hp Unified OSS Console

Unified OSS Console
V2.1

Installation Guide

Edition: 1.0

For Linux (RHEL 6.5)

June 2015

Legal Notices

Warranty

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

License Requirement and U.S. Government Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

Trademark Notices

Adobe®, Acrobat® and PostScript® are trademarks of Adobe Systems Incorporated.

Java™ is a trademark of Oracle and/or its affiliates.

Microsoft®, Internet Explorer, Windows®, Windows Server®, and Windows NT® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Firefox® is a registered trademark of the Mozilla Foundation.

Google Chrome® is a trademark of Google Inc.

Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of The Open Group.

X/Open® is a registered trademark, and the X device is a trademark of X/Open Company Ltd. in the UK and other countries.

Red Hat® is a registered trademark of the Red Hat Company.

Linux® is a registered trademark of Linus Torvalds in the U.S. and other countries.

Apache CouchDB, CouchDB, and the project logo are trademarks of The Apache Software Foundation

Node.js project. Joyent® and Joyent's logo are registered trademarks of Joyent, Inc

Contents

Preface 9

Chapter 1 11

Code Signing 11

Chapter 2 12

Introduction 12

Chapter 3 14

Pre-requisites 14

3.1 Overview 14
3.2 UOC Server Hardware Requirements 15
3.3 Supported Web Browsers 15
3.4 Client PC / Laptop Hardware Requirements 15
3.5 Mobile Device Hardware Requirements 16
3.6 Operating System 16
3.7 Users 16
3.8 Node.js 17

Chapter 4 18

Deployment Examples 18

4.1 Monolithic 18
4.2 Multiple instances 19
4.3 Multiple servers 20

Chapter 5 21

Installation 21

5.1 Installation locations 21
5.2 Installation media 21
5.3 Interactive script 21
5.4 User profile 22
5.5 Setup 22
5.6 OSSA add-on configuration 23
5.7 Server startup 23
5.8 Red Hat Linux firewall settings 23

Chapter 6 24

Installation validation 24

Chapter 7 ..... **26**

Uninstallation **26**

Chapter 8 ..... **27**

Administration **27**

8.1 Start UOC Server 27
8.2 Check running UOC Server 27
8.3 Stop UOC Server 27
8.4 UOC Inventory 28
8.5 Start CouchDB Server 28
8.6 Stop CouchDB Server 28

Chapter 9 ..... **29**

Platform Configuration **29**

9.1 UOC Server 29
9.2 UOC Data Import 29

Chapter 10 ..... **33**

Platform Backup **33**

10.1 Apache CouchDB 33
10.1.1 Backup CouchDB database files 33
10.1.2 Restore CouchDB database files 34
10.1.3 Replicate CouchDB 34
10.2 UOC Data 35

Chapter 11 ..... **36**

Security Guide **36**

11.1 Terminology 36
11.2 Overview 37
11.3 Authentication 38
11.3.1 Enable Local Authentication 39
11.3.2 Enable SSO / SAML Configuration 39
11.3.3 Configure JSON Web Token 40
11.3.4 Generate JSON Web Token 41
11.4 User and Password Management 43
11.5 Role Based Access Control (RBAC) 43
11.5.1 Roles 44
11.5.2 Permissions 46
11.6 CouchDB Database 48
11.6.1 Built-in Administration 49
11.6.2 Configuration 49
11.7 Secure Socket Layer (SSL) 50

11.7.1 SSL Certificates....50
11.7.2 Overview....51
11.7.3 GUI Database (Couchdb server)....52
11.7.4 UOC ←→ GUI Database....52
11.7.5 UOC ←→ Web browser....53
11.7.6 UOC ←→ Identity Provider (IdP) / SSO....54
11.7.7 UOC ←→Domain or data server (ex: OSS Analytics server)....55
11.8 Logging....56
11.9 Auditing....56
11.9.1 Sessions....57
11.9.2 Resource access....58

Chapter 12....60

Troubleshooting....60
12.1 Logging....60
12.1.1 Server logs....60
12.1.2 Http requests....61
12.2 Web Browser Console....61

Figures

Figure 1 – UOC V2.1 pre-requisites 14
Figure 2: *Deployment Examples – Monolithic* 18
Figure 3: *Deployment Examples – Multiples UOC Instances* 19
Figure 4: *Deployment Examples – Multiples Servers* 20
Figure 5 – Data Import Overview 30
Figure 6: CouchDB Futon Replication Interface 35
Figure 7: UOC Server - Security Overview 37
Figure 8 – UOC Authentication modes (SAML / Local) 38
Figure 9 – Example of generation of JSON Web Token (http://jwt.io) 42
Figure 10 – RBAC - ANSI INCITS 359-2004 43
Figure 11 – Example of operator level 1 dashboard 46
Figure 12 – Support Secure Socket Layer (SSL) 51

Tables

Table 1 - Software versions 9
Table 2 –Hardware requirements for UOC V2.1 on Linux 15
Table 3 –Supported Web browsers 15
Table 4 –Hardware requirements for client PC 16
Table 5 –Hardware requirements for mobile devices 16
Table 6 – RBAC – Default Roles / Permissions 45
Table 7 – RBAC –List of user interface permissions 48

Preface

This guide describes how to install the product on the various supported platforms.

Product Name: **Unified OSS Console**

Product Version: V2.1

Kit Version: V2.1.0

Intended Audience

Here are some recommendations based on possible reader profiles:

- Administrators
- Integrators

Software Versions

The term UNIX is used as a generic reference to the operating system, unless otherwise specified.

Product Version	Supported Operating systems
Unified OSS Console V2.1	Red Hat Enterprise Linux Server release 6.5
Unified OSS Console V2.1 - Add-ons OSS Analytics V1.1	Red Hat Enterprise Linux Server release 6.5

Table 1 - Software versions

Typographical Conventions

`Courier` Font:

- Source code and examples of file contents.
- Commands that you enter on the screen.
- Pathnames
- Keyboard key names

Italic Text:

- Filenames, programs and parameters.
- The names of other documents referenced in this manual.

Bold Text:

- To introduce new terms and to emphasize important words.

Associated Documents

The following documents contain useful reference information:

- Unified OSS Console V2.1.0 – Release Notes
- Unified OSS Console V2.1.0 – User Guide

Support

Please visit our HP Software Support Online Web site at
https://softwaresupport.hp.com/

for contact information, and details about HP Software products, services, and support.

The Software support area of the Software Web site includes the following:

- Downloadable documentation.
- Troubleshooting information.
- Patches and updates.
- Problem reporting.
- Training information.
- Support program information.

Chapter 1

Code Signing

This Software Product from HP is digitally signed and accompanied by Gnu Privacy Guard (GnuPG) key.

On Red Hat Enterprise Linux, HP-UX, Windows and Solaris platforms:

Below mentioned procedure* allows you to assess the integrity of the delivered Product before installing it, by verifying the signature of the software packages.

Pick the signature (.sig) file shipped along with the product and use following GPG command

gpg --verify <product.sig> <product>
Example: gpg --verify VPNSVP-X51-3A.zip.sig VPNSVP-X51-3A.zip

Note: Look for the comments shown below in the command output
Good signature from "Hewlett-Packard Company (HP Code signing Service)"

Note: If you are not familiar with signature verification using GPG and intended to verify HP Product signature, follow the steps given below.

1. Check whether gnupg gpg is installed on the system. If no, install gnupg gpg

2. Configure GPG for accepting HP signature. The steps are the following:
    a. Log as root on your system
    b. Get the hpPublicKey from following location: https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=HPLinuxCodeSigning and save it as hpPublicKey.pub
    Note that the hpPublicKey file will be located in the root's home directory.
    c. Follow the instruction found at above URL in the "Verification using GPG" section.

**HP strongly recommends using signature verification on its products, but there is no obligation. Customers will have the choice of running this verification or not as per their IT Policies.*

Chapter 2

Introduction

It is recommended to read the Release Notes document before proceeding with the installation.

The OSS console software kit is targeted for Red Hat Enterprise Linux V6.5 (x86-64) only. The software may run properly on many other Linux distributions but no support is given to them.

The software kit is made of 4 rpm packages:

- Apache CouchDB V1.6.0 (pre-requisite)
- NodeJS V0.10.38 (pre-requisite)
- HP Unified OSS Console V2.1
- HP Unified OSS Console V2.1 – Add-on OSS Analytics V1.1

***Unsupported Products**:*

*Node (http://**nodejs**.org) and CouchDB (http://**couchdb**.apache.org) are pre-requisite not supported by HP. Please go to official web sites for details, documentation, patches and updates.*

The CouchDB database is used to store internal configuration data (like views or workspace definitions), but as there is no "official" RHEL6 rpm package for the version 1.6.0, we ship a pre-built one (not supported by HP), including all necessary dependencies. CouchDB can be installed on a remote server if required.

See http://couchdb.apache.org/ for details. The CouchDB server can be accessed remotely through a web-based administration console or a RESTful API.

***Important**: A specific CouchDB v1.6.1 has been release for RHEL7. Based on your system, check carefully the version you need to install.*

The NodeJS package is provided for convenience (not supported by HP) and can be installed by the OSS console installer on demand. Preferably, you can install it with a standard “yum” update. See http://nodejs.org/ for details.

---

*NodeJS requires a version **0.10.38** to make sure the SSL vulnerability is fixed in UOC Server*

Important: The UOC core package, the OSSA add-on and NodeJS must be installed on the same system.

Chapter 3

Pre-requisites

3.1 Overview

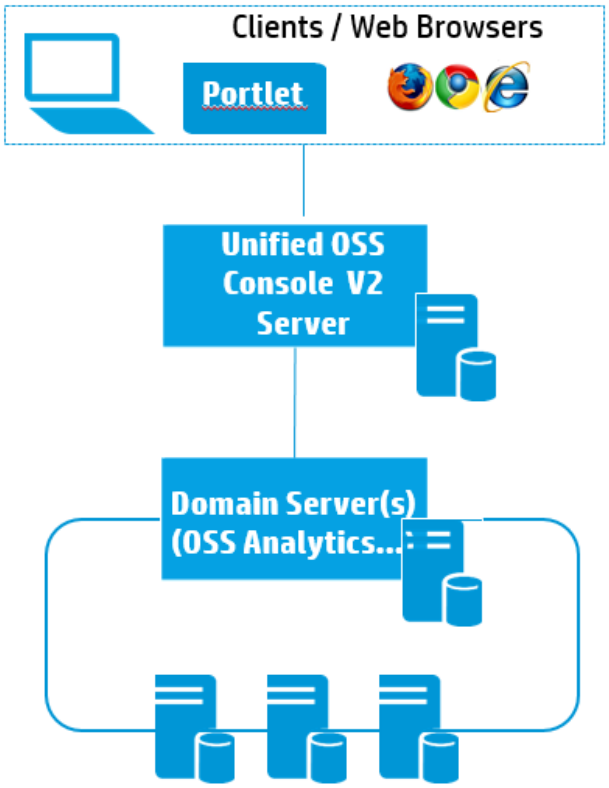

Figure 1 – UOC V2.1 pre-requisites

UOC V2 require an installation on a Linux Server (UOC V2 Server), and web browsers as client to this server. It could be a PC, laptop or mobile device.

Note: The UOC V2 Server supports virtual machine

3.2 UOC Server Hardware Requirements

The table below lists the recommanded hardware requirements for a UOC server installation.

Recommanded hardware is : HP ProLiant BL465c or DL360p Gen8

Appropriate sizing is of course subject to real volume of data, throughput and/or number of concurrent users. For an optimum sizing exercise, please contact the product manager.

Hardware required will also be driven by domain servers associated to the UOC Server.

Hardware	Recommanded	Optimum
CPU	1x Intel® Xeon® E5-2640 2.5GHz/6-core	Needs sizing
RAM	16 GB	Needs sizing
Hard disk Size	100 GB	Needs sizing
Hard Disk Size	1x 10 Gbps Ethernet Ports on board/Dual Port FC HBA	Needs sizing

Table 2 –Hardware requirements for UOC V2.1 on Linux

*This table does not list the hardware requirement to support the **HP OSS Analytics Server**. Please refer to the OSSA installation guide to get this information and install successfully the OSSA Server (if needed).*

3.3 Supported Web Browsers

Only the following web browser are supported:

Web Browser	Version	Web site
Microsoft Internet Explorer	10 or later	http://windows.microsoft.com/en-us/internet-explorer/download-ie
Mozilla Firefox	V32 or later	https://www.mozilla.org/en-US/firefox
Google Chrome	V37 or later	https://www.google.com/chrome

Table 3 –Supported Web browsers

3.4 Client PC / Laptop Hardware Requirements

UOC is fully compliant with mobile device and provide responsive screens.

Requirements	Minimal	Recomanded
CPU	2 cores	4 cores
RAM	1 GB	2 GB
WIFI	*802.11b/g/n*	*802.11ac*
Display Size	*14''*	*24''*

Table 4 –Hardware requirements for client PC

3.5 Mobile Device Hardware Requirements

UOC is fully compliant with mobile device and provide responsive screens.

Requirements	Minimal	Recomanded
CPU	2 cores	4 cores
RAM	1 GB	2 GB
WIFI	*802.11b/g/n*	*802.11ac*
Display Size	*Any*	*Tablet 10''*

Table 5 –Hardware requirements for mobile devices

3.6 Operating System

The only officially supported Operating System is: **Red Hat Linux 6.5 x86-64**.

You can check the current Linux release information by executing one of the following commands:

```
# lsb_release -id
```

or

```
# cat /etc/issue
```

3.7 Users

You need to have the **root** credentials for installing the packages. However, installed files will be owned by standard users, and no processes will run under the root account.

For security reasons, no Linux user is created automatically during the installation phase. The required users must therefore be created manually prior to the installation.

We recommend the creation of the following users on the system:

- **couchdb**: user owning the couchdb installation and running the couchdb processes or service.
- **uoc**: user owning all OSS console files, and running the web server (node.js process).

These two users are the ones expected by default. You can change them during the installation (see below).

If no specific user is given as command line option, and if the default ones (i.e. couchdb and uoc) do not exist on the system, **the installation will abort**.

You can create these users with the following commands:

```
# useradd -m uoc
# useradd -m couchdb
```

And don't forget to change the passwords:

```
# passwd uoc
# passwd couchdb
```

For security reasons, it is strongly recommended to change default passwords and use complex password. There is no check to identify weak passwords.

3.8 Node.js

The UOC server relies on Node.js V0.10.38 or above.

Node.js must therefore be installed on the system before installing UOC.

You can install node.js through npm (require an Internet connection) or using the standard rpm package:

```
# yum install nodejs
# node --version
v0.10.38
```

*NOTE: For convenience purposes, a NodeJS package is provided in the OSS console kits, and can be installed automatically by the installation script, but remember that **HP does not support NodeJS kit.***

Chapter 4

Deployment Examples

UOC Server is extremely scalable and supports multiple ways of deployment in order to adjust the volume of concurrent users and data.

Here are some examples of classic deployments. These ones need to be refined according to projects and use cases.

4.1 Monolithic

It is the simplest installation using only one machine to install all kits:

- CouchDB
- UOC V2
- UOC V2 – Add-on OSSA
- Data server (ex: OSS Analytic Server)

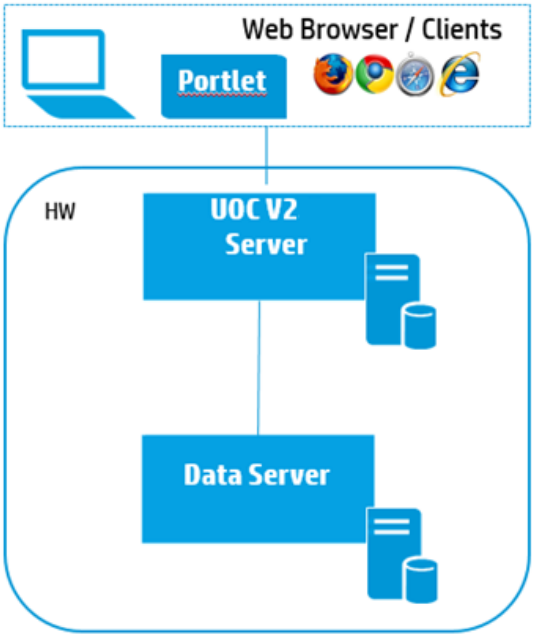

Figure 2: ***Deployment Examples – Monolithic***

4.2 Multiple instances

In this deployment, several instances of the UOC V2 server are running on one machine so as to use all its cores. The OSSA server uses a separate machine to handle all the requests coming from all UOC V2 users.

One machine with:

- CouchDB
- UOC V2
- UOC V2 – Add-on OSSA

One machine with:

- Domain or Data Server (ex: OSS Analytic Server)

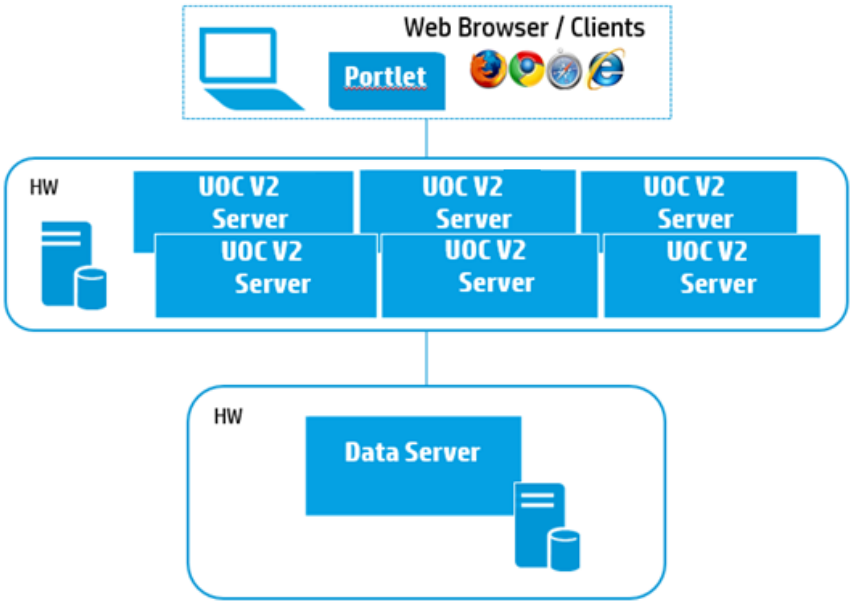

Figure 3: ***Deployment Examples – Multiples UOC Instances***

4.3 Multiple servers

In this deployment, several machines are used to run several instances of the UOC V2 server to support a large volume of data and users. The OSSA server is also installed on several separate servers to handle all the requests coming from all UOC V2 users.

One machine with:

- CouchDB

Several machines with:

- UOC V2
- UOC V2 – Add-on OSSA

One machine with:

- Domain or Data Servers (ex: OSS Analytic Server)

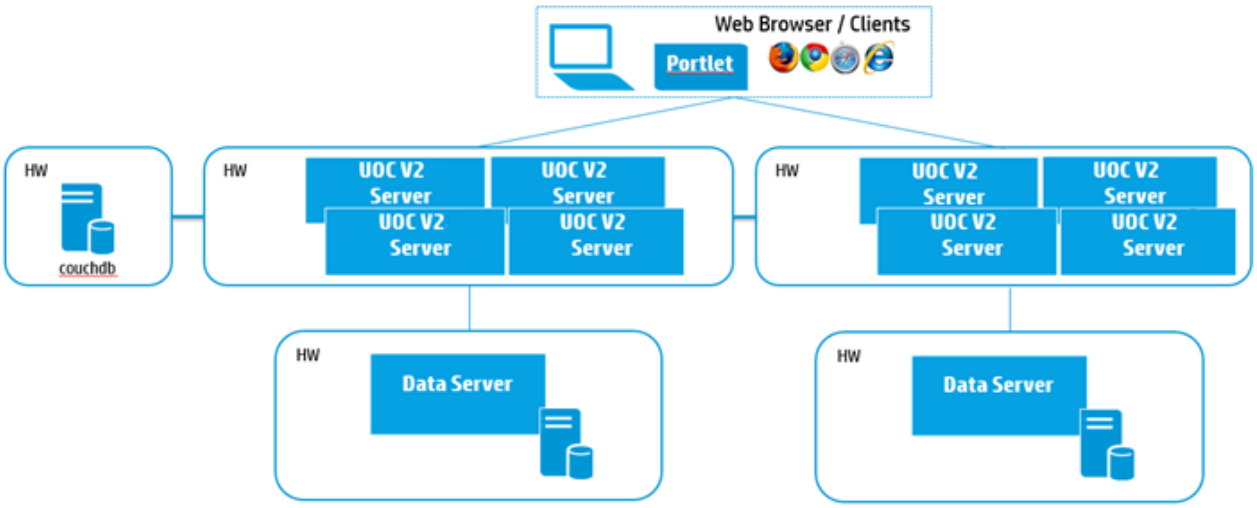

Figure 4: ***Deployment Examples – Multiples Servers***

Chapter 5

Installation

5.1 Installation locations

RPM packages have default installation locations:

- /opt/couchdb
- /opt/uoc2
- /var/opt/uoc2

5.2 Installation media

The Unified OSS Console V2.1 comes in a standard tar file:

```
UOCV2.1.0-MR.tar
```

Unpack the archive in a temporary directory of your choice:

```
$ tar xvf UOCV2.1.0-MR.tar
uoc2_kit/
uoc2_kit/install.sh
uoc2_kit/uoc-2.1.0-MR.x86_64.rpm
uoc2_kit/uoc-addon-ossa-1.1.0-MR.x86_64.rpm
uoc2_kit/nodejs-0.10.38-1nodesource.el6.x86_64.rpm
uoc2_kit/couchdb-1.6.0-1.el6.x86_64.rpm
uoc2_kit/README
```

5.3 Interactive script

The OSS Console is installed using an interactive shell script that will prompt for important options, like what packages to install, target locations on disk and users. For all options, a default value is proposed. If you want a standard installation, fully monolithic, with all default values, you can use the -s option (for **s**ilent or **s**cratch).

Interactive script:

```
# sudo install.sh
```

Or, install everything on the same system, using default values, no questions asked.

```
# sudo install.sh -s
```

5.4 User profile

The UOC ships an environment file, to be sourced by the uoc user. This will set in particular the UOC2_HOME and UOC2_DATA environment variables to the correct values and update the PATH to locate the uoc2 command.

```
# cat /var/opt/uoc2/.environment.sh >> /home/uoc/.bash_profile
# su - uoc
$ source /home/uoc/.bash_profile
$ which uoc2
/opt/uoc2/bin/uoc2
```

Source operation will override your PATH.

5.5 Setup

After a first installation, you need to create the CouchDB database and initialize the mandatory data. You can do this automatically:

Log first using the couchdb user, and execute the following script.

```
$ unset http_proxy
$ /opt/uoc2/scripts/setup.sh
```

You can execute this script as a normal user. However, if you wish to start the local CouchDB server on a newly installed local system, CouchDB credentials will be required.

Regarding the UOC V2 server, you can choose a few important parameters, like the CouchDB server hostname and port.

For this, edit the following file (JSON syntax):

```
$ vi /var/opt/uoc2/server/public/conf/config.json
```

If you use a basic local CouchDB installation you can keep the default values.

5.6 OSSA add-on configuration

If you use the OSS Analytics add-on, you need to provide the host and port of the OSSA server. For this, please edit the following file (JSON syntax):

```
$ vi /opt/uoc2/server/public/addons/plugins/ossa/config.json
```

5.7 Server startup

The server can be started with the following command:

```
$ uoc2 start
```

If you wish to change the TCP port in use, you can give a new port number on the command line:

```
$ uoc2 -p 2222 start
```

Then open the URL http://localhost:3000 (or any host/port combination) to log in the application.

---

*Unified OSS Console Default Port is **3000***

5.8 Red Hat Linux firewall settings

Netfilter is a host-based firewall for Linux operating systems. It is included as part of the Linux distribution and it is activated by default on RHEL6. This firewall is controlled by the program called iptables. Netfilter filtering takes place at the kernel level, before a program can even process the data from the network packet.

Therefore, when iptables is up and filtering packets, its settings should be modified in order to let the UOC server work properly. In particular, incoming HTTP(s) request on the UOC server port should be allowed.

Please refer to your system admin manual for configuring the firewall if necessary.

```
$ man iptables
```

Chapter 6

Installation validation

With default locations, the following files should be installed:

```
$ tree -L 1 /opt/uoc2
/opt/uoc2
├── bin
├── client
├── data -> /var/opt/uoc2/data
├── data.kitting
├── install
├── LICENSE.txt
├── logs -> /var/opt/uoc2/logs
├── node_modules
├── nohup.out
├── scripts
├── server
└── server.js

9 directories, 3 files
$ tree -L 1 /var/opt/uoc2
/var/opt/uoc2
├── client
├── data
├── logs
└── server

4 directories, 0 files
```

You can also check the installed packages on your system:

```
$ uoc2 inventory

HP UOC packages currently installed:

package                          summary
-------------------------------------------------------------

uoc-2.1.0-MR                     HP Unified OSS Console V2.1
uoc-addon-ossa-1.1.0-MR          HP Unified OSS Console V2.1 -
Addon OSS Analytics V1.1
nodejs-0.10.38-1.el6             JavaScript runtime
couchdb-1.6.0-1.el6              A document database server,
accessible via a RESTful JSON API
```

Chapter 7

Uninstallation

Installed packages can be removed interactively with the following command:

```
$ /opt/uoc2/scripts/uninstall.sh
```

It is also possible to uninstall all kits using the NPM commands.

Note:

This uninstallation has not removed the data directory and its possible customization. So the next installation of the Unified OSS Console V2 will re-use these data and will not override them.

Chapter 8

Administration

To ease administration of the UOC Server, several commands are available for the platform administrator.

These commands are only available for simple deployment (one instance of UOC Server per machine). For advanced deployment with multiples instance of the same machine, multiple UOC Server on several machines… these commands may be not appropriate.

Follow the documentation of advanced tools (node balancer, high availability, …) to put in place such configurations.

8.1 Start UOC Server

To start the UOC Server, run the following command with the user uoc.

```
$ uoc2 start
```

If you wish to change the TCP port in use, you can give a new port number on the command line:

```
$ uoc2 -p 2222 start
```

8.2 Check running UOC Server

To start the UOC Server, run the following command:

```
$ uoc2 show
```

8.3 Stop UOC Server

To stop the UOC Server, run the following command:

```
$ uoc2 stop
```

8.4 UOC Inventory

To list the installed kits on the platform, an administrator can run the following command:

```
$ uoc2 inventory

HP UOC packages currently installed:

package                          summary
-------------------------------------------------------------------

uoc-2.1.0-MR                     HP Unified OSS Console V2.1
uoc-addon-ossa-1.1.0-MR          HP Unified OSS Console V2.1 - Addon OSS
Analytics addon V1.1
nodejs-0.10.38-1.el6             JavaScript runtime
couchdb-1.6.0-1.el6              A document database server, accessible via a
RESTful JSON API
```

8.5 Start CouchDB Server

To start the CouchDB Server, run the following command with the user root.

```
$ /etc/init.d/couchdb restart
```

8.6 Stop CouchDB Server

To stop the CouchDB Server, run the following command with the user root.

```
$ /etc/init.d/couchdb stop
```

Chapter 9

Platform Configuration

It is possible to configure these following setting for the UOC platform.

9.1 UOC Server

The UOC Server has a configuration file where it is possible to customize some paramaters. The configuration file is stored in **<install_dir>/server/public/conf/config.json**

Example:

```
 {...
"server": {
    "protocol" : "http",
    "port" : "3000",
    "timeout" : 0
 },
...}
```

Where

- **Protocol** is the protocol used by the server (http or https). It can be ovveridded by an environment variable named **PROTOCOL.**
- **Port** is the port of the server (default is 3000). It can be overrided by an environment variable named **PORT.**
- **Timeout** is the timeout in seconds allowed for all http/https requests. Defualt is 0s (unlimited). It can also ovveruided by the environment variable **TIMEOUT.**

9.2 UOC Data Import

During startup of the UOC server, there are 2 steps:

1. the server browse all available plugin to contact all associated domain server (or data server), and collect all available value pack and their GUI resources (like workspaces and views).

   All graphical resources (Data UI) found are imported into the GUI database for sharing and usage with all users.

The reference is the GUI database to access to the graphical resource. Backup this database is strongly recommended to avoid any lost.

See 10.1 Apache CouchDB for detailed options for backup UOC data.

All value pack definitions (Metadata UI) are loaded dynamically in the UOC server and keep until the stop of the server.

2. the server browse all data defined locally in a specific local directory **<install_dir>/data**

This directory defines several definitions to use to initialize the GUI database:

- Workspace categories
- Launch categories
- Launches
- Roles
- Permissions
- States
- Users (local authentication mode only)

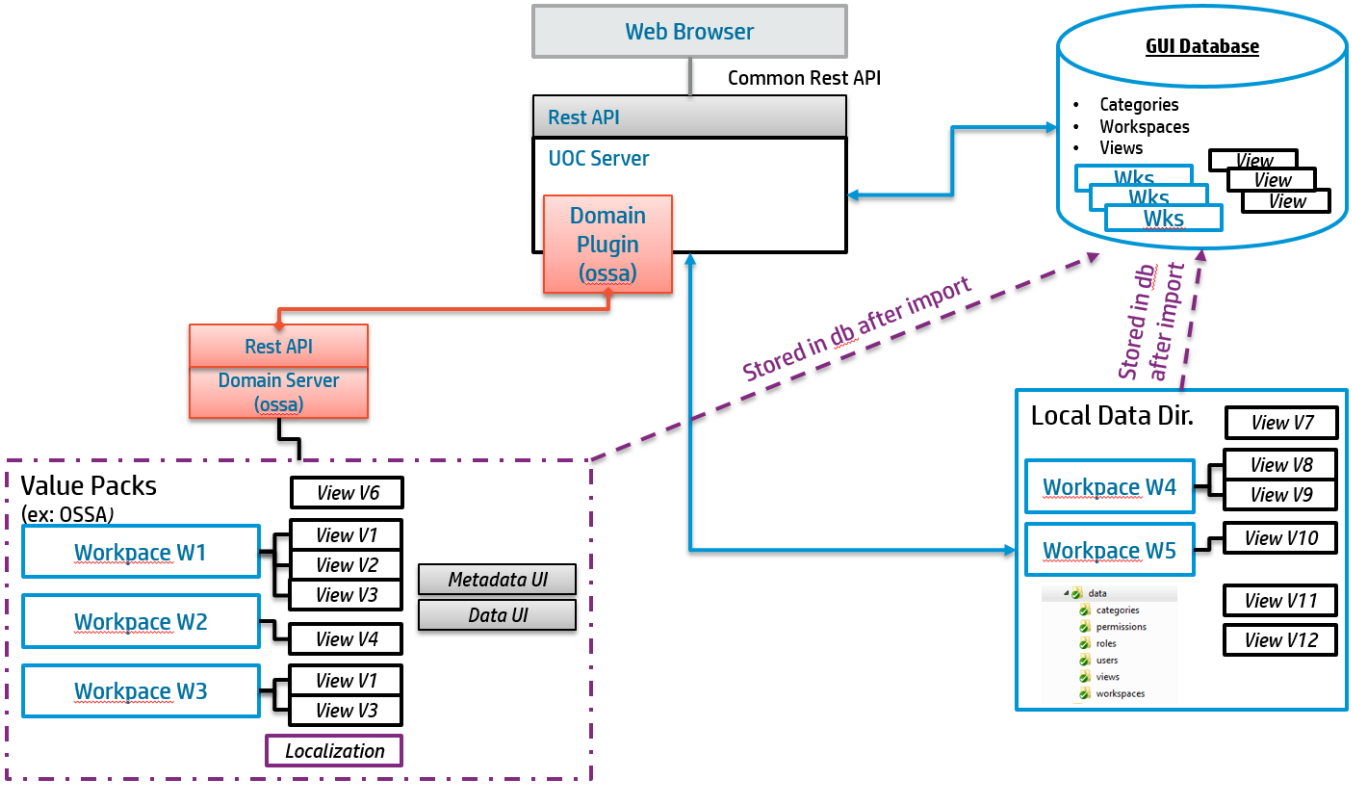

Figure 5 – Data Import Overview

Default behavior *executes these 2 steps:*

1) browse the plugin and their domain server to collect and import resources from servers, and

2) import ONLY additional data present in the local directory.

It is possible to customize the data import policy to ignore one specific step and give priority to data found on the server or the local directory.

An administrator can easily choose to always override in the GUI database data using the last data found on the server during the UOC start and make sure the reference is always on the server and up to date for UOC.

To customize the data import policy, edit the configuration file:

<install_dir>/server/public/conf/config.json

```
...
  "startup": {
    "loadLocalUIData": true,
    "overwriteLocalUIData": false,
    "loadRemoteUIData": true,
    "overwriteRemoteUIData": false
  }
...
```

Property	Value	Default	Description
loadLocalUIData	true | false	true	Import local graphical data from <install_dir>/data into the GUI database
overwriteLocalUIData	true | false	false	Override GUI database with local graphical data if the same identifier is found. Reference is the local data directory.
loadRemoteUIData	true | false	true	Import graphical data retrieved from packages in all defined server into the GUI database
overwriteRemoteUIData	true | false	false	Override GUI database with server graphical data if the same identifier is found. Reference is all servers data.

When the platform administrator starts the UOC server, the console displays the current settings.

It is strongly recommended to turn off override options and make sure identifier of objects (workspace, view…) are not duplicated between the 2 sources (local and server).

The GUI database is the reference for all the graphical objects of UOC. Server's packages usually provide default workspaces and views, but these objects can be customized by integrator,operators and view designer.

Chapter 10

Platform Backup

There are 2 types of data to backup on the platform:

- Data stored in the document Database (couchdb) like views, workspaces, users, permissions, roles...
- Configuration files updated to defines correctly hostname, port, specific settings (authentication...)

10.1 Apache CouchDB

This part of the document assumes that the installation of Apache CouchDB was made in the default directory /opt/couchdb. In case another directory was used, be careful to adapt the following commands.

The following two parts describe how to backup or replicate all UOC CouchDB databases or one or several of these databases. UOC database names are:

- categories
- permissions
- roles
- users (local authentication only)
- views
- workspaces
- launches
- launch-categories

You have two options to save/restore a couchdb database. You can save the data directory or you can replicate data from the database to another instance of couchdb (replication). Replication is the recommended method because it grants you to always have a working couchdb database. You will not need any restore operations.

10.1.1 Backup CouchDB database files

First, stop CouchDB and applications that could use it like UOC.

Then, save the directory /opt/couchdb/var/lib/couchdb.

(replace DIRECTORY by your backup destination directory)

```
tar -cvf DIRECTORY/couchdb.tar /opt/couchdb/var/lib/couchdb
```

If you want to backup only specific databases files, you can save the files named by the database's name and with the extension *.couch*.

Restart CouchDB.

10.1.2 Restore CouchDB database files

This part assumes you generated an archive by following the steps described in the previous section above (see 10.1.1 Backup CouchDB database files)

If the CouchDB database targeted for the backup recovery is new, be sure to have correctly and totally installed it and initialized it.

Stop CouchDB and applications that could use it like UOC.

Use your backup archive to restore CouchDB database files.

(replace DIRECTORY by your backup archive directory)

```
tar -xvf DIRECTORY/couchdb.tar /opt/couchdb/var/lib
```

Restart CouchDB.

10.1.3 Replicate CouchDB

Replication synchronizes two copies of the same database, allowing users to have low latency access data no matter where they are. These databases can live on the same server or on two different servers—CouchDB doesn't make a distinction. If you change one copy of the database, replication will send these changes to the other copy.

10.1.3.1 Simple Replication with the Admin Interface

You can run replication from your web browser using Futon, CouchDB's built-in administration interface.

Prerequisites:

- CouchDB has to be started
- Access to http://SERVER_IP:5984/_utils has to be enabled.

If it is not the case, stop CouchDB and modify this file:

```
nano /opt/couchdb/var/config/couchdb/local.ini
```

Add the following two lines under [httpd]:

```
[httpd]
port = 5984
bind_address = 0.0.0.0
```

Start CouchDB.

Open your browser to http://SERVER_IP:5984/_utils. On the right-hand side, you will see a list of things to visit in Futon. Click on “Replication.”

Futon will show you an interface to start replication. You can specify a source and a target by either picking a database from the list of local databases or filling in the URL of a remote database.

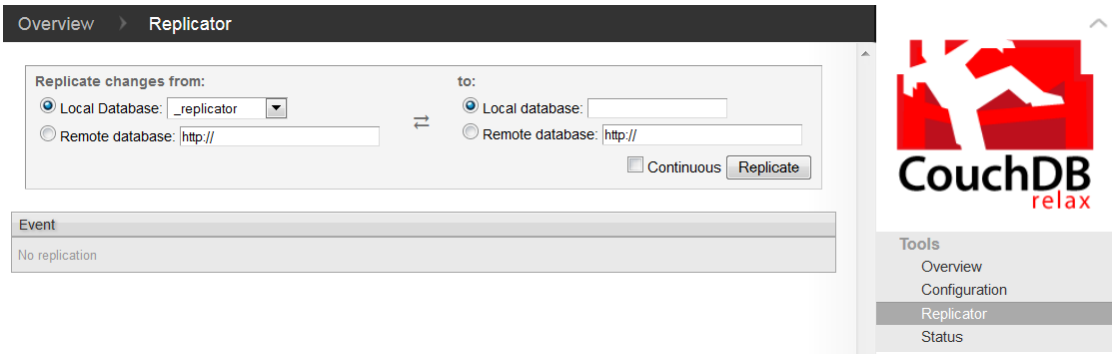

Figure 6: CouchDB Futon Replication Interface

Click on the Replicate button, wait a bit, and have a look at the lower half of the screen where CouchDB gives you some statistics about the replication run or, if an error occurred, an explanatory message.

For additional information about CouchDB replication possibilities, please visit http://guide.couchdb.org/editions/1/en/replication.html.

10.2 UOC Data

All the configuration files and data that the administrator can customize on the UOC platform are saved in the <install_data_dir> (usually /var/opt/uoc2)

After uninstallation, these files are not removed and will not be overridden by a new UOC installation.

It is recommended to integrate these data directories in a backup process.

Chapter 11

Security Guide

This chapter highlight security topic and give all the useful information to harden the UOC platform.

11.1 Terminology

Given terminology may vary from one person another, or depending on the context the following terms and their meaning are defined below.

User Security	Refers to security mechanism supported by HP UOC which enables control of who the user is (authentication), what the user can do (authorization), and what the user did (auditing[1]).
Privacy	Refers to personally identifiable information about individuals (subscribers), including their behavior, service usage, location, etc. subject to privacy laws which vary from country to country. In the UOC context, it refers uniquely identifiable data relating to a person or persons which is collected, stored, processed, maintained, and made visible by HP UOC. For example, data can be considered as private when it combines the subscriber identity, the consumed services and urls, geographical location and more.
Encryption	Refers to the process of encoding messages (or information) in such a way those systems external to HP UOC and non HP UOC users (persons, hackers) cannot read it, but that authorized HP UOC Users can read. HP UOC uses encryption to protect exchanged data (between UOC modules and/or between UOC modules and external systems), stored data, as well as user access to the data. Note that in the telecom context, data carried between the user equipment (device) and the network can be encrypted. This is usually names ciphering. When ciphering is implemented in the network, gaining visibility on individual user transactions requires deciphering. In the HP UOC context, the different network and service transactions are collected via probes. As such ciphering/deciphering in network protocols is not relevant to the HP UOC platform.

[1] Another wide spread terminology is Accounting (in the context of AAA and related protocols such as Diameter). Here the scope is at OSS software application level rather than network protocol.

Audit	Refers in this document to the ability to log and track access to files, directories, and resources of the systems where HP UOC is installed. Note that auditing is not part of HP UOC per se, but can be setup at the OS level to log all user actions.
Hardening	Refers to providing various means of protection in a computer system to eliminate as many security risks as possible. This is typically done by removing all non-essential software programs and utilities from the computer(s) by configuring the system and network components properly, deleting unused files and applying the latest patches.

Finally, note that where words are written in italic, these refer to HP UOC specific files/and or command and/or syntax.

11.2 Overview

UOC Server has many security options that can be used to secure the platform and usage of the server.

All the black box below are described in this chapter.

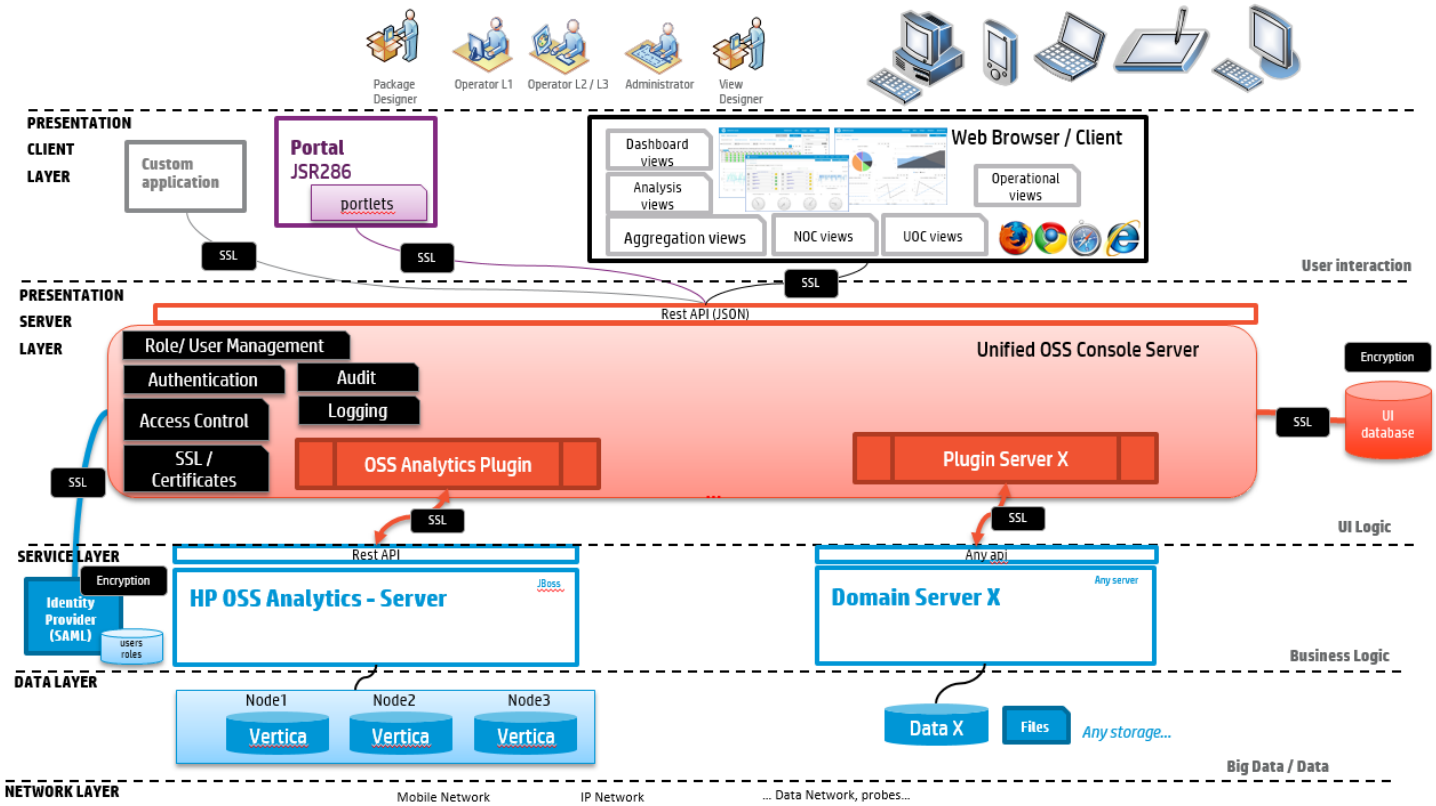

Figure 7: UOC Server - Security Overview

11.3 Authentication

The product supports two way of authentication:

- **SAML** (Security Assertion Markup Language): Unified OSS Console provides an integration with identity providers through the SAML V2.0 protocol. Users can be managed externally (LDAP, files ...) and the product supports the SSO using this identity provider. It is the recommended production mode for large volume of users and to grant a high level of security.

  Note: The Open source project Picket Link (http://picketlink.org) is an option as a identify provider and has been tested with our solution.

  see 11.3.2 Enable SSO / SAML Configuration

- **Local**: It is a built-in authentication mode based on a local Document database in charge of managing the users and their associated roles. It is mainly for demo purpose or very small deployment. This mode does not support the SSO and does not provide a high level of security. It is not recommended to use it in production.

  See 11.3.1 Enable Local Authentication

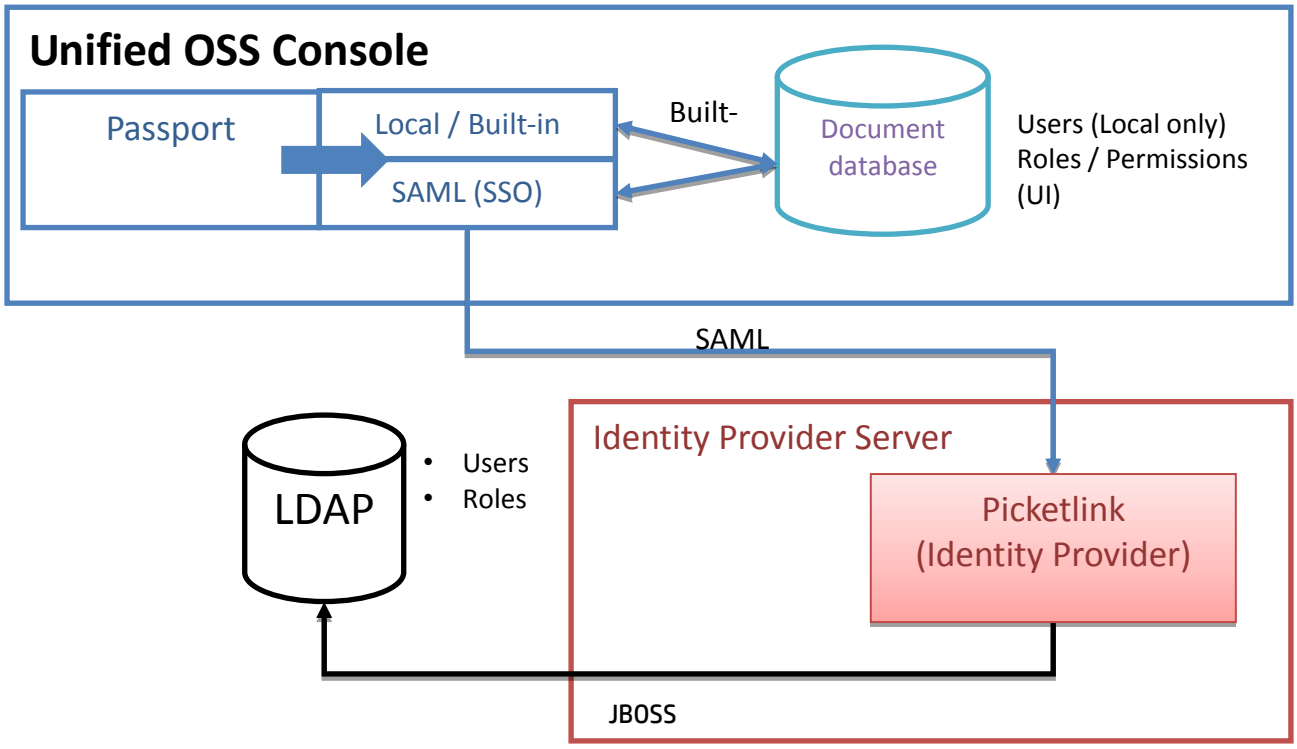

Figure 8 – UOC Authentication modes (SAML / Local)

To enforce the security, the connection to SAML Identity Provider and the SAML token can be encrypted with certificates.

11.3.1 Enable Local Authentication

To enable the local authentication mode, please edit the file <install_data_dir>/server/public/conf/config.json and set the authentication mode to 'local'. Also, ensure your database settings are accurate (couchdb host, user, password)

Example of config.json:

```
...
  "database": {
    "protocol": "http",
    "host": "127.0.0.1",
    "port": "5984",
    "username": "user",
    "password": "user"
  },

  "authentication": {
    "mode": " local ",
    ...
  },
...
```

11.3.2 Enable SSO / SAML Configuration

To enable the SAML authentication mode, please edit the file <install_data_dir>/server/public/conf/config.json

Example of config.json:

```
...
"authentication": {
      "mode": "saml"
  },
  "saml": {
    "idp": {
      "entryPoint": "http://localhost:8080/idp",
      "identifierFormat": "urn:oasis:names:tc:SAML:2.0:nameid-format:entity"
    },
```

```
    "sp": {
      "issuer": "http://localhost:3000"
    }
  },
...
```

The attribute ‘mode’ under ‘authentication’ set the SAML authentication mode.

The ‘saml’ part defines several mandatory options regarding the SAML authentication. The ‘idp’ subpart concerns your Identity Provider whereas ‘sp’ concerns your UOC server.

About the ‘idp’ subpart:

- ‘entryPoint’ sets your Identity Provider entry point for authentication requests. Authentication requests are carried in the URL query string of an HTTP GET request.
- identifierFormat’ indicates what SAML name identifier format you want to use.

About the ‘sp’ subpart:

- ‘issuer’ defines the EntityID of the UOC server (Service Provider), usually the URL to access to your UOC application.

Note: it is strongly recommended to turn on the SSL certificate encryption between UOC Server and the Identity Provider, and increase the security level using SSL certificate to encrypt the SAML token used for the user session (see your identify provider documentation to enable SSL support)

11.3.3 Configure JSON Web Token

Authentication generates JSON Web Token in the user session. 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 digitally signed using JSON Web Signature (JWS).

You can get more details about this token on the website:
https://tools.ietf.org/html/draft-ietf-oauth-json-web-token-32

Our JSON Web token integrates several informations like:

- Issued token generation date
- Expiration token date
- User identifier
- User name

- A set of user roles (list of role identifier)

To configure this token generation, lease edit the file <install_data_dir>/server/public/conf/config.json and set the jwt section.

You can change the algorithm used by the generator (default is HS256), the secret passphrase and the expiration time (in minutes)

Example of config.json:

```
...
  "authentication": {
      ...
    "jwt": {
      "algorithm": "HS256",
      "expiresInMinutes": 1440,
      "secret": "TheWalkingSkeleton"
    }
  },
...
```

It is possible to customize the JSON Web Token algorithm (default is HS256) , expiration time (default is 24h) and secret passphrase in the same configuration file.

11.3.4 Generate JSON Web Token

It is possible to generate valid JSON Web Token externally using multiple tools.

You can use for example the following web sites:

http://jwt.io/

http://jwtbuilder.jamiekurtz.com/

The generated token can be used to call the UOC REST API. For example, it is very useful if you want to export regularly PDF reports from UOC.

To generate a valid token, you must ensure the configuration of the authentication match the parameters you define for your external tools (passphrase, user id, role identifiers…)

It is also possible to get a valid token in the preferences of the user.

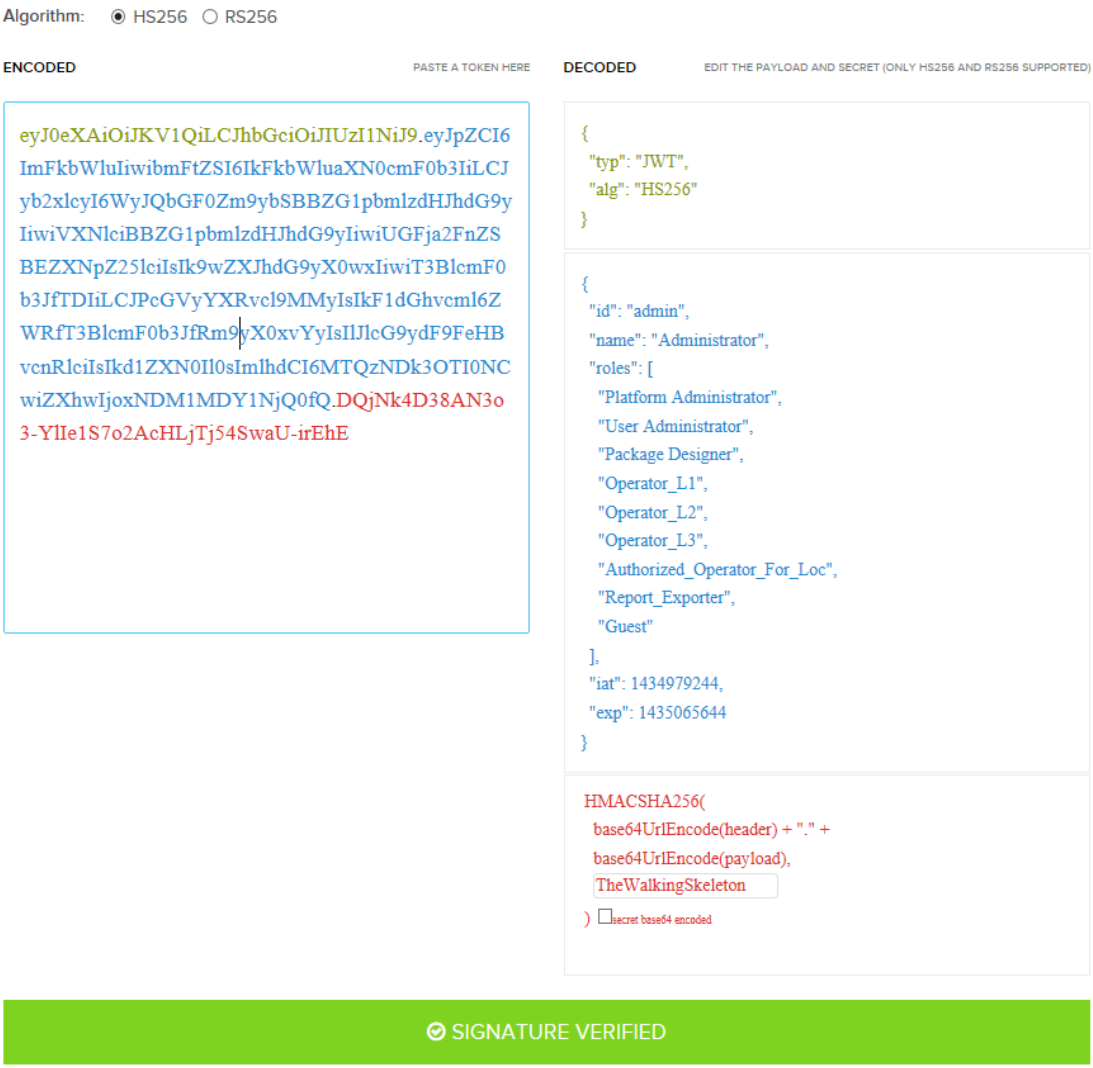

Figure 9 – Example of generation of JSON Web Token (http://jwt.io)

It is mandatory to follow this syntax to make sure the token is valid in the Unified OSS Console:

{

"**id**": "admin",

"**name**": "Administrator",

"**roles**": [

"Platform Administrator",

"User Administrator",

"Package Designer",

"Operator_L1",

"Operator_L2",

"Operator_L3",

],

"**iat**": 1434979244,

"**exp**": 1435065644

}

Where:

- Id is the identifier of the user
- Name is the name of the user
- Roles are the list of role identifier associated to the user
- Iat is the issued-at time
- Exp is the expiration date/time

11.4 User and Password Management

If the local authentication mode is used, all the user sensitive information like the user password is stored in the GUI Database (couchDB).

As to the password, they are stored in a protected form of 128 bits long, hashed using the **PBKDF2** algorithm.

In order to hash the passwords, a salt is generated using a cryptographically secure pseudo-random function and is 64 bits long as recommended by the standard. The salt is then prepanded to the hashed password (64 bits long) saved into the database.

Note: it is strongly recommended to turn on the SSL certificate encryption between UOC Server and the GUI Database to increase the security level and avoid any password sniffing.

11.5 Role Based Access Control (RBAC)

The product integrate a strong role based access control that drive the user interface and the access to information. The Definition is based on the standard ANSI INCITS 359-2004.

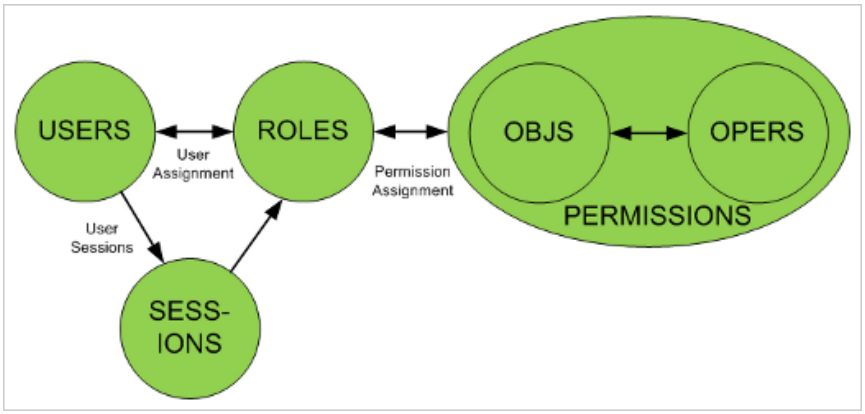

Figure 10 – RBAC - ANSI INCITS 359-2004

A role is Job function or title which defines an authority level

A User will have one or several roles (ex: Operator Level 1)

A role has one or several permissions

A permission is an approval of a mode of access to a resource. It is defined by an operation and an object or resource (ex: create user)

As soon as the end user has been authenticated, a list of his roles are checked, a list of permissions are loaded and the graphical interface always apply these access right for display and actions (ex: Only a user administration can have access to administration page in charge of creating users in the platform, only an operator level 3 can open a set of advanced analysis dashboards, etc...)

11.5.1 Roles

By default, a set of roles are defined by default:

- Guest
- User Administrator
- Platform Administrator
- Operator Level 1
- Operator Level 2
- Operator Level 3
- Package (or value pack) Designer
- View Designer
- Report Exporter

And this can be extended through your identity provider for the SAML authentication, or using the Role Administration page in the application for the local built-in authentication.

Default set of permission on the user interface are:

Category	*Operation*	*Object*	Guest	Platform Administrator	User Administrator	Operator L1	Operator L2	Operator L3	Package Designer	View Designer
Workspace Management										
	browse	workspace	x			x	x	x		x
	edit	workspace					x	x		x
	create	workspace					x	x		x
	delete	workspace					x	x		x
	save	workspace					x	x		x
	create	view								x
	edit	view								x
	delete	view								x
	add	view					x	x		x
	remove	view					x	x		x
	configure	widget				x	x	x		x
	configure	datasource					x	x		x
	configure	filter				x	x	x		x
	configure	top					x	x		x
	export	report						x		x
	export	data					x	x		x
Category Management										
	create	category		x						x
	delete	category		x						x
	edit	category		x						x
	browse	category		x						x
Launch Category Management										
	create	launch_category		x						x
	delete	launch_category		x						x
	edit	launch_category		x						x
	browse	launch_category		x						x
Package Management										
	browse	package		x					x	x
Theme Management										
	configure	theme		x						
Addons Management										
	browse	layout		x						x
	browse	widget		x						x
	browse	plugin		x						x
	browse	menu-bar		x						x
	browse	menu-item		x						x
	browse	theme		x						x
	browse	module		x						x
User Management										
	create	user			x					
	delete	user			x					
	edit	user			x					
	browse	user			x					
Role Management										
	create	role			x					
	delete	role			x					
	edit	role			x					
	browse	role			x					
Platform Management										
	edit	setting		x						
	browse	token		x						
Launch Management										
	create	launch		x						x
	delete	launch		x						x
	edit	launch		x						x
	execute	launch		x			x	x		x

Table 6 – RBAC – Default Roles / Permissions

The list of roles associated to a user will impact the user interface available and operations he is able to execute.

It is strongly recommended to tune fine grain the role and permissions of a user for security reasons.

Example: An operator level 1 has very few available actions and limited access to specific dashboard. It will also impact the available list of dimensions and facts in the analysis tool.

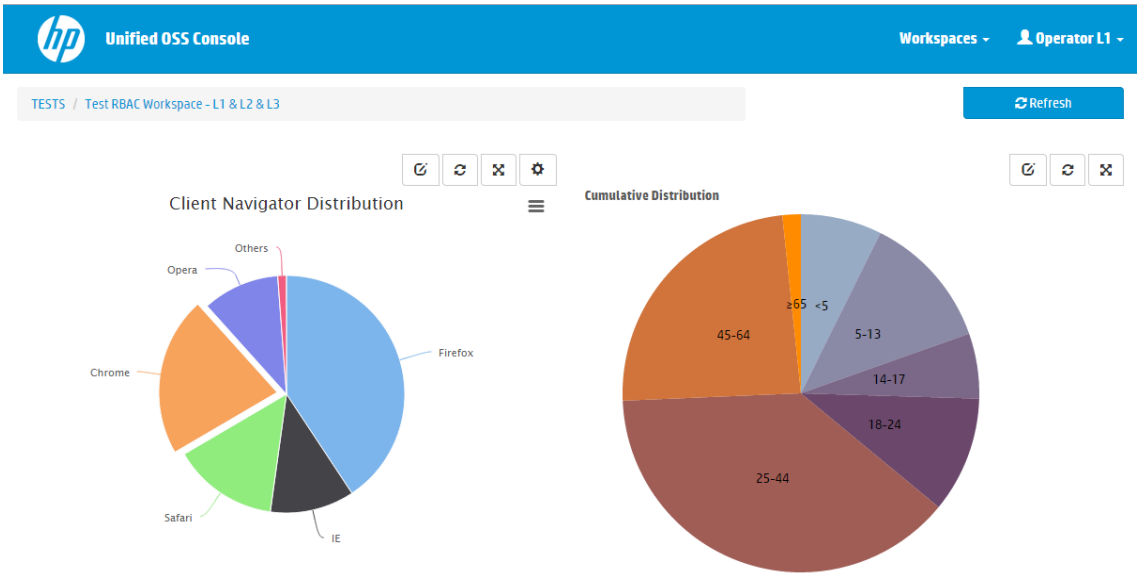

Figure 11 – Example of operator level 1 dashboard

In this dashboard, the menu is simple and only allows access to workspaces. The Workspace management operations have been hidden (save, delete…).

11.5.2 Permissions

UOC has an internal list of pre-defined permissions that will impact the User interface. When an administrator define a new role, he needs to associate some of these permissions to indicate to UOC what actions are available for the connected user.

The permission list is today not extensible and only User interface oriented, and deeply linked to the UOC web application. It is totally different from permissions you may need to tune the privacy setting.

Here is the list of available permission:

Group	Operation	Object	Identifier	Description
Workspace Management	Browse	Workspace	browse_workspace	Allows the user to browse and display workspaces
	Create	Workspace	create_workspace	Allows the user to create workspaces
	Delete	Workspace	delete_workspace	Allows the user to delete workspaces
	Save	Workspace	save_workspace	Allows the user to save or save as… workspaces
	Edit	Workspace	edit_workspace	Allows the user to edit properties of workspaces
	Create	View	create_view	Allows the user to create new views
	Delete	View	delete_view	Allows the use to delete views
	Edit	View	edit_view	Allows the user to edit views
	Add	View	add_view	Allows the user to add views to existing workspaces

Group	Operation	Object	Identifier	Description
	Remove	View	remove_view	Allows the user to remove views from an existing workspace
	Configure	Widget	configure_widget	Allows the user to access to the configuration panel of a widget
	Configure	Datasource	configure_datasource	Allows the user to select the data to analyze
	Configure	Filter	configure_filter	Allows the user to define filters on dimension for the data to analyze
	Configure	Top	configure_top	Allows the user to define top filters for the data to analyze
	Export	Data	export_data	Allows the user to export data
	Export	Report	export_report	Allows the user to export report
Launch Category Management	Browse	Launch Category	browse_launch_category	Allows the user to browse and display workspaces
	Create	Launch Category	create_ launch_category	Allows the user to create launch categories
	Delete	Launch Category	delete_ launch_category	Allows the user to delete launch categories
	Save	Launch Category	save_ launch_category	Allows the user to save or save as… launch categories
	Edit	Launch Category	edit_ launch_category	Allows the user to edit properties of launch categories
Category Management	Browse	Category	browse_category	Allows the user to browse and display workspaces
	Create	Category	create_ category	Allows the user to create workspace categories
	Delete	Category	delete_ category	Allows the user to delete workspace categories
	Save	Category	save_ category	Allows the user to save or save as… workspace categories
	Edit	Category	edit_ category	Allows the user to edit properties of workspace categories
Theme Management	Configure	Theme	configure_theme	Allows a user to modify the selected theme browsing the available list of themes.
Package Management	Browse	Package	browse_package	Allows a user to browse packages available in the platform
Add-ons Management	Browse	Layout	browse_layout	Allows a user to browse layout available in add-ons
	Browse	Widget	browse_widget	Allows a user to browse widgets available in add-ons
	Browse	Plugin	browse_plugin	Allows a user to browse plugins available in add-ons
	Browse	Menu item	browse_menu_item	Allows a user to browse menu items available in add-ons
	Browse	Menu bar	browse_menu_bar	Allows a user to browse menu bars available in add-ons
	Browse	Menu theme	browse_theme	Allows a user to browse themes available in add-ons
	Browse	Menu module	browse_module	Allows a user to browse modules available in add-ons
User Management	Browse	User	browse_user	Allows a user to browse available users on the platform (local authentication mode only)
	Create	User	create_user	Allows a user to create a new user on the platform (local

Group	Operation	Object	Identifier	Description
				authentication mode only)
	Delete	User	delete_user	Allows a user to delete an existing user on the platform (local authentication mode only)
	Edit	User	edit_user	Allows a user to edit an existing user on the platform (local authentication mode only)
Role Management	Browse	Role	browse_role	Allows a user to browse available roles on the platform
	Create	Role	create_role	Allows a user to create a new role on the platform
	Delete	Role	delete_role	Allows a user to delete an existing role on the platform
	Edit	Role	edit_role	Allows a user to edit an existing role on the platform
Platform Management	Edit	Setting	edit_setting	Allows a user to change settings on the platform
	Browse	token	browse_token	Allows a user to browse his authentication token on the platform
Launch Management	Create	Launch	create_launch	Allows a user to create new launches
	Delete	Launch	delete_launch	Allows a user to delete launches
	Edit	Launch	edit_launch	Allows a user to edit launches
	Execute	Launch	execute_launch	Allows a user to execute launches

Table 7 – RBAC –List of user interface permissions

11.6 CouchDB Database

A CouchDB server hosts named databases, which store **documents**. Each document is uniquely named in the database, and CouchDB provides a RESTful *HTTP API* for reading and updating (add, edit, delete) database documents.

All the information related to CouchDB can be found on the official web site: http://couchdb.apache.org

11.6.1 Built-in Administration

Administration can easily administrate the database using Futon, the built-in administration interface.

http:<host>:<port>/_utils

Example:

http://127.0.0.1:5984/_utils

Futon provides full access to all of CouchDB's features. Futon let you create and destroy databases; view and edit documents. Futon is also protected by a user/password. It is strongly recommended to use complex password to secure access direct to this database.

Note: this GUI database does not contains sensitive data but definitions used by the web application. So, it is recommended to enable the SSL support, do not use default user/password to increase the security level.

11.6.2 Configuration

It is possible to configure the CouchDB database to use for UOC server editing the following configuration file.

<install_dir>/server/public/conf/config.json

...

```
"database": {
    "protocol": "http",
    "host": "127.0.0.1",
    "port": "5984",
    "username": "user",
    "password": "user",
    "adminPassword": "password_of_couchdb_admin_user"
},
```

...

It is possible to setup the protocol to use (default is http). It is strongly recommended to enable the SSL support to access to the GUI database.

It is possible to configure the host, port, user and password of the couchDB user.

It is possible to specify the password of the CouchDB user "admin".

This definition is used by the UOC server only to access to the database.

11.7 Secure Socket Layer (SSL)

11.7.1 SSL Certificates

There are 2 types of certificates:

- Self-signed: A self-signed certificate is an identity certificate that is signed by the same entity whose identity it certifies. This term has nothing to do with the identity of the person or organization that actually performed the signing procedure. In technical terms a self-signed certificate is one signed with its own private key.
- Certificate Authority signed: A digital signature from a certificate authority (CA) attests that a particular public key certificate is valid (i.e., contains correct information). Users, or their software on their behalf, check that the private key used to sign some certificate matches the public key in the CA's certificate.

It is strongly recommended to use in production certificate provided by a Certifiate authority for security reasons.

11.7.1.1 Generate Certificates

Here is an example of private key and certificate request (csr) generation with command-lines.

First step is to generate a private key file, then generate a CSR (Certificat Signing Request). A CSR or Certificate Signing request is a block of encrypted text that is generated on the server that the certificate will be used on. It contains information that will be included in your certificate such as your organization name, common name (domain name), locality, and country. It also contains the public key that will be included in your certificate. A private key is usually created at the same time that you create the CSR

```
openssl genrsa -des3 -out server.key 2048
openssl req -new -key server.key -out server.csr
mv server.key server.key.org
openssl rsa -in server.key.org -out server.key
```

Second step is to generate a SSL certificate from the CSR, then obtain a certificate file from the certificate request file (csr ➔ crt).

Generate a Self-Signed SSL Certificate

```
openssl x509 -req -days 365 -in server.csr -signkey
server.key -out server.crt
```

Generate a Authority Signed SSL Certificate

Ask your signing authority provider : submit the server.csr and get the ssl certificate server.crt

11.7.2 Overview

By default, secured communication across the server's components is not enabled, since it requires first the administrator to install his platform's certificate for the SSL authentication.

For very high security requirements, it is strongly recommended to activate SSL with authorities certificates (certified) to protect all the http channel.

The Unified OSS Console supports SSL encryption at several communication channel to grant a high level of security in production mode.

- UOC Server ←→ Web browser
- UOC Server ←→GUI Database
- UOC Server ←→ Identity Provider (IdP) / SSO
- UOC Server ←→Domain or data server (ex: OSS Analytics server)

Certificates can be generated and configured in the platform configuration to enable the SSL mode. These certificates can be self-signed but it strongly recommended to use a trusted authority to generate appropriate trusted certificates.

All certificates need to be copied in the <install_data_dir>/server/public/ssl

---

The following HTTP communications can be encrypted with one or different certificates.

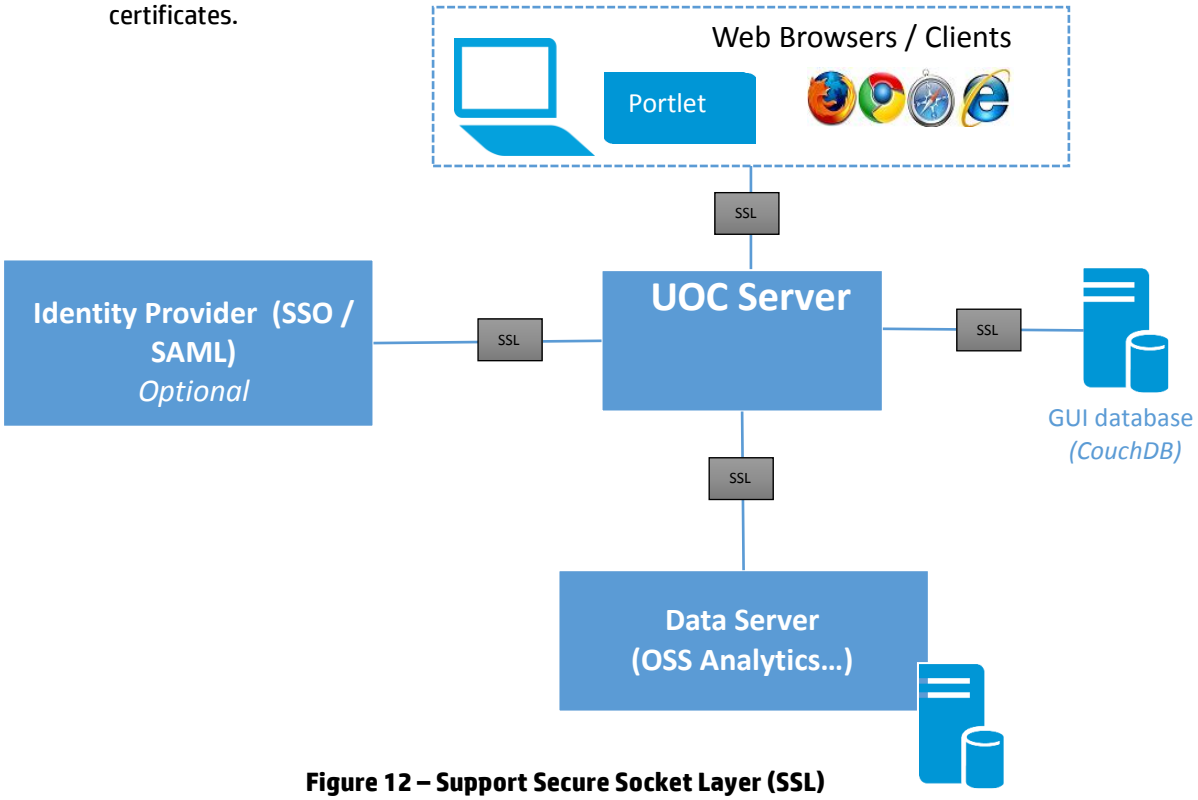

Figure 12 – Support Secure Socket Layer (SSL)

Note: Certificates are stored in a certificates directory on the UOC Server. These certificates can be self-signed (less secure) or trusted (recommended).

If you use the automatically generated self signed example certificate, the encryption is RSA (key length: 2048).

But, by using your company SSL certificate, you could have other encryption algorithm and key length depending on the company policy.

11.7.3 GUI Database (Couchdb server)

To enable SSL support in the GUI database (couchdb server), edit this file before (re)starting the Couchdb server:

<couchdb_install_dir>/etc/couchdb/local.ini

Uncomment the httpsd line and set the path to your server key and crt files , like this:

...

[daemons]

; enable SSL support by uncommenting the following line and supply the PEM's below.

; the default ssl port CouchDB listens on is 6984

httpsd = {couch_httpd, start_link, [https]}

[ssl]

cert_file = /path/to/ssl/files/server.crt

key_file = /path/to/ssl/files/server.key

...

11.7.4 UOC ←→ GUI Database

To configure SSL access from the UOC server to the GUI database,

Edit the protocol, host and port as needed in this file:

<install_dir>/server/public/conf/config.json

Signed certificate case:

Add the name of a certification authority certificate file, if needed. This file is obtained from you signing provider and must be copied in:

<install_dir>/server/public/ssl

```
...
  "database": {
    "protocol": "https",
    "host": "127.0.0.1",
    "port": "6984",
    "username": "user",
    "password": "user",
    "ssl": {
        "caCertFile": "gui_db_ca.crt"
    }
  },
...
```

Self signed certificate case:

Add the strictSSL parameter to disable strict certificate checking.

```
...
  "database": {
    "protocol": "https",
    "host": "127.0.0.1",
    "port": "6984",
    "username": "user",
    "password": "user",
    "ssl": {
        "strictSSL": false
    }
  },
...
```

11.7.5 UOC ←→ Web browser

To enable the SSL support in the UOC server. The server private key and certificate files must be copied in

<install_dir>/server/public/ssl

then edit the protocol, port, and name of the key and certificate files in this file:

<install_dir>/server/public/conf/config.json

```
...
"server": {
    "protocol" : "https",
    "port" : "3443",
    "privateKey" : "server.key",
    "certificate" : "server.crt"
  },
...
```

Access a SSL enabled UOC server from a browser

When using a signed certificates, a certification authority certificate file obtained from your signing provider should be imported in the browser.

11.7.6 UOC ←→ Identity Provider (IdP) / SSO

The UOC server supports the use of signature and encryption of SAML assertions (SHA1/RSA).

To enable these optional features, please edit the file **<install_data_dir>/server/public/conf/config.json.**

As previously mentioned, all the certificates must be copied in the folder **<install_data_dir>/server/public/ssl**.

Only certificates in the PEM format are supported as to the SAML configuration.

Example of SAML configuration with SSL token encryption:

```
"saml": {
    "idp": {
      "entryPoint": "http://localhost:8080/idp",
      "identifierFormat": "urn:oasis:names:tc:SAML:2.0:nameid-format:entity",
      "certificate": "idpcert.pem"
    },
    "sp": {
      "issuer": "http://localhost:3000",
      "privateKey": "nodekey.pem"
    },
    "signature": true,
    "encryption": true
  },
```

See 11.3.2 Enable SSO / SAML Configuration

for the description of the basic settings, additional security attributes are described here.

About the 'idp' part:

- **'certificate'** set the public certificate of your Identity Provider. This certificate is used for signing the generated SAML assertions.

About the 'sp' part:

- **'privateKey'** set the private certificate of your UOC2 server. This certificate is used both for verifying the signature of the assertions and for decrypting encrypted assertions coming from the Identity Provider.

'signature' enable the signature support of SAML assertions. Enabling this option implies to have set the options 'certificate' and 'privateKey'.

'encryption' enable the encryption of SAML assertions (Attribute statements). Enabling this option implies to have set the option 'privateKey'.

11.7.7 UOC ←→Domain or data server (ex: OSS Analytics server)

Example of configuration for the OSSA (OSS Analytics) server case.

To configure SSL access from the UOC server to the OSSA server,

Edit the protocol, host and port as needed in this file:

<install_dir>/server/public/addons/plugins/ossa/config.json

Signed certificate case:

Add the name of a certification authority certificate file, if needed. This file is obtained from you signing provider and must be copied in:

<install_dir>/server/public/ssl

```
...
  "servers": {
   "ossa" : {
        "protocol": "https",
        "host": "ossa_server_system_full_hostname_or_ip_address",
        "port": "8443",
        "ssl": {
          "caCertFile": "ossa_server_ca.crt"
        }
    }
  },
...
```

Self signed certificate case:

Add the strictSSL parameter to disable strict certificate checking.

```
...
  "server": {
   “ossa” : {
        "protocol": "https",
        "host": "ossa_server_system_full_hostname_or_ip_address",
        "port": "8443",
        "ssl": {
            "strictSSL": false
        }
    }
    },
...
```

11.8 Logging

The UOC Server has customizable logger to ease troubleshooting of the UOC Platform.

See 12.1 Logging for detailed information to tune correctly the logs.

***Important**: These logs are only accessible to platform administrator for troubleshooting reason and may content sensitive information related to the privacy, especially if the debug level is enabled.*

11.9 Auditing

The UOC server logs special acitvities and access to resources to ease auditing. It is recommended to tune correctly the audit log policy (file size, rollup files,...) to fit your needs.

The UOC platform provides several security audit logs:

- Sessions (login. Logout)
- Resources (accesses and operations)

***Important**: These logs are only accessible to platform administrator and may content sensitive information related to the privacy (IP address...)*

11.9.1 Sessions

Log all information related to the user session (login, logout…) are stored in a log file in the server at **<install_data_dir>/logs/sessions.log**

It is possible to see user id, date/time, and IP address to ease the audit.

Example of sessions audit:

```
..
2015-05-09 16:23:46.692 - DUPONT3 - admin has logged out (127.0.0.1)
2015-05-09 16:23:53.796 - DUPONT3 - operator_l1 has logged in (127.0.0.1)
2015-05-09 16:24:40.998 - DUPONT3  - Paul has logged out (127.0.0.1)
2015-05-09 16:24:45.886 - DUPONT3 - admin has logged in (127.0.0.1)
2015-05-09 17:08:45.040 - DUPONT3 – Lilian  has logged in (127.0.0.1)
...
```

It is possible to refine the logging policy for this sessions logs customizing the “session-logger “ appender in the file :
<install_data_dir>/server/public/conf/log4js.json

Only log level information is supported but you can customize the format on the line (layout pattern).

Default logging policy:

```
{
   "type": "file",
   "filename": "logs/sessions.log",
        "level": "INFO",
        "maxLogSize": 2048000,
        "backups": 3,
   "category": "sessions-logger",
  "layout": {
    "type": "pattern",
    "pattern": "%d{ISO8601} - %h - %m"
  }
```

11.9.2 Resource access

Log all resource accesses, operations, applications and data requests. They are stored in a log file in the server at **<install_data_dir>/logs/security.log**

It tracks all actions done by a user on resources with date/time, IP address, user id ... to ease the audit.

Only log level information is supported but you can customize the format on the line (layout pattern).

Example of security audit:

..

2015-05-02 18:43:33.970 - DUPONT3 - (Node Server) : WIDGET : Browse all widgets (path:client/addons//vodafone/widgets)

2015-05-02 18:43:34.094 - DUPONT3 - (Node Server) : PLUGIN : Browse workspaces for plugin cea

2015-05-02 18:43:34.095 - DUPONT3 - (Node Server) : PLUGIN : Browse views for plugin cea

2015-05-02 18:43:57.109 - DUPONT3 - User: admin (127.0.0.1) : WORKSPACE : Browse all workspaces

2015-05-02 18:43:57.177 - DUPONT3 - (Node Server) : PLUGIN : Browse all plugins (path:server/addons/plugins/)

2015-05-02 18:43:57.181 - DUPONT3 - User: admin (127.0.0.1) : CATEGORY : Browse all categories defined for workspaces

2015-05-02 18:44:06.390 - DUPONT3 - User: admin (127.0.0.1) : WORKSPACE : Workspace DataFormatting_TimeKpiSeries has been opened

2015-05-02 18:44:06.437 - DUPONT3 - User: admin (127.0.0.1) : VIEW : View DataFormat_TimeKpi has been opened

2015-05-02 18:44:06.498 - DUPONT3 - User: admin (127.0.0.1) : LAYOUT : Layout layout-1-1-1 has been accessed

2015-05-02 18:44:06.536 - DUPONT3 - User: admin (127.0.0.1) : WIDGET : Widget hp-time-selector has been accessed

2015-05-02 18:44:31.250 - DUPONT3 - User: admin (127.0.0.1) : PLUGIN : Get data for plugin ossa with the url http://abc.hp.com:8080/ossa/packages/MBBQOE_Trial/facts/volume_up_sum/volume_down_sum/dims/BRAND/timewindow/1405505700000/1405689300000?b=1&granularity=15

2015-05-02 18:54:29.892 - DUPONT3 - User: admin (127.0.0.1) : PLUGIN : Get data for plugin ossa with the url http://abc.hp.com:8080/ossa/packages/MBBQOE_Trial/facts/volume_up_sum/volume_down_sum/dims/BRAND/timewindow/1405505700000/1405689300000?b=1&granularity=15&top=5

...

It is possible to refine the logging policy for this sessions logs customizing the “session-logger “ appender in the file :
<install_data_dir>/server/public/conf/log4js.json

Only log level information is supported but you can customize the format on the line (layout pattern).

Default logging policy:

```
{
    "type": "file",
    "filename": "logs/security.log",
    "level": "INFO",
    "maxLogSize": 2048000,
    "backups": 3,
    "category": "security-logger",
     "layout": {
      "type": "pattern",
      "pattern": "%d{ISO8601} - %h - %m"
}
```

Chapter 12

Troubleshooting

12.1 Logging

The UOC Server has customizable logger to ease troubleshooting of the UOC Platform.

***Important**: These logs are only accessible to platform administrator for troubleshooting reason and may content sensitive information related to the privacy, especially if the debug level is enabled.*

12.1.1 Server logs

All log for the UOC server can be found under **<install_data_dir>/logs/server.log**

It is possible to refine the logging policy for this server logs customizing the “server-logger “ appender in the file :
<install_data_dir>/server/public/conf/log4js.json

This log file support multiple level of log (info, warning, error, warning, debug). Default level is warning and you and customize the format on the line (layout pattern).

Default logging policy:

```
{
  "type": "file",
  "filename": "logs/server.log",
      "level": "WARN",
      "maxLogSize": 2048000,
      "backups": 3,
  "category": "server-logger"
}
```

12.1.2 Http requests

Log all http requests done by clients (web browsers) are logged in the UOC server in the logging directory.

You can find this log under **<install_data_dir>/logs/http.log**

It is possible to refine the logging policy for this server logs customizing the "server-logger " appender in the file :
<install_data_dir>/server/public/conf/log4js.json

Default log level is warning and customize the format on the line (layout pattern).

Default logging policy:

```
{
    "type": "file",
    "filename": "logs/http.log",
        "level": "WARN",
        "maxLogSize": 2048000,
        "backups": 3,
    "category": "express-logger"
}
```

12.2 Web Browser Console

If you have issues with the web browser, please check carefully you are using a supported web navigator. HTML 5 is only supported in very last web browser versions.

There is no logging at the client side and no persistence of console messages. If the problem is persistant, it is recommended to help the support team by opening the web browser console and check any unexpected message (error, warning...). This console can be visible usually by pressing F12 (check your navigator documentation for details).

---

*If you notice unexpected behavior after changes done at the UOC Server, It is strongly recommended to **clear the web browser cache** and restart from a fresh web browser usage.*
