA - Use Case Analysis EOEPCA.TN.005 https://eoepca.github.io/use-case-analysis	Issue 1.0, 02/08/2019
[EP-FM]	Exploitation Platform - Functional Model, ESA-EOPSDP-TN-17-050	Issue 1.0, 30/11/2017
[TEP-OA]	Thematic Exploitation Platform Open Architecture, EMSS-EOPS-TN-17-002	Issue 1, 12/12/2017
[WPS-T]	OGC Testbed-14: WPS-T Engineering Report, OGC 18-036r1, http://docs.opengeospatial.org/per/18-036r1.html	18-036r1, 07/02/2019
[WPS-REST-JSON]	OGC WPS 2.0 REST/JSON Binding Extension, Draft, OGC 18-062, https://raw.githubusercontent.com/opengeospatial/wps-rest-binding/develop/docs/18-062.pdf	1.0-draft
[CWL]	Common Workflow Language Specifications, https://www.commonwl.org/v1.0/	v1.0.2

Reference	Document Details	Version
[TB13-AP]	OGC Testbed-13, EP Application Package Engineering Report, OGC 17-023, http://docs.opengeospatial.org/per/17-023.html	17-023, 30/01/2018
[TB13-ADES]	OGC Testbed-13, Application Deployment and Execution Service Engineering Report, OGC 17-024, http://docs.opengeospatial.org/per/17-024.html	17-024, 11/01/2018
[TB14-AP]	OGC Testbed-14, Application Package Engineering Report, OGC 18-049r1, http://docs.opengeospatial.org/per/18-049r1.html	18-049r1, 07/02/2019
[TB14-ADES]	OGC Testbed-14, ADES & EMS Results and Best Practices Engineering Report, OGC 18-050r1, http://docs.opengeospatial.org/per/18-050r1.html	18-050r1, 08/02/2019
[OS-GEO-TIME]	OpenSearch GEO: OpenSearch Geo and Time Extensions, OGC 10-032r8, http://www.opengeospatial.org/standards/opensearchgeo	10-032r8, 14/04/2014
[OS-EO]	OpenSearch EO: OGC OpenSearch Extension for Earth Observation, OGC 13-026r9, http://docs.opengeospatial.org/is/13-026r8/13-026r8.html	13-026r9, 16/12/2016
[GEOJSON-LD]	OGC EO Dataset Metadata GeoJSON(-LD) Encoding Standard, OGC 17-003r1/17-084	17-003r1/17-084
[GEOJSON-LD-RESP]	OGC OpenSearch-EO GeoJSON(-LD) Response Encoding Standard, OGC 17-047	17-047
[PCI-DSS]	The Payment Card Industry Data Security Standard, https://www.pcisecuritystandards.org/document_library?category=pcidss&document=pci_dss	v3.2.1
[CEOS-OS-BP]	CEOS OpenSearch Best Practise, http://ceos.org/ourwork/workinggroups/wgiss/access/opensearch/	v1.2, 13/06/2017
[OIDC]	OpenID Connect Core 1.0, https://openid.net/specs/openid-connect-core-1_0.html	v1.0, 08/11/2014

Reference	Document Details	Version
[OGC-CSW]	OGC Catalogue Services 3.0 Specification - HTTP Protocol Binding (Catalogue Services for the Web), OGC 12-176r7, http://docs.opengeospatial.org/is/12-176r7/12-176r7.html	v3.0, 10/06/2016
[OGC-WMS]	OGC Web Map Server Implementation Specification, OGC 06-042, http://portal.opengeospatial.org/files/?artifact_id=14416	v1.3.0, 05/03/2006
[OGC-WMTS]	OGC Web Map Tile Service Implementation Standard, OGC 07-057r7, http://portal.opengeospatial.org/files/?artifact_id=35326	v1.0.0, 06/04/2010
[OGC-WFS]	OGC Web Feature Service 2.0 Interface Standard – With Corrigendum, OGC 09-025r2, http://docs.opengeospatial.org/is/09-025r2/09-025r2.html	v2.0.2, 10/07/2014
[OGC-WCS]	OGC Web Coverage Service (WCS) 2.1 Interface Standard - Core, OGC 17-089r1, http://docs.opengeospatial.org/is/17-089r1/17-089r1.html	v2.1, 16/08/2018
[OGC-WCPS]	Web Coverage Processing Service (WCPS) Language Interface Standard, OGC 08-068r2, http://portal.opengeospatial.org/files/?artifact_id=32319	v1.0.0, 25/03/2009
[AWS-S3]	Amazon Simple Storage Service REST API, https://docs.aws.amazon.com/AmazonS3/latest/API	API Version 2006-03-01

1.4. Terminology

The following terms are used in the Master System Design.

Term	Meaning
Admin	User with administrative capability on the EP
Algorithm	A self-contained set of operations to be performed, typically to achieve a desired data manipulation. The algorithm must be implemented (codified) for deployment and execution on the platform.
Analysis Result	The *Products* produced as output of an *Interactive Application* analysis session.

Term	Meaning
Analytics	A set of activities aimed to discover, interpret and communicate meaningful patters within the data. Analytics considered here are performed manually (or in a semi-automatic way) on-line with the aid of *Interactive Applications*.
Application Artefact	The 'software' component that provides the execution unit of the *Application Package*.
Application Deployment and Execution Service (ADES)	WPS-T (REST/JSON) service that incorporates the Docker execution engine, and is responsible for the execution of the processing service (as a WPS request) within the ‘target’ Exploitation Platform.
Application Descriptor	A file that provides the metadata part of the *Application Package*. Provides all the metadata required to accommodate the processor within the WPS service and make it available for execution.
Application Package	A platform independent and self-contained representation of a software item, providing executable, metadata and dependencies such that it can be deployed to and executed within an Exploitation Platform. Comprises the *Application Descriptor* and the *Application Artefact*.
Bulk Processing	Execution of a *Processing Service* on large amounts of data specified by AOI and TOI.
Code	The codification of an algorithm performed with a given programming language - compiled to Software or directly executed (interpretted) within the platform.
Compute Platform	The Platform on which execution occurs (this may differ from the Host or Home platform where federated processing is happening)
Consumer	User accessing existing services/products within the EP. Consumers may be scientific/research or commercial, and may or may not be experts of the domain
Data Access Library	An abstraction of the interface to the data layer of the resource tier. The library provides bindings for common languages (including python, Javascript) and presents a common object model to the code.
Development	The act of building new products/services/applications to be exposed within the platform and made available for users to conduct exploitation activities. Development may be performed inside or outside of the platform. If performed outside, an integration activity will be required to accommodate the developed service so that it is exposed within the platform.
Discovery	User finds products/services of interest to them based upon search criteria.
Execution	The act to start a *Processing Service* or an *Interactive Application*.

Term	Meaning
Execution Management Service (EMS)	The EMS is responsible for the orchestration of workflows, including the possibility of steps running on other (remote) platforms, and the on-demand deployment of processors to local/remote ADES as required.
Expert	User developing and integrating added-value to the EP (Scientific Researcher or Service Developer)
Exploitation Tier	The Exploitation Tier represents the end-users who exploit the services of the platform to perform analysis, or using high-level applications built-in on top of the platform's services
External Application	An application or script that is developed and executed outside of the Exploitation Platform, but is able to use the data/services of the EP via a programmatic interface (API).
Guest	An unregistered User or an unauthenticated Consumer with limited access to the EP's services
Home Platform	The Platform on which a User is based or from which an action was initiated by a User
Host Platform	The Platform through which a Resource has been published
Identity Provider (IdP)	The source for validating user identity in a federated identity system, (user authentication as a service).
Interactive Application	A stand-alone application provided within the exploitation platform for on-line hosted processing. Provides an interactive interface through which the user is able to conduct their analysis of the data, producing *Analysis Results* as output. Interactive Applications include at least the following types: console application, web application (rich browser interface), remote desktop to a hosted VM.
Interactive Console Application	A simple *Interactive Application* for analysis in which a console interface to a platform-hosted terminal is provided to the user. The console interface can be provided through the user's browser session or through a remote SSH connection.
Interactive Remote Desktop	An Interactive Application for analysis provided as a remote desktop session to an OS-session (or directly to a 'native' application) on the exploitation platform. The user will have access to a number of applications within the hosted OS. The remote desktop session is provided through the user's web browser.
Interactive Web Application	An Interactive Application for analysis provided as a rich user interface through the user's web browser.
Key-Value Pair	A key-value pair (KVP) is an abstract data type that includes a group of key identifiers and a set of associated values. Key-value pairs are frequently used in lookup tables, hash tables and configuration files.
Kubernetes (K8s)	Container orchestration system for automating application deployment, scaling and management.

Term	Meaning
Login Service	An encapsulation of Authenticated Login provision within the Exploitation Platform context. The Login Service is an OpenID Connect Provider that is used purely for authentication. It acts as a Relying Party in flows with external IdPs to obtain access to the user's identity.
EO Network of Resources	The coordinated collection of European EO resources (platforms, data sources, etc.).
Object Store	A computer data storage architecture that manages data as objects. Each object typically includes the data itself, a variable amount of metadata, and a globally unique identifier.
On-demand Processing Service	A *Processing Service* whose execution is initiated directly by the user on an ad-hoc basis.
Platform (EP)	An on-line collection of products, services and tools for exploitation of EO data
Platform Tier	The Platform Tier represents the Exploitation Platform and the services it offers to end-users
Processing	A set of pre-defined activities that interact to achieve a result. For the exploitation platform, comprises on-line processing to derive data products from input data, conducted by a hosted processing service execution.
Processing Result	The *Products* produced as output of a *Processing Service* execution.
Processing Service	A non-interactive data processing that has a well-defined set of input data types, input parameterisation, producing *Processing Results* with a well-defined output data type.
Products	EO data (commercial and non-commercial) and Value-added products and made available through the EP. *It is assumed that the Hosting Environment for the EP makes available an existing supply of EO Data*
Resource	A entity, such as a Product, Processing Service or Interactive Application, which is of interest to a user, is indexed in a catalogue and can be returned as a single meaningful search result
Resource Tier	The Resource Tier represents the hosting infrastructure and provides the EO data, storage and compute upon which the exploitation platform is deployed
Reusable Research Object	An encapsulation of some research/analysis that describes all aspects required to reproduce the analysis, including data used, processing performed etc.
Scientific Researcher	Expert user with the objective to perform scientific research. Having minimal IT knowledge with no desire to acquire it, they want the effort for the translation of their algorithm into a service/product to be minimised by the platform.

Term	Meaning
Service Developer	Expert user with the objective to provide a performing, stable and reliable service/product. Having deeper IT knowledge or a willingness to acquire it, they require deeper access to the platform IT functionalities for optimisation of their algorithm.
Software	The compilation of code into a binary program to be executed within the platform on-line computing environment.
Systematic Processing Service	A *Processing Service* whose execution is initiated automatically (on behalf of a user), either according to a schedule (routine) or triggered by an event (e.g. arrival of new data).
Terms & Conditions (T&Cs)	The obligations that the user agrees to abide by in regard of usage of products/services of the platform. T&Cs are set by the provider of each product/service.
Transactional Web Processing Service (WPS-T)	Transactional extension to WPS that allows adhoc deployment / undeployment of user-provided processors.
User	An individual using the EP, of any type (Admin/Consumer/Expert/Guest)
Value-added products	Products generated from processing services of the EP (or external processing) and made available through the EP. This includes products uploaded to the EP by users and published for collaborative consumption
Visualisation	To obtain a visual representation of any data/products held within the platform - presented to the user within their web browser session.
Web Coverage Service (WCS)	OGC standard that provides an open specification for sharing raster datasets on the web.
Web Coverage Processing Service (WCPS)	OGC standard that defines a protocol-independent language for the extraction, processing, and analysis of multi-dimentional coverages representing sensor, image, or statistics data.
Web Feature Service (WFS)	OGC standard that makes geographic feature data (vector geospatial datasets) available on the web.
Web Map Service (WMS)	OGC standard that provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases.
Web Map Tile Service (WMTS)	OGC standard that provides a simple HTTP interface for requesting map tiles of spatially referenced data using the images with predefined content, extent, and resolution.
Web Processing Services (WPS)	OGC standard that defines how a client can request the execution of a process, and how the output from the process is handled.
Workspace	A user-scoped 'container' in the EP, in which each user maintains their own links to resources (products and services) that have been collected by a user during their usage of the EP. The workspace acts as the hub for a user’s exploitation activities within the EP

1.5. Glossary

The following acronyms and abbreviations have been used in this report.

Term	Definition
AAI	Authentication & Authorization Infrastructure
ABAC	Attribute Based Access Control
ADES	Application Deployment and Execution Service
ALFA	Abbreviated Language For Authorization
AOI	Area of Interest
API	Application Programming Interface
CMS	Content Management System
CWL	Common Workflow Language
DAL	Data Access Library
EMS	Execution Management Service
EO	Earth Observation
EP	Exploitation Platform
FUSE	Filesystem in Userspace
GeoXACML	Geo-specific extension to the XACML Policy Language
IAM	Identity and Access Management
IdP	Identity Provider
JSON	JavaScript Object Notation
K8s	Kubernetes
KVP	Key-value Pair
M2M	Machine-to-machine
OGC	Open Geospatial Consortium
PDE	Processor Development Environment
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIP	Policy Information Point
RBAC	Role Based Access Control
REST	Representational State Transfer
SSH	Secure Shell
TOI	Time of Interest
UMA	User-Managed Access

Term	Definition
VNC	Virtual Network Computing
WCS	Web Coverage Service
WCPS	Web Coverage Processing Service
WFS	Web Feature Service
WMS	Web Map Service
WMTS	Web Map Tile Service
WPS	Web Processing Service
WPS-T	Transactional Web Processing Service
XACML	eXtensible Access Control Markup Language

Chapter 2. Overview

2.1. Building Block Overview

Content Description

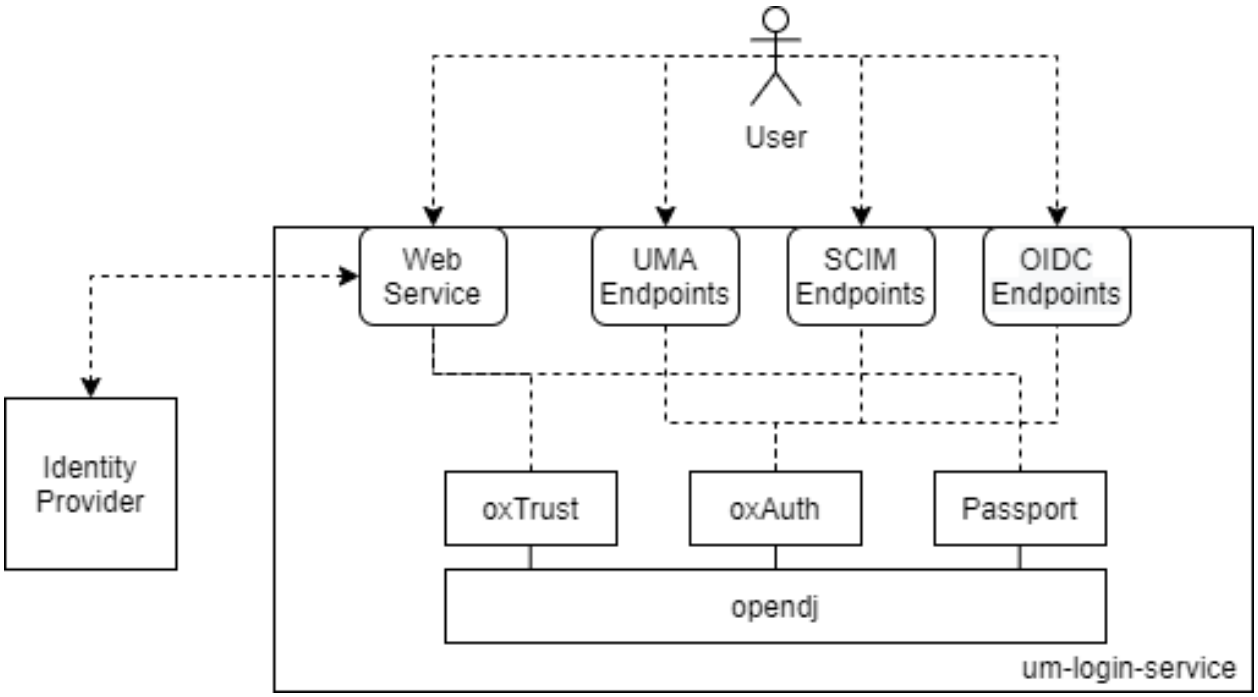
This section contains:

- High-Level Description of the Building Block
- Context within EOEPCA

The Login Service building block provides an OIDC and UMA compliant solution enabling authentication and authorization mechanisms within an Exploitation Platform. Other building blocks, such as the Policy Decision Point, Policy Enforcement Point and User Profile rely on this Building Block to provide several standard interfaces.

As stated in the EOEPCA Master System Design Document (ref. [PENDING]), the Login Service establishes itself as a client (Relying Party) for external Identity Providers (IDPs) allowing users coming from external domains to authenticate into the Exploitation Platform. In addition to this functionality, the Login Service also allows any Platform component to authenticate itself (without the need for an End-User to initiate the flow) therefore facilitating security aspects of the internal workflows of a Platform.

The figure below, identifies the main workflows on which the Login Service participates:



Based on this workflow, the Login Service enables the following functionality aspects:

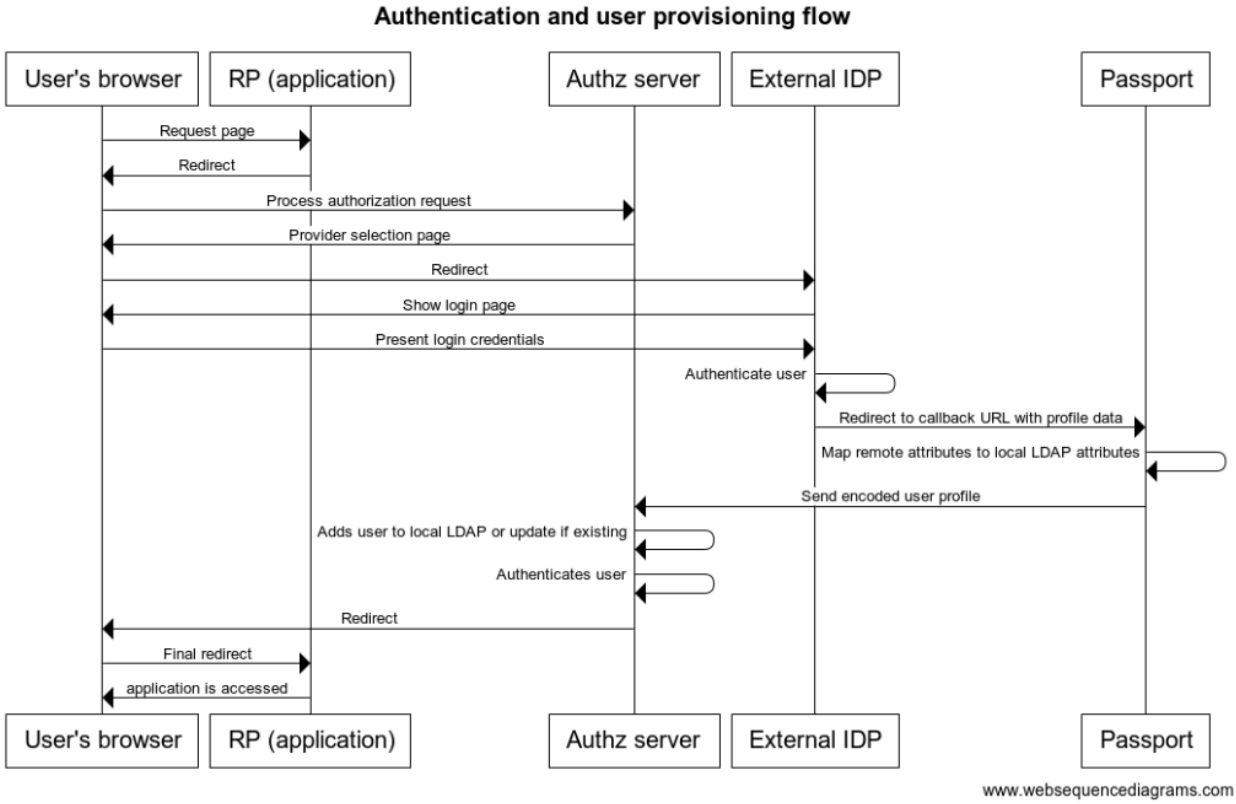
2.1.1. End-User Authentication through external IDPs

The Login Service establishes itself as a client to all supported external IDPs, with the necessary trust relationships in place to support the authentication flow. End-Users participating in this flow

will propagate - with their necessary consent - a subset of the information stored in the external IDP into the Login Service back-end in order to facilitate unique identification of users and the Billing and Accounting functionalities of the overall Platform.

2.1.1.1. Authentication Flow

The following image indicates the sequence of events necessary for a user to authenticate using an external Identity Provider (IDP).



2.1.1.2. Unique identification of End-Users

By default, the Login Service extracts the email account stored in the external Identity Provider and checks that the user has not logged in previously with any other external IDP under the same email account. Whenever that's the case, the authentication flow skips the persistence of End-User attributes to the back-end and directly considers the user authenticated.

2.1.2. Single Sign-On

Single Sign-On (SSO) comes as one of the main benefits of using OIDC as authentication standard, allowing Service Providers (Client Applications) that utilize the Login Service to skip the step that requires input of credentials by the end-user if the Login Service detects that a session for the user is ongoing within the platform.

2.1.3. Single Log-Out

Similar to the SSO functionality, applications can request the End-User session to be terminated across the platform, and register themselves to receive this log-out request (in order to also handle it internally).

2.1.4. Authorization Scopes

Authorization scopes are definitions used to protect resources, by enabling associated policies that ensure a connecting client has the access permissions required for a given resource.

2.1.4.1. Public Access

This specific scope allows for any end user or client application to freely access a resource, without any restrictions.

2.1.4.2. Authenticated Access

This scope is associated with a policy that verifies if the user, making the request, has an ongoing active session.

2.1.4.3. Protected Access

This scope delegates user verification to the Policy Decision Point (PDP) building block, via XACML-compliant communication.
The protected access will verify the JWT's credentials if is a external issuer to the platform and will be indicated via XACML to the PDP.

2.1.4.4. Protected Read

This scope also delegates user verification to the Policy Decision Point (PDP) building block, via XACML-compliant communication.
The protected read will verify the HTTP verb (HEAD and GET) used to the request to the PEP endpoint and then will be indicated via XACML to the PDP with a value of 'read' in the action-id.

2.1.4.5. Protected Write

This scope has the same functionality like the previous scope and it is used in Policy Decision Point (PDP) building block, via XACML-compliant communication.
In this case the protected write will verify the HTTP verb (PUT, POST and DELETE) used to the request to the PEP endpoint and then will be indicated via XACML to the PDP with a value of 'write' in the action-id.

2.1.4.6. Is Operator

This scope is used in conjuction with the isOperator custom user attribute, to allow the Policy Enforcement Point (PEP) to query the OIDC endpoint and establish if a user is an authorized operator for resources.

2.1.5. Custom UMA RPT policies

Defining custom UMA RPT Policies allows for fine-tuning of scope-dependent actions that were defined above, such as introspection and a first step validation of JWT tokens and other attributes.

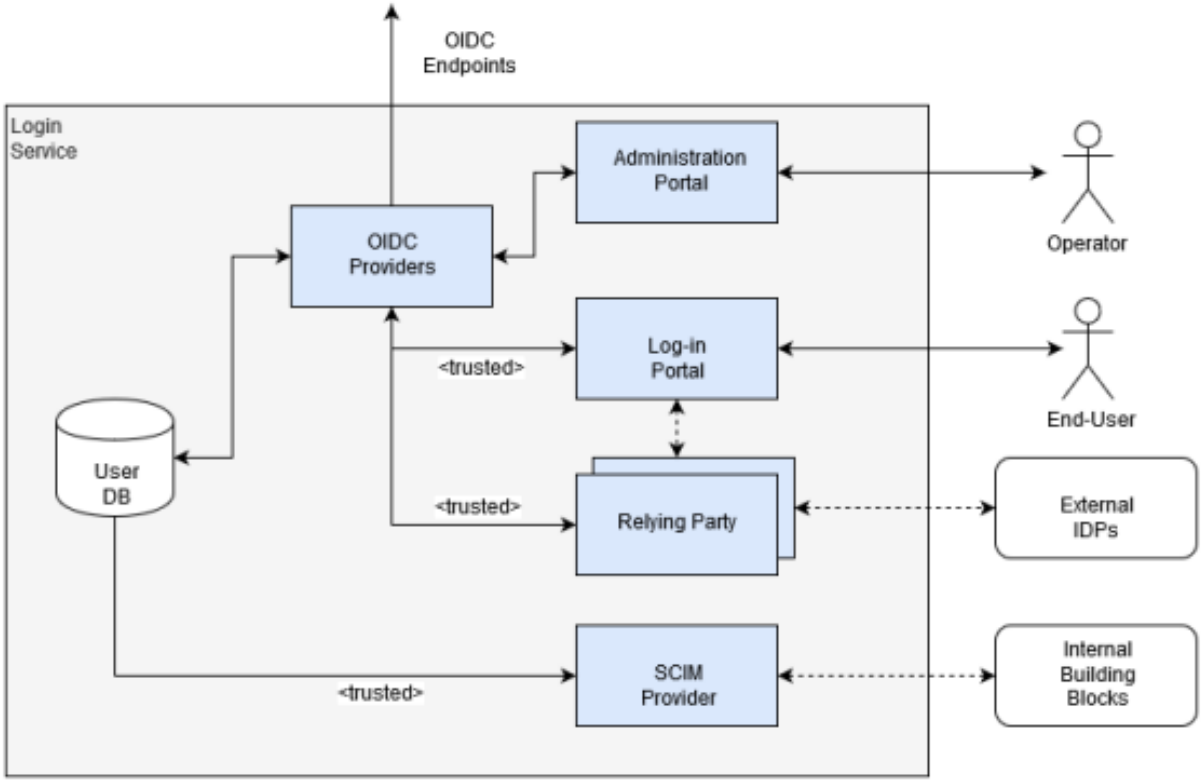
2.2. Static Architecture

Content Description

This section contains:

- Diagram and description of the major logical components within the Building Block

The Login Service heavily relies on Free and Open Source Software that already implements the interfaces and functionality required. The following diagram describes the main components of this building block:



- The Authentication and Authorization Service enables both OIDC and UMA flows for the whole Platform
- The Administration Service provides and Administration GUI and a SCIM Implementation allowing direct interaction with the End-User Back-End
- The Back-End Service stores the necessary information for the other components to provide Session Management, Authorization and persists external IDP information generated or relayed during authentication.
- The Relying Party component allows to register as a RP in any external IDP.

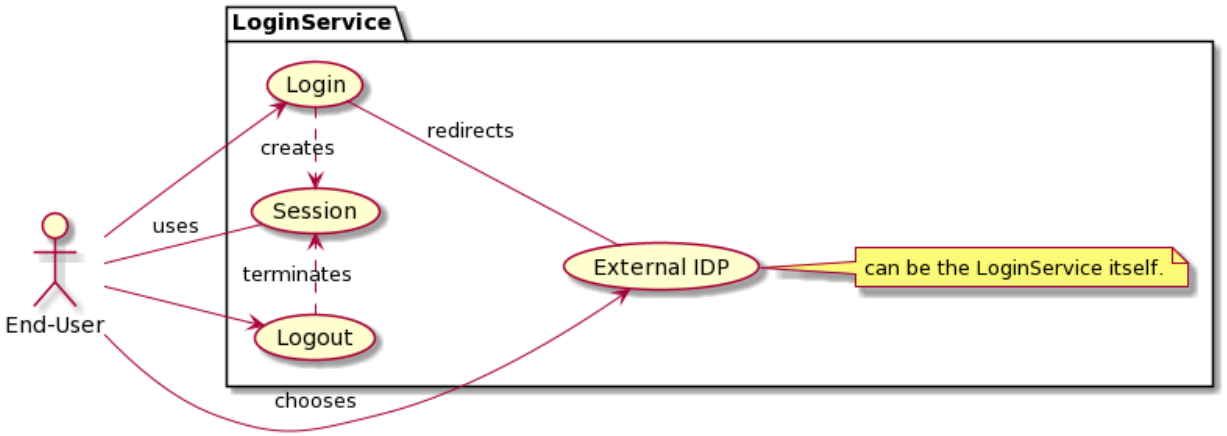
The Section for the Building Block Design [Building Block Design] contains detailed descriptions and references to the Open Source components used in this Building Block.

2.3. Use Cases

Content Description

This section contains:

- Diagrams and definition of the use cases covered by this Building Block



This diagram covers the following use cases:

2.3.1. LS-UC-001: Single Sign-on

Upon performing a Login action, a new Session is generated that is then used by the user to perform actions on the Exploitation Platform.

2.3.2. LS-UC-002: Single Log-out

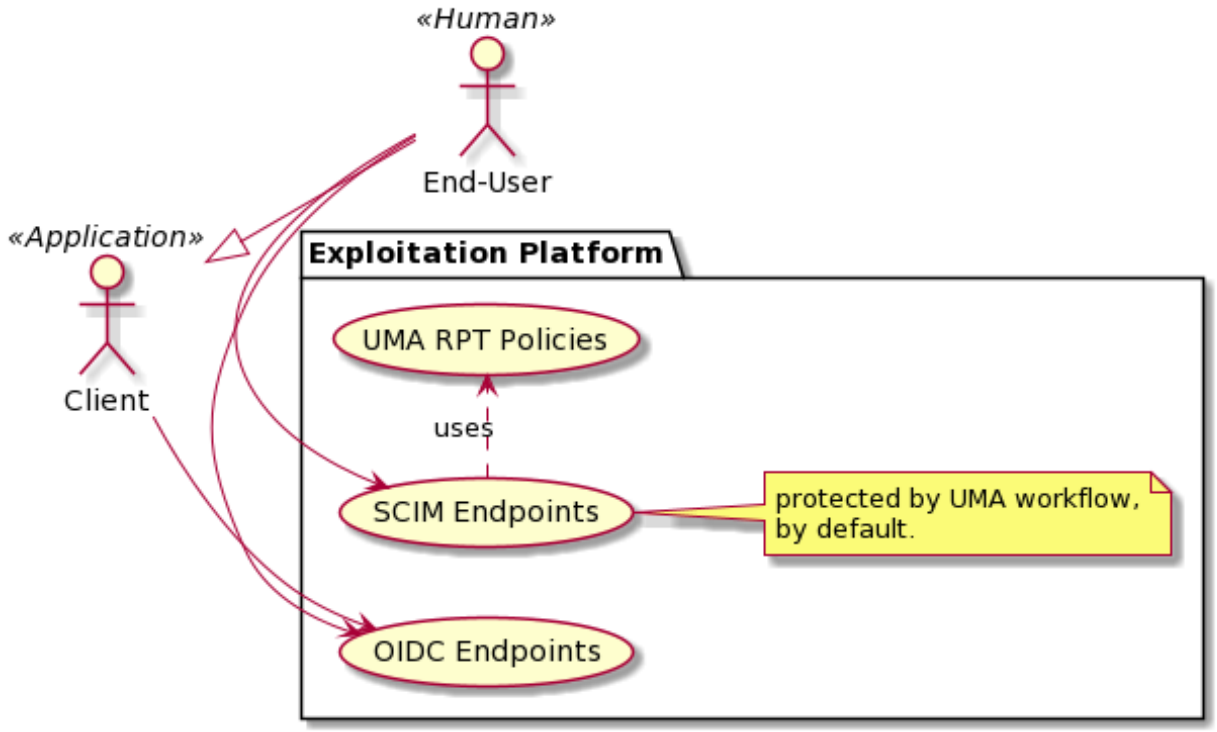
Upon performing a Logout action, the created Session is destroyed, preventing further action by the End-User on the Exploitation Platform, until a new Login action is taken.

2.3.3. LS-UC-003: End-User chooses Identity Provider

When performing a Login action, the End-User can choose if they wish to use an External Identity Provider to delegate authentication (or use the Login Service itself).

2.3.4. LS-UC-004: External Identity Provider Authentication

If an External Identity Provider is chosen to delegate authentication, the Login Service redirects to it in order to proceed with the Login action. This is best explained by the sequence diagram present in [Authentication Flow]



This diagram covers the following use cases:

2.3.5. LS-UC-004: UMA workflow

By default, some actions are protected by a UMA (User-Managed access) workflow, that is already described in the Master System Design document (ref. [PENDING]). As part of this process, any security policies that have been previously established as [[[_authentication_scopes]]] are enacted.

2.3.6. LS-UC-005: Request User Attributes through SCIM

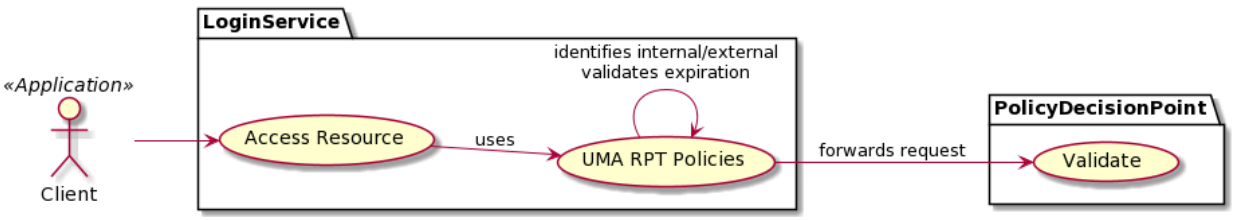
Requesting a User's attributes can be done via SCIM's .well-known endpoints, best by using a previously registered client to call them and perform UMA authentication.

2.3.7. LS-UC-006: Resource registration

Resource registration can be done directly via OIDC's .well-known endpoints, best by using a previously registered client to call them using basic authentication.

2.3.8. LS-UC-007: Dynamically register a client

The Exploitation Platform allows a client to dynamically register with itself, by calling OIDC .well-know endpoints.



This diagram covers the following use cases:

2.3.9. LS-UC-008: JWT expiration validation

When accessing a resource, the Login Service is capable, on a first step, of validating the expiration date on a JWT to avoid passing expired authentication attempts to the Policy Decision Point.

2.3.10. LS-UC-009: External JWT identification

The Login Service is capable of identifying whether a passed JWT is internal or external do the Exploitation Platform. In case of external JWTs, the Login Service is also capable of passing any external attributes contained in the request, including the issuer, to the Policy Decision Point.

2.4. External Interfaces

Content Description

This section contains:

- Listing of technical external interfaces (with other Building Blocks)

2.4.1. Authentication (AuthN) Interface

The Login Service exposes an OpenID Connect interface through a .well-known URI that describes all standard endpoints.

2.4.2. Authorization (AuthZ) Interface

The Login Service exposes a User Managed Access (UMA) interface through a .well-known URI that describes all standard endpoints.

2.4.3. Identity Management Interface

The Login Service exposes a System for Cross Domain Identity (SCIM) Interface through a .well-known URI that describes all standard endpoints.

2.4.4. Administration Interface

A web service is made available for administrators and operators to manage the configuration aspects of the Login Service without the need to authenticate using external IDPs.

2.4.5. External IDP Discovery Interface

A landing web page interface for Authentication Requests is made available, allowing users to select their preferred external IDP and initiate authentication flow.

2.5. Required Resources

Content Description

This section contains:

- List of HW and SW required resources for the correct functioning of the building Block
- References to open repositories (when applicable)

2.5.1. Software

The following Open-Source Software is required to support the deployment and integration of the Login Service:

- Authentication and Authorization Service
  - oxAuth - Gluu Inc. - https://github.com/GluuFederation/oxAuth
- Administration Service
  - oxTrust - Gluu Inc. - https://github.com/GluuFederation/oxTrust
- Back-end Service
  - OpenDJ/LDAP Distribution - https://github.com/GluuFederation/docker-opendj
- OIDC Compliant, extensible Relying Party
  - Passport.js - https://github.com/jaredhanson/passport
- Deployment, Configuration and Integration Tooling
  - Persistence system load/backup/restore components - https://github.com/EOEPCA/um-login-persistence
  - Kubernetes secret and config Tooling - https://github.com/GluuFederation/gluu-docker/tree/3.1.6/examples/kubernetes/minikube
  - Reverse Proxy exposing API interfaces - Nginx/Ingress

Chapter 3. Building Block Design

Content Description

This section contains:

- A concise breakdown of the Building Block in several independent services (when applicable). For each component, the following subsections are added:
  - Overview and purpose: indicating the functionality covered by the component
  - SW Reuse and Dependencies: indicating reuse of third party open source solutions (if any) and any pre-required Dependencies
  - Interfaces: both internal to the building block and those exposed externally
  - Data: Data usage of the building block, data flow and any GDPR concerns should be addressed here
  - Applicable Resources: links and references to (Reference Docs), and repositories.

When a breakdown is necessary, a general overview of the building block can be given. On the contrary, no breakdown indicates a single component development with the same expected sections.

3.1. Authentication and Authorization Service

3.1.1. Overview and Purpose

The login service provides both authentication and authorization via it's oxAuth component. This component also provides the OpenID Connect (OIDC) and UMA flows, exposed through endpoints to external clients.

It also provides authentication via web service, allowing for simple login interactions with users.

3.1.2. Software Reuse and Dependencies

This service depends solely on oxAuth software, from gluu (see applicable resources and the overview for this building block)

3.1.3. Interfaces

3.1.3.1. OIDC Endpoints

A listing of endpoints to interact with OIDC is exposed on a .well-know URL (default is .well-known/openid-configuration).

3.1.3.2. UMA Endpoints

A listing of endpoints to interact with UMA is exposed on a .well-know URL (default is .well-known/uma2-configuration).

3.1.4. Data

3.1.4.1. Configuration

Configuration is obtained via the persistance module, which loads scripts, custom html, and data to LDAP and/or their respective places.

Direct customization without the "persistance" module is not possible at this time.

3.1.4.2. Data flow

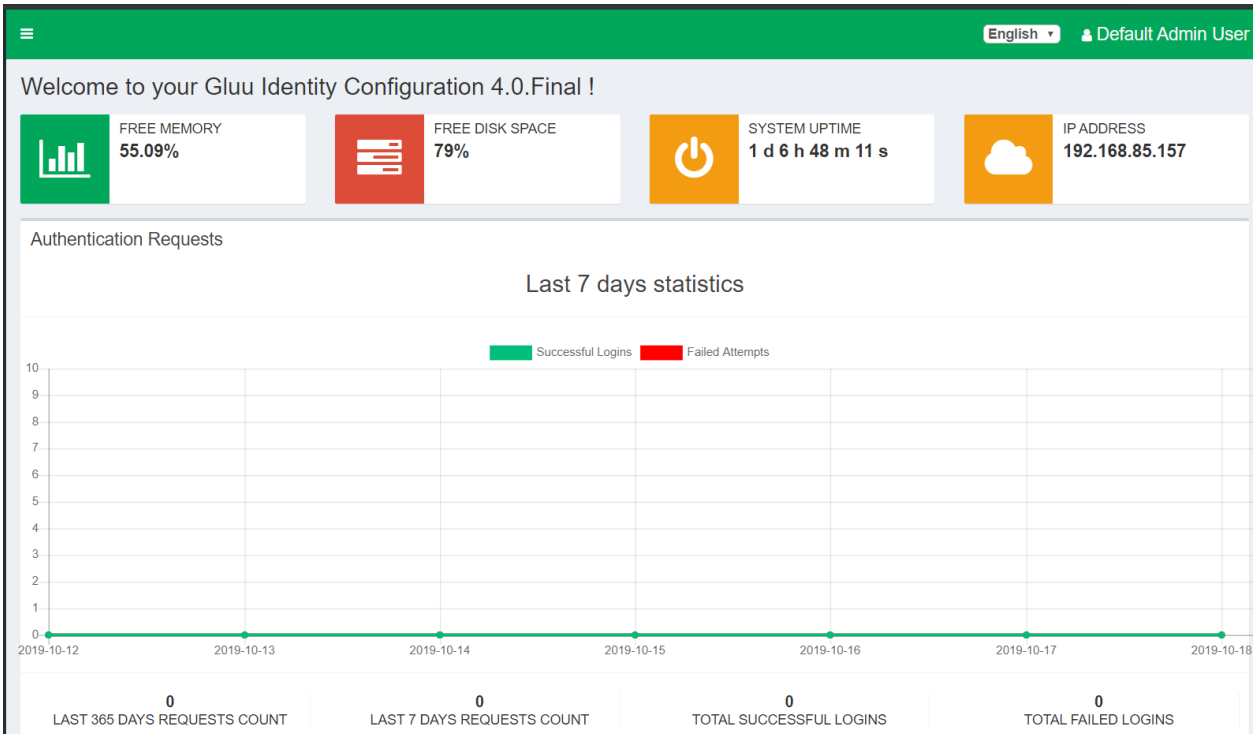
All information handled by this service is read/written to the LDAP instance deployed at the backend.

3.1.5. Applicable Resources

- oxAuth (gluu) - https://gluu.org/docs/de/reference/oxauth/
- oxAuth (gluu @ Github) - https://github.com/GluuFederation/docker-oxauth

3.2. Administration Service

3.2.1. Overview and Purpose



The login service provides administration functionality for users with elevated priviledges. This allows admins to have an overview of the entire system, as well as configure anything necessary

from this interface. Some of the configurations available from it are:

- OAuth Clients
- UMA Scopes & Resources
- Edit/Add/Remove Users

A complete description of available configurations can be found here: https://gluu.org/docs/gluu-server/4.1/admin-guide/oxtrust-ui/#oxtrust-admin-gui

3.2.2. Software Reuse and Dependencies

- **oxTrust**: A gluu component, providing the web interface and interaction
- **opendj**: The instance of LDAP providing the backend store for information

3.2.3. Interfaces

3.2.3.1. SCIM Endpoints

A listing of endpoints to interact with SCIM is exposed on a .well-know URL (default is .well-known/scim-configuration).

3.2.3.2. Web Interface

A web interface is exposed to allow administrators and local operators to login and manage configuration aspects of the Login Service.

3.2.4. Data

3.2.4.1. Configuration

Configuration is obtained via the persistance module, which loads scripts, custom html, and data to LDAP and/or their respective places.

Direct customization without the "persistance" module is not possible at this time.

3.2.4.2. Data flow

All information handled by this service is read/written to the LDAP instance deployed at the backend.

3.2.5. Applicable Resources

- oxTrust (gluu) - https://gluu.org/docs/de/reference/oxtrust/
- oxTrust (gluu @ Github) - https://github.com/GluuFederation/docker-oxtrust

3.3. Back-End Service

3.3.1. Overview and Purpose

The backend service is composed of the opendj component, which serves an LDAP service, storing all of the data and configuration of the platform.

3.3.2. Software Reuse and Dependencies

- **opendj**: The instance of LDAP providing the backend store for information

3.3.3. Interfaces

Not applicable, all actions are done internally on the LDAP instance.

3.3.4. Data

3.3.4.1. Data Flow

The service itself doesn't process data, but stores it in an LDAP database.
Thus, all flows follow the LDAP protocol for connection and exchange of data.

3.3.4.2. Configuration

There is no configuration available for this service at this time

3.3.5. Applicable Resources

- Persistance, LDAP - Gluu - https://gluu.org/docs/gluu-server/reference/persistence/#ldap
- OpenDJ, docker - https://github.com/GluuFederation/docker-opendj

3.4. Relying-Party Service

3.4.1. Overview and Purpose

Gluu has a bundled middleware project called Passport.js, the purpose of which is to facilitate authentication using an external identity provider. To this end, the OIDC interface can be called by an IDP.

3.4.2. Software Reuse and Dependencies

- **oxTrust**: A gluu component, providing the web interface and interaction
- **passport**: Gluu's container instance of Passport.js' implementation

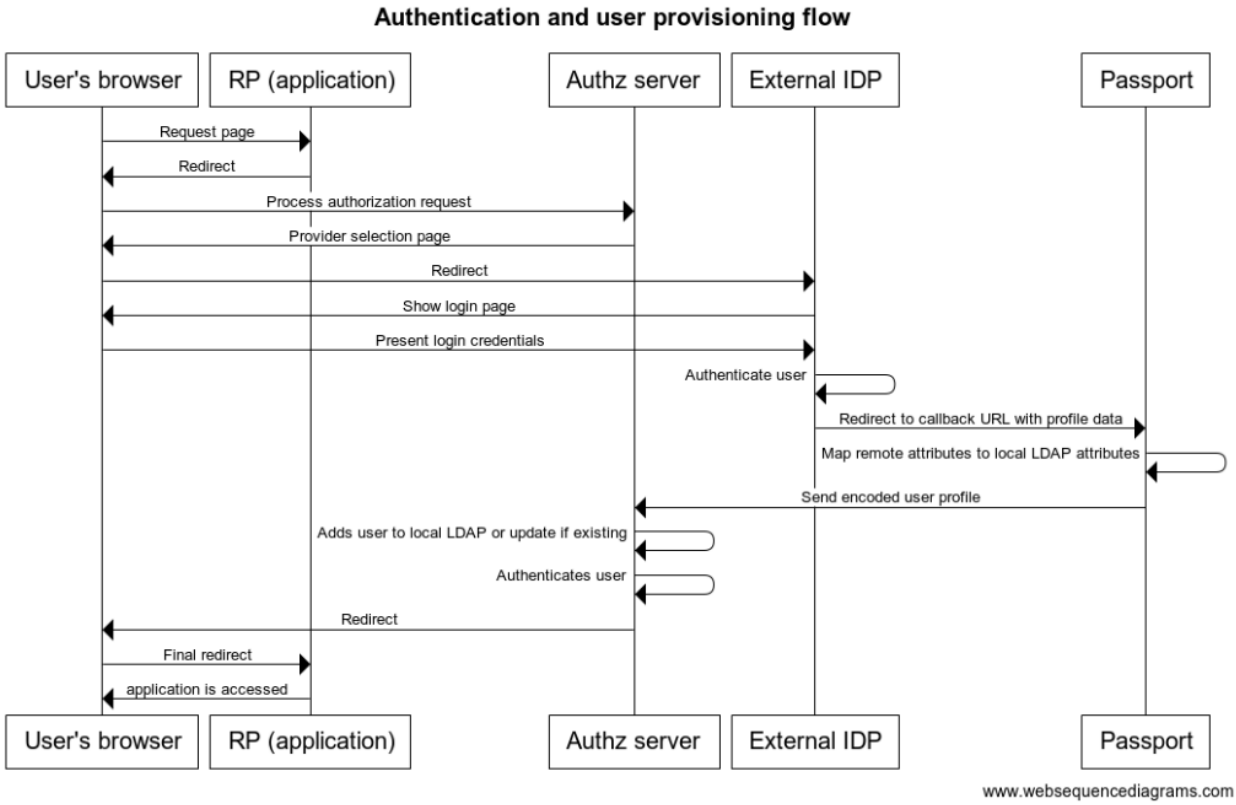
3.4.3. Interfaces

3.4.3.1. OIDC Endpoints

A listing of endpoints to interact with OIDC is exposed on a .well-know URL (default is .well-known/openid-configuration).

3.4.4. Data

3.4.4.1. Data Flow



3.4.4.2. Configuration

Configuration is obtained via the persistance module, which loads scripts, custom html, and data to LDAP and/or their respective places.

Direct customization without the "persistance" module is not possible at this time.

3.4.5. Applicable Resources

- Passport.js - http://www.passportjs.org/
- Passport (Github) - https://github.com/jaredhanson/passport
- oxPassport (gluu) - https://gluu.org/docs/de/reference/oxpassport/
- oxPassport (gluu @ Github) - https://github.com/GluuFederation/docker-oxPassport

3.5. Logging

3.5.1. Design

Logging accross the EOEPCA Building Blocks works much in the same way, by usage of a log helper class to initiate a Python logger, handler and formater that simultaneously outputs log messages to console and a log file. These log files are set on a rotation, with a 1GB limit per each, with the 10 latest log files being kept in memory.

A new configuration yaml file is added to the building block, containing initialization parameters.

3.5.2. Log message format

INFO level log messages follow the following format:

- TIME: in ISO 8601 format, "%Y-%m-%dT%H:%M:%S%z"
- LEVELNAME: INFO by default
- COMPONENT: "LGS"
- SUBCOMPONENT: OxTrust or OxAuth
- ACTION IDENTIFIER: N/A
- ACTION TYPE: AUTHORIZATION or AUTHENTICATION
- LOG CODE: Unique code identifying log message type
- ACTIVITY: Detailed log message, check reference table

3.5.3. Log message codes

Subcomponent division is as follows:

- 00xx: OxTrust
- 01xx: OxAuth

Table 2. Log Codes

Log Code	Structure
0001	{"User":user,"Description":"Invalid JWT signature","JWT":jwt}
0002	{"User":user,"Description":"Expired JWT","JWT":jwt}
0003	{"User":user,"Description":"User authenticated","JWT":jwt}
0004	{"User":user,"Description":"User authentication failed","JWT":jwt}
0005	{"User":user,"Description":"No provider, basic auth completed"}
0006	{"User":user,"Description":"No provider, basic auth failed"}
0007	{"User":user,"Description":"Pre-registered provider, authenticating"}
0008	{"User":user,"Description":"Mail login, email value missing, authentication failed"}

Log Code	Structure
0009	{"User":user,"Description":"Mail login, authentication completed"}
0010	{"User":user,"Description":"Mail login, authentication failed"}

Chapter 4. User Story Traceability

Table 3. User Stories

User Story Code	Description	Building Block Use Case	Master Use Case
EOEPCA-188	Implementation of Helm Charts	-	CI/CD task
EOEPCA-66	Setup of CI/CD Configuration for "Login Service"	-	CI/CD task
EOEPCA-22	End-User Request for consent on attribute release	-	EOEPCA-UC-0103/EOEPCA-UC-1707
EOEPCA-19	Uniquely Identification of End-Users	LS-UC-001	EOEPCA-UC-0101
EOEPCA-18	End-User Logout	LS-UC-002	-
EOEPCA-17	End-User Authentication using external IDPs (Github)	LS-UC-001	EOEPCA-UC-0102
EOEPCA-16	Local Access to Administration Portal	LS-UC-001/LS-UC-002/LS-UC-003/LS-UC-004	EOEPCA-UC-0109/EOEPCA-UC-0111
EOEPCA-10	Setup and tailoring of OIDC and UMA provider	LS-UC-004/LS-UC-005/LS-UC-006/LS-UC-007	EOEPCA-UC-0507/EOEPCA-UC-1003/EOEPCA-UC-1004/EOEPCA-UC-1005
EOEPCA-9	Initial setup of CI/CD and Development Tooling	-	CI/CD task
EOEPCA-176	Policy Checks over external End-Users	-	EOEPCA-UC-0107
EOEPCA-174	Automation of Security Concerns	-	-
EOEPCA-13	Reusable SCIM Client Implementation	LS-UC-006/LS-UC-007	EOEPCA-UC-0507/EOEPCA-UC-1003/EOEPCA-UC-1004/EOEPCA-UC-1005
EOEPCA-11	Reusable OIDC Client Implementation	LS-UC-006/LS-UC-007	EOEPCA-UC-0507/EOEPCA-UC-1003/EOEPCA-UC-1004/EOEPCA-UC-1005

<< End of Document >>