

3D Juump Infinite Administration Manual

[AM_EN] version 3.1

Introduction	7
Disclaimer.....	9
1 - Documentation coverage	9
2 - Your system security is your responsibility	10
3 - Documentation liability	10
4 - Third-party software	10
Overview.....	11
1 - Introduction.....	11
2 - Definitions	12
2.1 - Digital mock-up (DMU).....	12
2.2 - 3D Juump Infinite infrastructure	13
3 - Components.....	15
4 - DMU Flow	17
Roles.....	20
1 - ∞Directory.....	20
2 - ∞Server	21
3 - ∞Proxy.....	22
4 - ∞StaticProxy	23
Access-rights	25
1 - Overview	25
2 - Tags	25
2.1 - Users and teams.....	26
2.2 - Servers, builds and connections	26
3 - Rules.....	26
3.1 - User connection to a server	26
3.2 - User access to a build	26
3.3 - User assets administration.....	26
3.4 - User licence borrowing	26

3.5 - Proxy build replication	27
3.6 - Proxy assets synchronization	27
Installation	28
1 - Infinite services installation.....	28
1.1 - Requirements	28
1.2 - Third party service version	29
1.3 - Install scenarios	30
1.4 - Test the ∞Directory front-end web page.....	30
1.5 - Install form settings description	30
2 - LMX.....	33
2.1 - Run a LM-X server	33
2.2 - Configure the LM-X server	33
2.2.1 - Activate the pay-per-use usage database.....	33
2.2.2 - Install the license file	34
2.2.3 - Extract the ∞Directory keys.....	34
3 - OpenId Connect	34
Administration.....	37
1 - Web application interface	37
1.1 - Login.....	38
1.2 - Dashboard	38
1.3 - Team Management	39
1.3.1 - Users	39
1.3.2 - Teams.....	41
1.4 - Network.....	43
1.4.1 - Servers	43
1.4.2 - Connections.....	47
1.5 - Projects.....	49
1.5.1 - Create Projects	50
1.5.2 - Check a Project Status	51

1.5.3 - Request a Build	52
1.5.4 - Synchronize metadata	52
1.5.5 - Get descriptor.....	52
1.5.6 - Clean generation data	53
1.5.7 - List Builds.....	53
1.5.8 - Delete a Build	53
1.5.9 - Export a Build	53
1.5.10 - Delete Projects.....	53
1.6 - Settings.....	53
1.7 - Client Updates	54
1.7.1 - Register new Client Updates	55
2 - Error notification (Webhook).....	55
3 - Migration procedures	56
3.1 - Cluster migration.....	56
3.2 - Assets migration	56
Annexes	57
1 - Release notes	57
1.1 - 16/01/2024: version 3.1.16	57
1.2 - 08/12/2023: version 3.1.15	57
1.3 - 15/03/2021: version 3.1.14	58
1.4 - 21/02/2021: version 3.1.13	58
1.5 - 10/11/2020: version 3.1.12	58
1.6 - 07/07/2020: version 3.1.11	58
1.7 - 18/06/2020: version 3.1.10	58
1.8 - 02/06/2020: version 3.1.9	58
1.9 - 10/01/2020: version 3.1.8	59
1.10 - 17/12/2019: version 3.1.7.....	59
1.11 - 10/12/2019: version 3.1.6.....	59
1.12 - 07/11/2019: version 3.1.5.....	59

1.13 -	02/10/2019: version 3.1.4.....	59
1.14 -	22/08/2019: version 3.1.3.....	59
1.15 -	30/07/2019: version 3.1.2.....	60
1.16 -	18/07/2019: version 3.1.1.....	60
1.17 -	26/06/2019: version 3.1.0.....	60
1.18 -	25/02/2019: version 3.0.3.....	61
1.19 -	11/02/2019: version 3.0.2.....	61
1.20 -	12/09/2018: version 3.0.1.....	62
1.21 -	24/08/2018: version 3.0.0.....	62
2 -	Security guidelines	63
2.1 -	PostgreSQL.....	63
2.1.1 -	Recommended Version	63
2.1.2 -	Configuration guidelines	63
2.1.3 -	Known vulnerabilities	63
2.2 -	ElasticSearch	63
2.2.1 -	Recommended Version	63
2.2.2 -	Configuration guidelines	63
2.2.3 -	Known vulnerabilities	64
2.3 -	Apache CouchDB.....	64
2.3.1 -	Recommended Version	64
2.3.2 -	Configuration guidelines	64
2.3.3 -	Known vulnerabilities	64
2.4 -	Apache Lounge Web-server.....	64
2.4.1 -	Recommended Version	64
2.4.2 -	Known vulnerabilities	64
2.5 -	OpenSSL.....	64
2.5.1 -	Recommended Version	64
2.5.2 -	Known vulnerabilities	65
3 -	Third-party software licenses.....	65

- 4 - Range of use65
 - 4.1 - Minimum requirements.....65
 - 4.2 - Supported input formats66
 - 4.2.1 - Geometry66
 - 4.2.2 - Metadata.....66
 - 4.2.3 - Annotation67
 - 4.3 - Supported output formats67
 - 4.3.1 - Geometry67
 - 4.3.2 - Image67
 - 4.3.3 - Metadata.....67
 - 4.3.4 - Presentation68
 - 4.4 - Limits68
- 5 - Export control classification69

Introduction

This documentation is targeted to your system administrators and CAD database specialists.

- For the *system administrator*, this documentation describes the installation and administration of 3D Juump Infinite back-end servers and their underlying third party components. The main features of the administration web interface manual are also presented to cover most of the administration use cases: create users, groups and configure the generation of the data sets, setup replication.
- For the *CAD database specialist*, the data set generation procedure is detailed to let you build your own data from your information system back-ends.

This documentation covers the installation of the 3D Juump Infinite server components:

- ∞Directory, connecting all the components of the 3D Juump Infinite; with the web administration interface;
- ∞Server, in charge of the compilation, optimization and packaging of the data sets;
- ∞Proxy, a proximity relay for data set broadcasting;
- ∞StaticProxy, a stub server for broadcasting data set from packaged files;

An Administration chapter is provided to explain the security policy and describe some maintenance points.

The last part is dedicated to 3D Juump client application deployment:

- 3D Juump Infinite client application installation
- Installer customization

For a primer installation, we suggest to start with the setup procedure of an ∞Directory and an ∞Server installations, validate it with the web administration interface. Then, if you want to construct your own data set, read the chapter dedicated to the data set generation procedure. Eventually, read the last chapter to deploy the client applications on your end-user host machines. The authentication of the users on the 3djuump infinite system is forwarded to an openid connect authentication server (Authenticator). The ∞Directory does not hold any authentication mechanism, but defers such authentication. There are public servers that can provide such an authentication mechanism, for example auth0 (<https://auth0.com/>). The system administrator may choose an open-source solution hosted on its own server infrastructure such as keycloak (<https://www.keycloak.org/>).

Disclaimer

1 - Documentation coverage

The information outlined in this documentation is intended to be used for the following purposes:

- Installation of 3D Juump Infinite back-end and front-end on Microsoft Windows and Linux;
- Creation of a specialized 3D Juump Infinite data sets specific to customer needs;
- Diffusion of 3D Juump Infinite data sets by 3D Juump Infinite back-end;
- Administration of 3D Juump Infinite users, groups and components.

The developed works define any piece of software you are developing with the 3D Juump Infinite software and data sets which have been generated by 3D Juump Infinite software. REAL FUSIO shall not be liable for any damage or injury arising out of any person or entity about these developed works; and the access or inability to access to the data sets built and served by the 3D Juump Infinite server infrastructure.

The developed works with 3D Juump Infinite software shall conform to design and implementation guidelines and restrictions described in the documentation. The software functions available for development are documented. Undocumented functionality should not be utilized without REAL FUSIO express consent.

2 - Your system security is your responsibility

You are responsible for the security of your system.

The developed works on the 3D Juump Infinite server should neither compromise data integrity; nor security of data and applications. The software accesses should be enforced by the 3D Juump Infinite server software proper configuration, third party software administration and the developed works. Your system administrator should read all documentations provided with this product to fully understand the available features.

3 - Documentation liability

The source code and template examples provided thorough this manual are only intended for educational purposes and are not intended to be a substitute to your developed works to be customized to comply with your security, performance or robustness requirements.

Although reasonable effort is made to ensure that the information in our documentation is complete and accurate at the time of release, REAL FUSIO cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in such documentation may be incorporated in future versions.

4 - Third-party software

The following software items are provided along with 3D Juump Infinite installers and scripts:

- Executables and plugins located in *third-party* subfolder (PostgreSQL, CouchDB, ElasticSearch, Apache, PHP, ...)
- Libraries and their dependencies (enumerated in licensing document)

They are provided as facility tools for educational and illustration purposes only. You are responsible for their proper configurations and usage in compliance with your enterprise software deployment & security policies and with the software license of each third-party components.

Although reasonable effort is made to ensure that these software items are up to date at the time of release, REAL FUSIO cannot assume responsibility for any defects or flaws caused by these third-party software items. Updated versions might be incorporated in future versions.

Overview

1 - Introduction

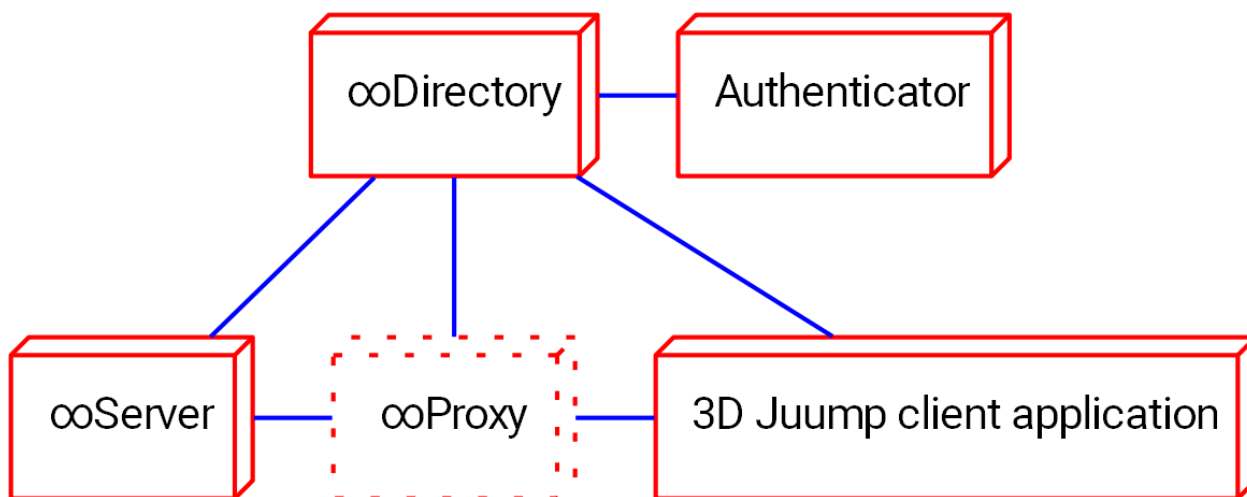
3D Juump Infinite is a software suite allowing the users to browse an entire DMU¹ in 3D on a standard PC².

It relies on a publication server (or **∞Server**), responsible for the preparation and optimization of the CAD source data, a network of relays (or **∞Proxy**) for broadcasting, and a central management server (or **∞Directory**).

On the end user side, the 3D Juump Infinite DMU client application browser is simply called **3D Juump Infinite**.

¹ Digital Mock-Up

² Personal Computer



3D Juump servers and client

2 - Definitions

In order to explain the design of 3D Juump Infinite, it is mandatory to introduce several keywords related to the DMU and CAD product.

2.1 - Digital mock-up (DMU)

The digital mock-up (DMU) is composed of multiple elements:

Digital mock-up

A mock-up is a partial or complete representation of a system in order to preview, test or validate its aspects or its behavior. A digital mock-up is built from computer files storing a tree of three dimension geometries, displayed with a 3D rendering software, in our case, 3D Juump.

Part

A model or template, be it an assembly or a single element. This part is referenced by the digital mock-up but is not present (*instanced*) within it. It is not localized in the digital mock-up. Ex: a wheel.

Part Instance

One instance of a *part*, be it an assembly or a single element. This instance has a 3D representation located in the digital mock-up. Ex: the front-right wheel.

Product Structure

A hierarchy of *part instances*. Ex: a car with four wheels.

Part Number

The unique identifier of a *part*.

Metadata

Any textual or numeric information decorating a *part* or *part instance*, usually presented as a *key=value* pair. Ex: *provider='WheelEx Inc.'*.

Annotation

Any textual or graphical information decorating a *part* or *part instance*, represented in 3D as floating labels. Ex: contextual directions for assembly, tolerancing, etc.

Effectivity

A global parameter or option that changes the way the DMU should be assembled. Ex: What is the car drive-train? Is this car a 4-wheel or a 2-wheel drive vehicle?

Configuration

A set of effectivities. A digital mock-up in a given configuration is also called a configured digital mock-up. Ex: a 4-wheel, metallic paint, 150 hp engine car.

2.2 - 3D Juump Infinite infrastructure

Other keywords are specific to the 3D Juump Infinite infrastructure:

Project

An incremental set of data describing a DMU. It usually corresponds to a product (or product-line) level assembly. For instance, a vehicle manufacturer would probably opt for one project per car model.

Build

An optimized packaged DMU snapshot ready for publication. The DMU processed by 3D Juump Infinite back-end components and served to 3D Juump Infinite front-end client applications.

Connector

A piece of software that feeds a 3D Juump Infinite *project* with data describing the DMU.

DMU data source provider

Any component of your *information system* able to publish DMU data encompassing the part geometry, metadata or effectivity. These components could be a conjunction of your

CMS³, CRM⁴, your ERP⁵, your PDM⁶, your PLM⁷ or simply a file system folder containing all your geometry part files extracted from your CAD⁸ software.

Document

A document is a JSON⁹ exchange file used by the *Connector* and the 3D Juump Infinite back-end to transmit and store data bound to DMU *build* generation. JSON is an open standard human-readable text used to transmit data objects consisting of attribute-value pairs. Thus, this format is also used by 3D Juump Infinite third-party software and the 3D Juump Infinite front-end to exchange data.

DMU flow

A chain of your enterprise components and 3D Juump Infinite back-end components, in charge of DMU extraction from *DMU data source providers*, processing & broadcasting of DMU *builds*.

User

A person that uses 3D Juump Infinite, either through its web administration interface, through a 3D Juump Infinite client application or through one of the offered API. All users must be properly authenticated before being authorized to use 3D Juump Infinite.

Authenticator

A third-party server in charge with the user authentication. 3D Juump Infinite relies on OpenID Connect identity layer to delegate authentication.

Administrator

A user which can logon on the ∞Directory web administration interface. He is able to configure the *DMU flows*, to trigger the generation of DMU *builds* and create *users* and *teams*.

Tag

A keyword used to decorate a *build*, a *user* or a server/proxy component and which defines the access rights to the DMU.

³ Content Management System

⁴ Customer Relationship Management

⁵ Enterprise Resource Planning

⁶ Product Data Management

⁷ Product Lifecycle Management

⁸ Computer Aided Design

⁹ JavaScript Object Notation

Team

A set of users. It mainly acts as a helper concept that applies a common list of tags to its users.

Asset

Any 3D Juump Infinite client application setting (bookmarks, visibility layers, export configurations...) which can be created, manipulated and shared amongst 3D Juump Infinite *users* thanks to 3D Juump Infinite back-end components.

3 - Components

3D Juump Infinite is composed of several software entities.

∞Server

An ∞Server is in charge of the optimization and the packaging of the DMUs (or *projects*) into distributable *builds*. To let *connectors* declare DMUs, the ∞Server publishes a *connector API*.

∞Proxy

An ∞Proxy acts as a proximity relay for DMU broadcasting. It replicates *builds* from ∞Servers or other ∞Proxies. Since an ∞Server is itself an ∞Proxy, the deployment of an ∞Proxy is *not necessary* if your information system and network do not require such DMU broadcasting.

∞Directory

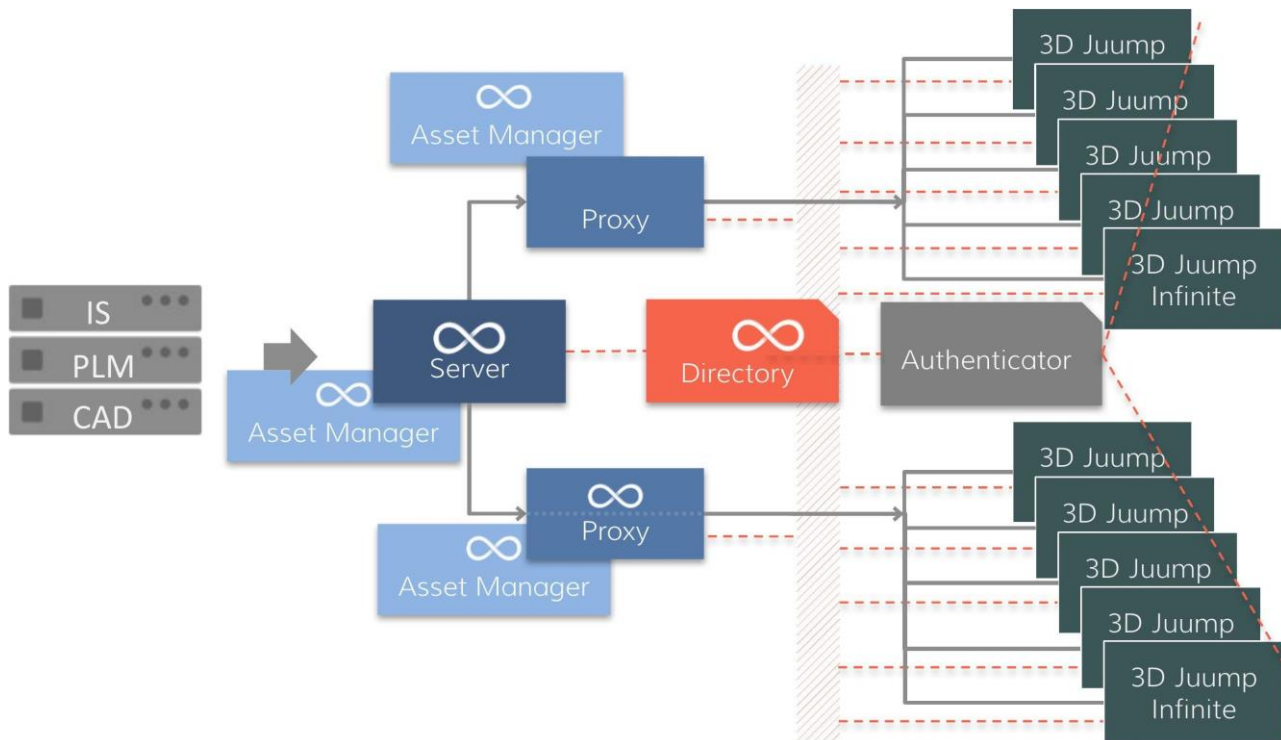
The ∞Directory is in charge of the security management. It monitors and defines the whole 3D Juump Infinite network and is responsible for access-rights management. This management is accessible via a *web administration interface* and programmable via the *directory API*.

3D Juump Infinite client application

The 3D Juump Infinite client application is the DMU browser itself, called *3D Juump Infinite*, when this name is unambiguous.

Web client or Native client

The 3D Juump Infinite client application comes in two flavors. One is the legacy native application (called "Native client") and the other is the new web-based application (called "Web client").



Overview

The previously presented software relies on several third-party components including databases and servers. In particular:

- a [PostgreSQL](#) service,
- a [CouchDB](#) service,
- an [ElasticSearch](#) service,
- an [Apache](#) HTTP service,
- a [LM-X](#) service.

PostgreSQL (or “Postgres”) is an SQL object-relational database management system (ORDBMS). It is used by 3D Juump Infinite ∞Directory, ∞Server and ∞Proxy as underlying database engine.

CouchDB is a document-oriented NoSQL¹⁰ database that uses JSON to store data, JavaScript as its query language using MapReduce, and HTTP for an API¹¹. It is used to store 3D Juump Infinite assets.

Elasticsearch is a distributed, multitenant-capable full-text search server with a RESTful web interface and schema-free JSON documents. Elasticsearch is built on top of on Apache Lucene, developed in Java and is released as open source under the terms of the Apache License. It provides full-text search capabilities to the 3D Juump Infinite client application.

¹⁰ No SQL, i.e. schema-less and non transactional

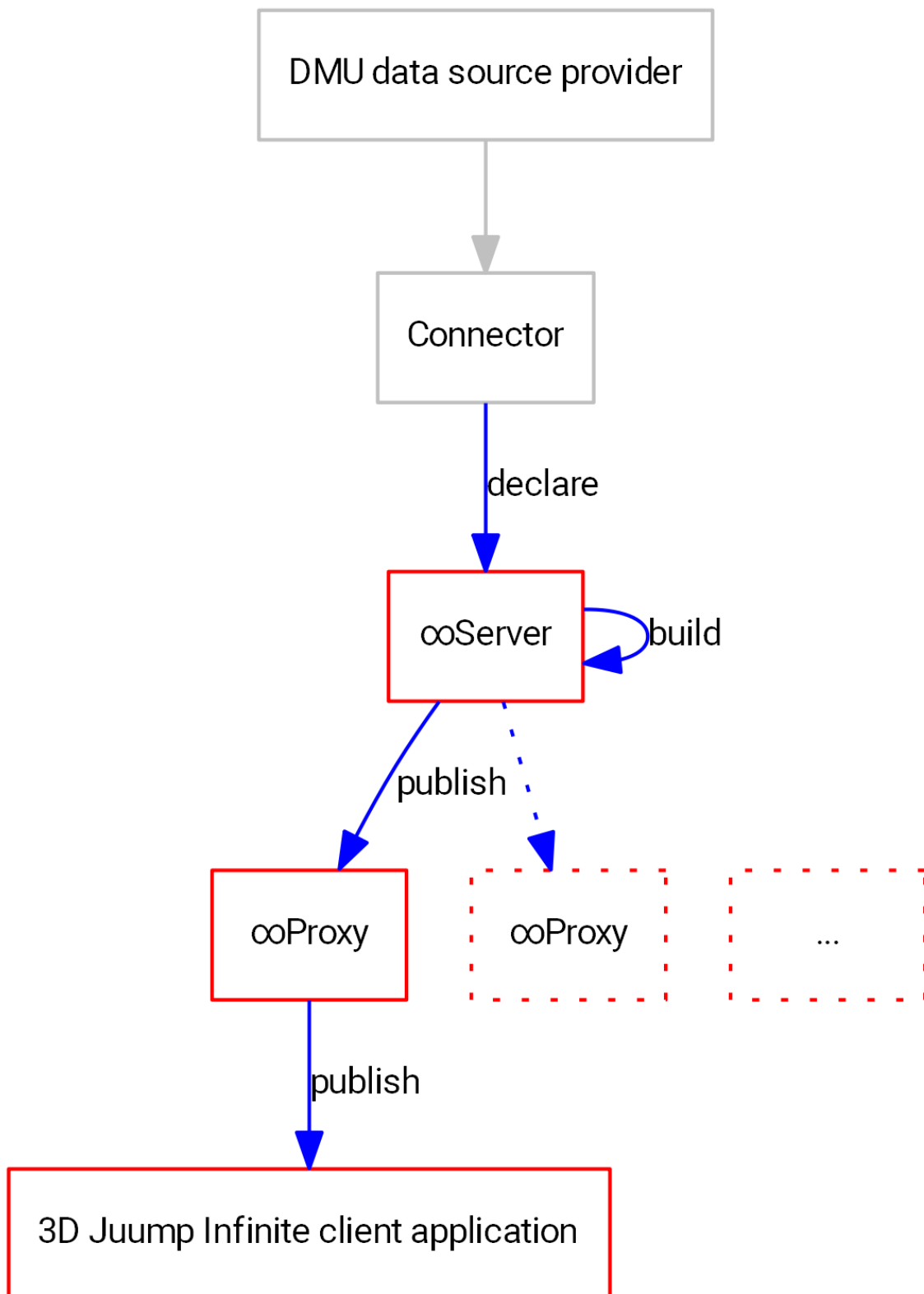
¹¹ Quoted from [CouchDB front page](#)

LM-X is a software licensing solution.

4 - DMU Flow

From the enterprise CAD data source to the 3D Juump Infinite client application, the DMU follows several steps:

- First, your enterprise defined **Connector**, bound to your DMU data source providers, incrementally declares the product structure and the associated metadata, effectivities and configurations to a *project* hosted on an ∞Server.
- Once the declared product structure is coherent, the connector triggers a *build*.
- The ∞**Server** compiles the product structure and the associated data into a hierarchical database where every leaf points to a geometry.
- Then the ∞Server incrementally pulls the geometry files from the connector and builds an optimized package.
- This *build* is then published on the ∞**Proxies** associated to this ∞Server and every connected ∞Proxy then starts to replicate the build locally, in order to obtain a cascade publication.



DMU flow

The Connector must be specified and realized upon your requirements. 3D Juump Infinite only provides interfaces and helpers to feed the ∞Server with such data. In the figure, the enterprise DMU data source providers and your Connector are in gray.

During this process, the ∞**Directory** keeps in touch with every server, be it the ∞Server or the network of relay ∞Proxies. Thus, it is able to handle 3D Juump Infinite client applications connection requests by routing them to the proper ∞Proxy.

For the sake of simplicity, we have described *one* DMU flow. 3D Juump Infinite let you define *several* DMU flows built upon your combination of Connectors bound to your DMU data source providers, attached to several ∞Servers and ∞Proxies.

Roles

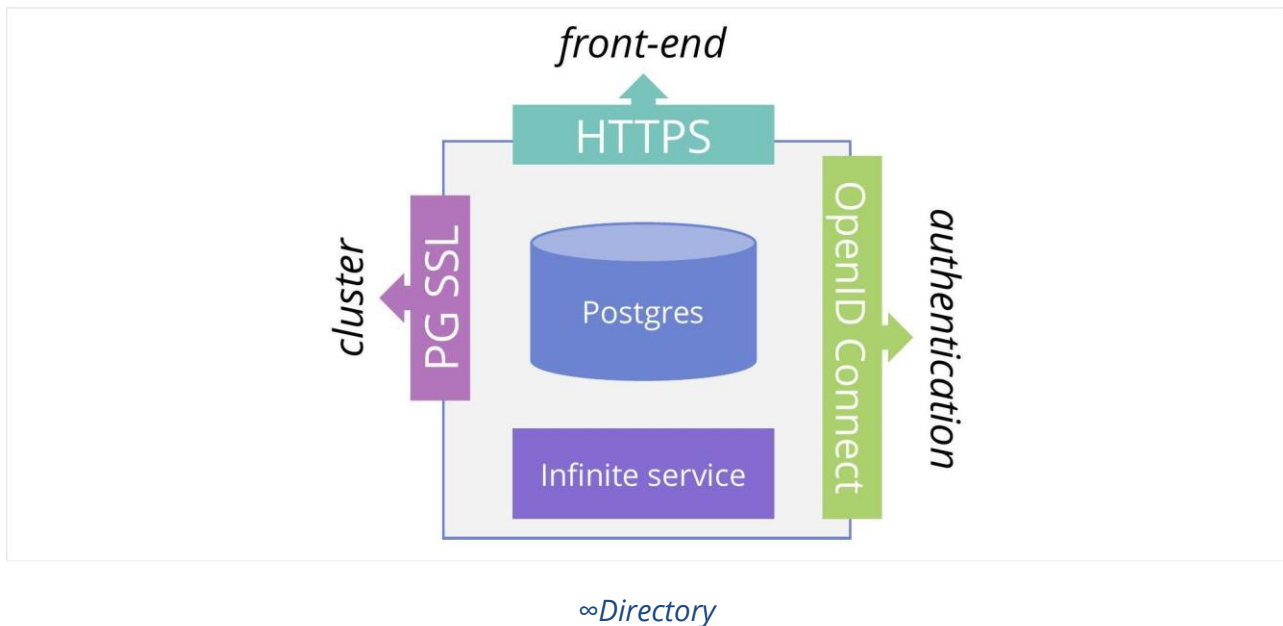
1 - ∞Directory

The ∞Directory is in charge of the security management. It monitors the whole 3D Juump Infinite network, from the ∞Servers to the relay ∞Proxies and up to the 3D Juump Infinite client applications. None of these software entities is able to work without the explicit authorization of the ∞Directory. It is also responsible for access-rights management, though it delegates user authentication to a third-party identity service.

The ∞Directory relies on several software components:

- a LM-X server,
- a PostgreSQL server,
- an Apache HTTP server,
- a dedicated service with the provided installers.

The ∞Directory is operated thanks to a web-application (or *front-end*) built upon a web-based API (or *Directory API*) hosted on the HTTP server. The use of this web interface is described in the administration chapter. The API is described in a dedicated document.



The ∞Directory delegates user authentication through OpenID Connect identity layer. Only authenticated users with proper access rights can operate the *Directory API* and the *front-end*, or open a DMU through one of the 3D Juump Infinite client applications and API.

Databases located on the ∞Directory are not directly visible to client-side users. Instead, several API are available and only accessible to authenticated users with proper access rights. Cluster-side server-to-server communication relies on PostgreSQL PG-SSL layer.

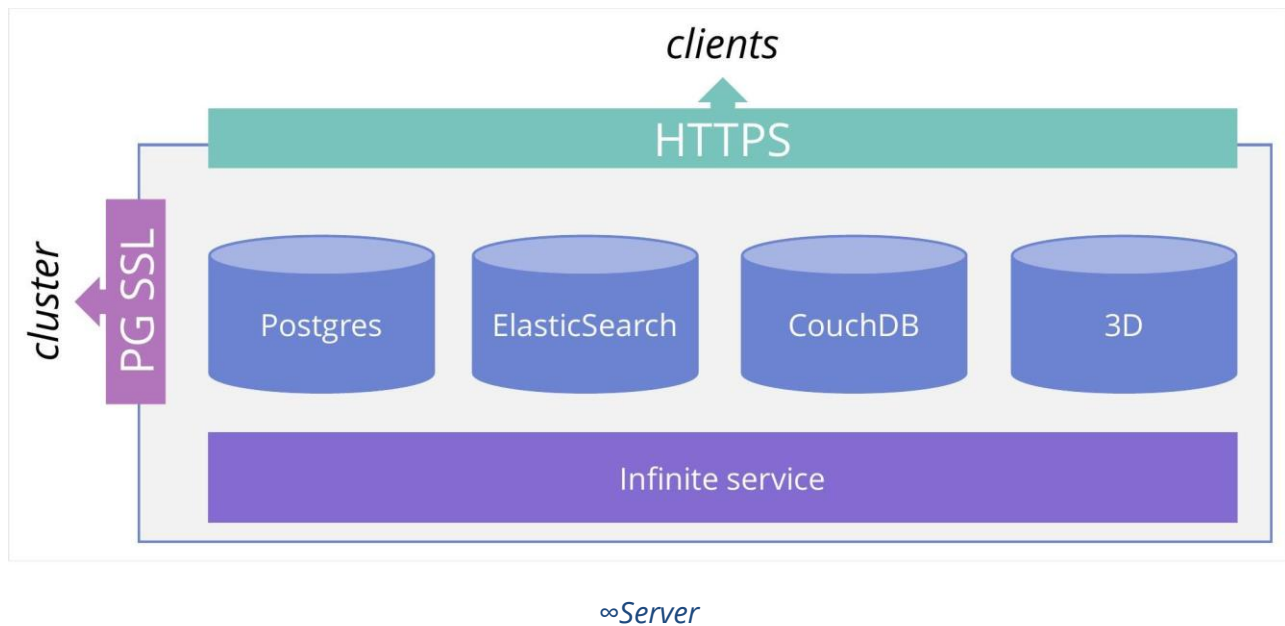
2 - ∞Server

The ∞Server is in charge of the compilation, optimization and packaging of the DMUs. It offers a web interface allowing the enterprise connectors to describe the DMUs (or *projects*). The ∞Server is also a ∞StaticProxy, therefore it publishes *builds* that may be browsed with the 3D Juump Infinite client applications. If you want to replicate the *builds* to multiple sites, you can also install additional ∞Proxies to broadcast the *builds* through your network.

The ∞Server relies on several software components:

- a PostgreSQL server,
- a CouchDB server,
- an ElasticSearch server,
- an Apache HTTP server,
- a dedicated service.

Most operations related to the ∞Server administration are controlled by the ∞Directory front-end.



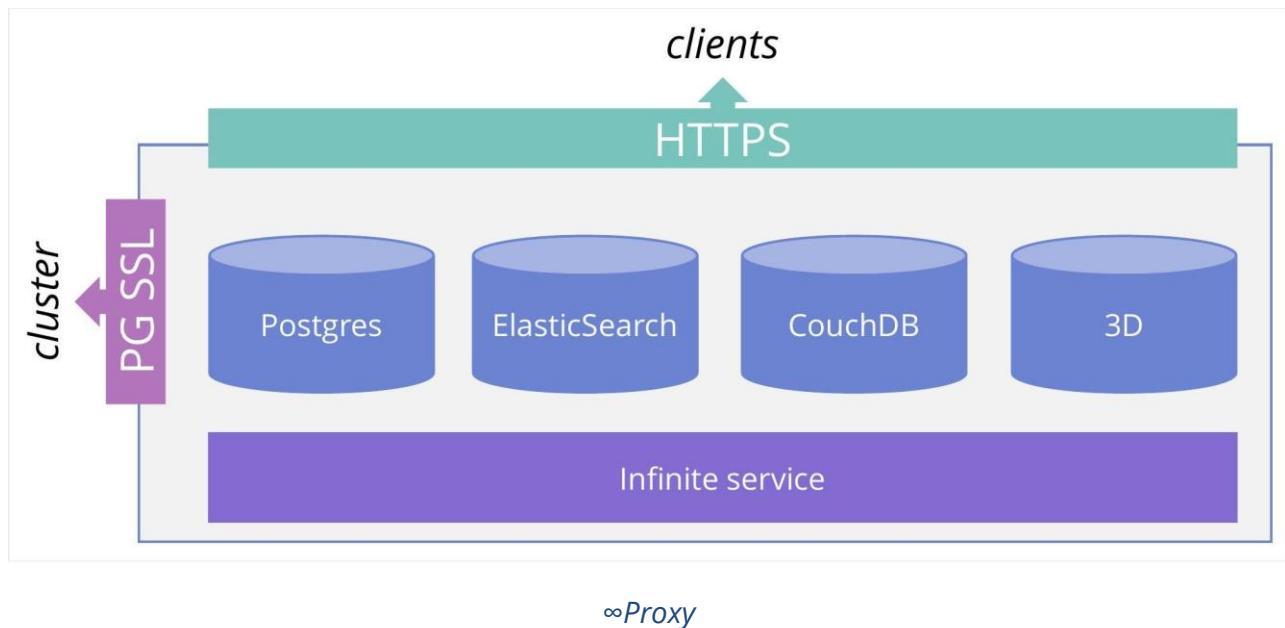
Databases located on the ∞Server are not directly visible to client-side users. Instead, several API are available and only accessible to authenticated users with proper access rights. Cluster-side server-to-server communication relies on PostgreSQL PG-SSL layer.

3 - ∞Proxy

Under the surveillance of the ∞Directory, a ∞Proxy acts as a proximity relay for DMU broadcasting. It replicates *builds* from ∞Servers or other ∞Proxies. 3D Juump Infinite client applications connect to ∞Proxies in order to browse DMUs. A ∞Proxy is not compulsory to distribute DMU data. For a single point distribution, you can use a ∞Server, which is also a ∞StaticProxy itself, to deliver the DMU to the 3D Juump Infinite client applications.

The ∞Proxy relies on several software components:

- a PostgreSQL server,
- a CouchDB server,
- an ElasticSearch server,
- an Apache HTTP server,
- a dedicated service.



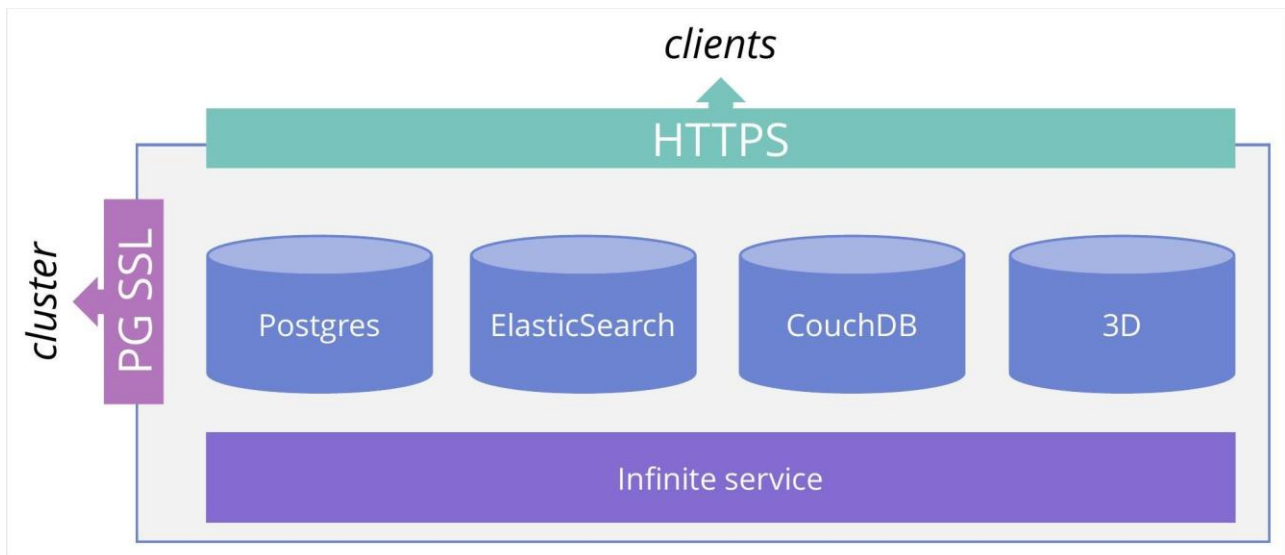
Databases located on the ∞Proxy are not directly visible to client-side users. Instead, several API are available and only accessible to authenticated users with proper access rights. Cluster-side server-to-server communication relies on PostgreSQL PG-SSL layer.

4 - ∞StaticProxy

Under the surveillance of the ∞Directory, a ∞StaticProxy acts as a stub server for DMU broadcasting. It is not connected to any streaming data but rather publishes *builds* extracted from DMU files (or *evojuumps*). 3D Juump Infinite client applications connect to ∞StaticProxies just as they would ∞Servers or ∞Proxies.

The ∞StaticProxy relies on several software components:

- a PostgreSQL server,
- a CouchDB server,
- an ElasticSearch server,
- an Apache HTTP server,
- a dedicated service.



∞StaticProxy

Databases located on the ∞StaticProxy are not directly visible to client-side users. Instead, several API are available and only accessible to authenticated users with proper access rights. Cluster-side server-to-server communication relies on PostgreSQL PG-SSL layer.

Access-rights

This chapter explains the access rights policy enforced at the ∞Directory level.

1 - Overview

3D Juump Infinite restricts the access to:

- the different APIs (Directory API, Connector API, Client API...),
- the different nodes of the cluster,
- the DMU builds,
- certain features exploiting the DMUs (export, download...),
- the assets,
- the licenses.

Most of these access limitations are configured thanks to tags. For instance, a user will only have access to the builds that match his tags.

2 - Tags

Tags are keywords decorating an actor of 3D Juump Infinite security model:

- users,

- teams,
- builds,
- servers,
- connections.

A tag is of the form "[namespace:]keyword". The *keyword* itself is user-defined and cannot contain a colon (:). The *namespace*, when presents, indicates the target feature of the tag and is limited to the values listed in the following chapters.

2.1 - Users and teams

Users and teams can bear any number of *namespaced* tags. A user also automatically inherits the tags from the teams he belongs to. For instance, a given user in the "guest" team will automatically inherits the tags from the this team in addition to his own personal tags.

2.2 - Servers, builds and connections

Servers, builds and connections can bear any number of general tags (no namespace).

3 - Rules

Access rights management is enforced by applying a set of strict tag-matching rules.

3.1 - User connection to a server

A user can only connect to a server/proxy if he bears all the matching tags: for each "*keyword*" tag of the target server/proxy, the user must have a matching "**connect:keyword**" tag.

3.2 - User access to a build

A user can access a feature of a build if he bears all the matching tags.

- Visualisation feature: for each "*keyword*" tag of the target build, the user must have a matching "**view:keyword**" tag.
- Screenshot feature: for each "*keyword*" tag of the target build, the user must have a matching "**export2D:keyword**" tag.
- Export feature: for each "*keyword*" tag of the target build, the user must have a matching "**export3D:keyword**" tag.
- Download feature: for each "*keyword*" tag of the target build, the user must have a matching "**download:keyword**" tag.

3.3 - User assets administration

A user can administrate assets pertaining to a project only if he bears a matching "**asset:admin._prj__xxx_**" where *_prj__xxx_* is the project id.

3.4 - User licence borrowing

A user can borrow a licence for offline use only if he bears the "**lmx:allowborrow**" tag.

3.5 - Proxy build replication

A proxy can only replicate a build along a server connection if either:

- the connection bears no tag at all (unlimited replication) OR
- the connection bears all the tags of the target build.

All the assets of the projects the replicated builds belong to will also be synchronized.

3.6 - Proxy assets synchronization

A proxy can only synchronize the assets of a project ("_prj_xxx") along an asset connection if either:

- the connection bears no tag at all (unlimited synchronization) OR
- the connection bears a tag with the project name ("_prj_xxx") without a namespace.

Asset replication will be set up only if the matching projects have at least one build.

Installation

1 - Infinite services installation

This chapter will present how to install an ∞Directory, ∞Server, ∞Proxy or ∞StaticProxy.

1.1 - Requirements

Each 3D Juump Infinite machine must be able to contact (e.g. ping) the ∞Directory host, which is the central point of the 3D Juump Infinite architecture. Each ∞Proxy must be able to contact ∞Server or ∞Proxy that will provide data to him. Your network administrator must provide you with one or several machine(s) which can be connected (in the network sense).

Before starting installation you should retrieve :

- The software delivery package containing Windows installers, install scripts, configuration files and this documentation
- At least one machine to host your ∞Directory / ∞Server
- **Administrator** rights on all machines
- Python 2.7 or 3.6
- For ∞Directory, an html 5 compliant browser
- For linux, an internet access to REAL FUSIO apt repository from all machines
- Optionnaly, a SSL certificate, such a certificate may be built

- Setup or use an existing LMX server, configured with REAL FUSIO vendor extension (such a server can be installed on the same host than ∞Directory [LMX])



YOU CANNOT INSTALL an ∞Server and an ∞Proxy or an ∞StaticProxy on the same machine.



All logins/passwords must be limited to alphanumerical characters and these separators: space ' ', minus '-', underscore '_' and point '.'¹².



You might require to temporary disable your antivirus during the installation procedure. We need to modify the windows firewall rules during installation. Some antivirus software (e.g. McAfee) may detect this as a potential threat.

Notes : All credentials will be stored in an encrypted form in order to ensure that only a trusted user of the machine can access this information.

The ∞Directory, ∞Server, ∞Proxy and ∞StaticProxy software run on any of the following operating systems:

- Microsoft Windows 7 and above, 64-bit version
- Linux Debian stretch, AMD64
- Linux Ubuntu 16.04 LTS (xenial) AMD64

Minimum hardware requirements are:

- Quad-core processor
- For the ∞Directory, ∞Proxy and ∞StaticProxy : 8GB of RAM
- For the ∞Server : 4GB of RAM by CPU Core (A Quad-core processor should have at least 16 GB of RAM).
- 1GB disk space for binaries + sufficient disk space for data (depending on your data sources)
- High-speed hard disk drive highly recommended

1.2 - Third party service version

3DJuump Infinite installs and uses third party services. Here is a table that lists expected version for these components.

Third Party Software	∞Directory	∞Server / ∞Proxy / ∞StaticProxy
LMX Server	4.8.13	-
PostgreSQL	10 (10.7+)	10 (10.7+)
ElasticSearch	-	6 (6.6.0+)
Java JRE Server	-	8 (8u222+)
CouchDB	-	1.6.1
Apache	2.4 (2.4.38+)	2.4 (2.4.38+)

¹² For information, the matching regex is "[a-zA-Z0-9_-.]+".

1.3 - Install scenarios

Depending on your needs we provide the following install scripts for the given install scenarios :

- `directory` : will install an ∞Directory on the machine.
- `server` : will install an ∞Server on the machine.
- `proxy` : will install an ∞Proxy on the machine.
- `static_proxy` : will install an ∞StaticProxy on the machine.
- `directory_and_server` : will install an ∞Directory and an ∞Server on the machine. It is recommended to use this scenario if you have only one machine available.
- `directory_and_static_proxy` : will install an ∞Directory and an ∞StaticProxy on the machine. It is recommended to use this scenario if you have only one machine available.

The folder `install_form` of the software delivery package contains the different install scripts for each scenario (e.g `directory_install.py`, `directory_and_server_install.py`, etc ...) and a corresponding form (e.g `directory_form.csv`, `directory_and_server_form.csv`, etc ...) that lists the installation settings. First edit the corresponding `xxxxxxx_form.csv` and fill all mandatory entries. You are free to update entries that are pre-filled. Optional entries should be left empty except if you have specific configuration needs. You will find a description of each entry in the following chapter [Install form settings description](#). Once you have finish editing csv file, open a terminal with **Administrator** rights and run the corresponding `xxxxxxx_install.py` python script. Then follow the instructions.

All install scripts support the following options :

- `-f` : force answer yes to all user input
- `-accept-eula` : automatically accept eula

Notes : You may provide your own SSL certicat and private key (in `x509` format without password protection) for the TLS communication. Please fill in `${hostname}` in the `xxxxxxx_form.csv` file. You may then proceed with the installation and override the autogenerated certificate and private key with your own files.

1.4 - Test the ∞Directory front-end web page

For any scenario involving the installation of an ∞Directory, once the installation is finished, the ∞Directory front-end administration page can be accessed through `https://directory_name:${apache_https_port}/directory` (see [Login in the web administration interface](#)).





The web administration interface front page cannot be displayed? Check if the ∞Directory service has started and is approved by your network firewall rules.




1.5 - Install form settings description

Install settings information :

- W : Windows only
- L : Linux only

- A : Applicable to all system
- O : Optional, the variable could be left blank, it has a default value

Setting name	Info	Description
<code>\${binary_basepath}</code>	W	Base folder where all software binaries will be installed if not overloaded by optional settings
<code>\${install_basepath}</code>	A	Base folder that will contain db and services data
<code>\${realfusio_appt}</code>	L	Url of apt repository, please contact your provider to get one
<code>\${hostname}</code>	AO	Fully Qualified Domain Name, used to configure Apache server and to generate SSL certificat if needed (Defaults to the machine hostname determined by the install script)
<code>\${generatewebdata}</code>	A	Set this variable to <i>true</i> to configure the ∞Server/∞Proxy as a data provider for the web api
<code>\${linux_oracle_java}</code>	L	Elasticsearch depends on Java. On Linux, there are 2 viable options to get java : OpenJDK (https://openjdk.java.net/) and Oracle Java (https://www.oracle.com/technetwork/java/index.html). Set this variable to <i>true</i> to use the Oracle version, but you will have to install it manually, and accept the Oracle Java terms and conditions during the installation.
<code>\${install_web_client}</code>	A	Set this variable to <i>true</i> to install the web api and the web client on your ∞Directory
<code>\${postgres_login}</code>	A	Postgres login for super user role used by the Infinite services to login to the PostgreSQL cluster
<code>\${postgres_password}</code>	A	Postgres password for super user role used by the Infinite services to login to the PostgreSQL cluster  this password should be strong, and should not contain blank characters.
<code>\${authconf_file}</code>	A	Path to a valid authentication json file. Requirements on such a file are listed in OpenId Connect
<code>\${ssl_verify_peer}</code>	A	Set this value to true and all connections between the servers (∞Directory, ∞Server, ∞Proxy, ∞StaticProxy) will check the legitimacy/validity of the TLS certificate during connections. Invalid certificates will prevent these elements from communicating. You will need to include the IPv4 address and the hostname of the server in each certificate. This setting is <i>false</i> by default.  Invalid certificates will prevent elements from communicating.
<code>\${lmx_server}</code>	A	Connection information to reach the LMX server, in LMX format : e.g. 6200@mylmxserver
<code>\${lmx_feature}</code>	A	LMX Directory feature that should be used by this ∞Directory, please contact your provider to get one
<code>\${apache_http_port}</code>	A	Port number for http protocol communication (usually 80)
<code>\${apache_https_port}</code>	A	Port number for https protocol communication (usually 443)
<code>\${postgres_port}</code>	A	Port number of PostgreSQL service (usually 5432)
<code>\${couchdb_port}</code>	A	Port number of CouchDB service (usually 5984)
<code>\${elastic_port}</code>	A	Port number of Elasticsearch service (usually 9200)
<code>\${directory_port}</code>	A	Port number on loopback for ∞Directory inter process communication (usually 2702)
<code>\${proxy_port}</code>	A	Port number on loopback for ∞Proxy/∞Server/∞StaticProxy inter process communication (usually 2703)
<code>\${apache_http_replication_port}</code>	A	Port number used on loopback to encapsulate CouchDB replication stream into an https stream (usually 8080)

<code>\${http_proxy}</code>	LO	Http forward proxying, if needed in your network to access the internet apt repository
<code>\${https_proxy}</code>	LO	Https forward proxying, if needed in your network to access the internet apt repository
<code>\${apache_log_folder}</code>	AO	Folder that will contain the Apache access and error logs (Defaults to : <code>\${install_basepath}/current/www/Logs</code>)
<code>\${ssl_folder}</code>	AO	Folder that will contain autogenerated certificat except if a specific one is specified (Defaults to : <code>\${install_basepath}/ssl</code>)
<code>\${privatekey_file}</code>	AO	SSL private key file. If empty a self-signed certificat will be generated (Defaults to : <code>\${ssl_folder}/private.key</code>)
<code>\${certificate_file}</code>	AO	SSL certificat file. If empty a self-signed certificat will be generated (Defaults to : <code>\${ssl_folder}/server.crt</code>)
<code>\${postgres_data_folder}</code>	AO	Folder that will contain the PostgreSQL cluster data (Defaults to : <code>\${install_basepath}/current/postgresql</code>)
<code>\${couchdb_data_folder}</code>	AO	Folder that will contain the CouchDB cluster data (Defaults to : <code>\${install_basepath}/current/couchdb/data</code>)
<code>\${couchdb_log_folder}</code>	AO	Folder that will contain the CouchDB log files (Defaults to : <code>\${install_basepath}/current/couchdb/Logs</code>)
<code>\${elastic_data_folder}</code>	AO	Folder that will contain the Elasticsearch node data (Defaults to : <code>\${install_basepath}/current/elasticsearch/data</code>)
<code>\${elastic_log_folder}</code>	AO	Folder that will contain the Elasticsearch log files (Defaults to : <code>\${install_basepath}/current/elasticsearch/Logs</code>)
<code>\${directory_log_folder}</code>	AO	Folder that will contain the ∞Directory service log files (Defaults to : <code>\${install_basepath}/current/service/Logs</code>)
<code>\${directory_webserver_folder}</code>	AO	Folder that will contain directory front-end files (Defaults to : <code>\${install_basepath}/current/www/directory</code>)  all files from the subfolder /public will be freely downloadable from <code>http://hostname:https_port/directory uri</code>
<code>\${directory_empty_root_folder}</code>	AO	Folder that will contain no files (Defaults to : <code>\${install_basepath}/current/www/empty</code>)  all files from this folder will be freely downloadable from <code>https://hostname:https_port/</code>
<code>\${directory_web_client_folder}</code>	AO	Folder that will contain the web client files (Defaults to : <code>\${install_basepath}/current/www/webclient</code>)  all files from this folder will be freely downloadable from <code>https://hostname:https_port/webclient uri</code>
<code>\${directory_fcgi_folder}</code>	AO	Empty Folder to serve the ∞Directory fcgi calls (Defaults to : <code>\${install_basepath}/current/www/directory_fcgi</code>)
<code>\${server_data_folder}</code>	AO	Folder that will contain the 3d data of the ∞Server service (Defaults to : <code>\${install_basepath}/current/service/3d</code>)
<code>\${server_log_folder}</code>	AO	Folder that will contain the ∞Server service log files (Defaults to : <code>\${install_basepath}/current/service/Logs</code>)
<code>\${server_webserver_folder}</code>	AO	Folder that will contain the web files of the ∞Server api (Defaults to : <code>\${install_basepath}/current/www/fileserver</code>)
<code>\${proxy_data_folder}</code>	AO	Folder that will contain the 3d data of the ∞Proxy service (Defaults to : <code>\${install_basepath}/current/service/3d</code>)
<code>\${proxy_log_folder}</code>	AO	Folder that will contain the ∞Proxy service log files (Defaults to : <code>\${install_basepath}/current/service/Logs</code>)

<code>\${proxy_webserver_folder}</code>	AO	Folder that will contain the web files of the ∞Proxy api (Defaults to : <code>\${install_basepath}/current/www/fileserver</code>)
<code>\${fcgi_ipc_folder}</code>	AO	Temporary folder used for fcgi inter-process communication (Defaults to : <code>\${install_basepath}/current/www/fcgi_ipc</code>)
<code>\${apache_folder}</code>	WO	Folder where the 3DJuump Infinite Apache Lounge binaries will be installed (Defaults to : <code>\${binary_basepath}/Apache</code>)
<code>\${postgres_folder}</code>	WO	Folder where the PostgreSQL binaries will be installed (Defaults to : <code>\${binary_basepath}/PostgreSQL</code>)
<code>\${directory_binary_folder}</code>	WO	Folder where the ∞Directory service binaries will be installed (Defaults to : <code>\${binary_basepath}/3DJuumpInfiniteDirectoryx64</code>)
<code>\${server_binary_folder}</code>	WO	Folder where the ∞Server service binaries will be installed (Defaults to : <code>\${binary_basepath}/3DJuumpInfiniteServerx64</code>)
<code>\${proxy_binary_folder}</code>	WO	Folder where the ∞Proxy/∞StaticProxy service binaries will be installed (Defaults to : <code>\${binary_basepath}/3DJuumpInfiniteProxyx64</code> or <code>\${binary_basepath}/3DJuumpInfiniteStaticProxyx64</code> depending on the scenario)
<code>\${infinite_postgres_plugins_folder}</code>	WO	Folder where the 3D Juump Infinite Postgres Plugins binaries will be installed (Defaults to : <code>\${binary_basepath}/3DJuumpInfinitePostgresPluginsx64</code>)
<code>\${couchdb_folder}</code>	WO	Folder where the CouchDB binaries will be installed (Defaults to : <code>\${binary_basepath}/CouchDB</code>)
<code>\${elastic_binary_folder}</code>	WO	Folder where the Elasticsearch binaries will be installed (Defaults to : <code>\${binary_basepath}/ElasticSearch</code>)
<code>\${java_binary_folder}</code>	WO	Folder where the Java binaries will be installed (Defaults to : <code>\${binary_basepath}/Java</code>)

2 - LMX

2.1 - Run a LM-X server

If you do not already have an available LM-X server (version 4.8.13), please run the corresponding installer (provided in the install package) and refer to the LM-X documentation. Provide the REAL FUSIO vendor extension file (*LibLmxvendor.so* on Linux or *LibLmxvendor.dll* on Windows) when required during the installation.

Note: it is not necessary that the LM-X server run on the same physical computer as the ∞Directory as long as it is visible (in a network sense) from both to the ∞Directory and the 3D Juump Infinite client application stations.

Remember the LM-X *license path* (see LM-X documentation) corresponding to your LM-X server (or LM-X cluster), usually something similar to "*6200@my-lmx-server*" (`${lmx_server}`).

2.2 - Configure the LM-X server

2.2.1 - Activate the pay-per-use usage database

In the LM-X configuration file (*lmx-serv.cfg*), make sure the *USAGE_DATABASE* parameter is uncommented and set to a valid path.

```
*****
# Specify a pay-per-use usage database, which can be used for billing
# purposes.
#
```

```
# The format of this database and an example of data printout is
# described in the LM-X end user documentation.
#
#*****
USAGE_DATABASE = d:\lmx\usage.db
```

Remember the path to the usage database as you may have to send this file to REAL FUSIO on a regular basis depending on your contract.

2.2.2 - Install the license file

Copy the license file (*.Lic) provided by REAL FUSIO to the LM-X server folder.

2.2.3 - Extract the ∞Directory keys

1. Open the license file in a text editor.
2. Find the lines starting with `FEATURE DIRECTORY. FEATURE DIRECTORY-PROD-5347F99D-5479-4071-A057-9351DCE3A9AF` (`VENDOR=REALFUSIO COUNT=1 VERSION=...` ...

Each such line corresponds to an instance of ∞Directory. In the given example, the key is `DIRECTORY-PROD-5347F99D-5479-4071-A057-9351DCE3A9AF` and it corresponds to a *production* ∞Directory (hence the `-PROD-`). Depending on your contract, you may find several ∞Directory keys in the license file:

- `DIRECTORY-PROD-...` is a key for a *production* ∞Directory.
- `DIRECTORY-EVAL-...` is a key for an *evaluation* ∞Directory.
- `DIRECTORY-TEST-...` is a key for a *test* ∞Directory.

Remember the ∞Directory key matching the type of the ∞Directory being installed (`${lmx_feature}`).

3 - OpenId Connect

User authentication is deferred by the ∞Directory to a third party server using [OpenId Connect code flow protocol](#). The following json describes how the ∞Directory will contact this server to authenticate the user and retrieve an `id_token`. `id_token` should be delivered as a valid [JWT \(rfc7519\)](#).

```
{
  "configuration_end_point": "https://authentication.server.host/.well-known/openid-configuration",
  "client_id": "...",
  "client_secret": "...",
  "hmac_secret": "..",
  "allowed_jwt_alg": ["HS256", "HS384", "HS512", "RS256", "RS384", "RS512", "ES256", "ES384", "ES512"],
  "use_PKCE": true,
  "verify_ssl_peer": true,
  "id_token_alias": {
    "custom_field_A": "email",
    ...
  }
}
```

- The `configuration_end_point` field is the [OpenID Provider configuration url](#)
- The `client_id` field is the application id generated by the authentication server
- The `client_secret` field is the secret associated by the authentication server to the application
- The `hmac_secret` field will be used to validate the jwt signed with HMAC algorithm. It MUST be set if `allowed_jwt_alg` contains any `HS*` entry else it MUST be empty
- The `allowed_jwt_alg` field is the list of algorithm that will be allowed for the JWT delivered by the authentication server
- The `use_PKCE` field allow to enable [Proof Key for Code Exchange \(rfc7636\)](#)
- The `verify_ssl_peer` field enables the verification of the authentication server SSL certificat by the ∞Directory
- The `id_token_alias` field allows to map custom id_token fields to generic ones. This field is optional

Following optional fields allowed to customize some end points if you need to send custom query parameters to the OpenID Provider or if result parsing is not consistent with the specification. The following json describes additional fields.

```
{
  ...,
  "authorize_end_point": {
    "res_alias": {
      "state": null,
      "error": null,
      "error_description": null,
      "code": null
    },
    "query_parameters": {
    }
  },
  "token_end_point": {
    "query_parameters": {
    }
  },
  "jwks_end_point": {
    "query_parameters": {
    }
  }
}
```

- The `authorize_end_point` block describes the “authorize end point” of the authentication server
 - The `query_parameters` allows to pass extra parameters to the authentication server. For example it could be used to force the value of the `prompt` parameter
 - The `res_alias` specifies the aliases of result fields if they are different from the OpenId Connect specification
- The `token_end_point` block describes the “token end point” of the authentication server. This end point will be used to retrieve the `id_token` of the user
 - The `query_parameters` allows to pass extra parameters to the authentication server

- The `jwt_end_point` block describes the “jwt end point” of the authentication server. This end point is used to retrieve public keys to verify JWT signed with `RS*` or `ES*` algorithm. Responses of this end point should follow [JWKS specification \(rfc7517\)](#)
 - The `query_parameters` allows to pass extra parameters to the authentication server

If your OpenID Provider does not have a configuration url, you can specify end points manually. This method is **Deprecated** and will be removed in future release. It also has limitations on issuer value. To use this method, `configuration_end_point` field should be omitted. The following json describes additional fields.

```
{
  ...,
  "host": "authentication.server.host",
  "port": 443,
  "authorize_end_point": {
    "path": "authorize",
    ...
  },
  "token_end_point": {
    "path": "oauth/token",
    ...
  },
  "jwt_end_point": {
    "path": ".well-known/jwks.json" ...
  }
}
```

- The `host` field is the host name of the authentication server (without https scheme or port)
- The `port` field is the HTTPS port of the authentication server
- The `authorize_end_point` block describes the “authorize end point” of the authentication server
 - The `query_parameters` allows to pass extra parameters to the authentication server. For example it could be used to force the value of the `prompt` parameter
 - The `res_alias` specifies the aliases of result fields if they are different from the OpenId Connect specification
- The `token_end_point` block describes the “token end point” of the authentication server. This end point will be used to retrieve the `id_token` of the user
 - The `path` field is the uri path to the token end point
- The `jwt_end_point` block describes the “jwt end point” of the authentication server. This end point is used to retrieve public keys to verify JWT signed with `RS*` or `ES*` algorithm. Responses of this end point should follow [JWKS specification \(rfc7517\)](#)
 - The `path` field is the uri path to the jwt end point

Administration

This chapter provides hints for the following administrative tasks:

- manage 3D Juump Infinite through its web application interface - deploy 3D Juump Infinite within a network with a network address translation (NAT) mechanism
- enumerates all the services started and opened and TCP/IP ports to assist you to properly install 3D Juump Infinite in conformance with your enterprise security policy
- activity script for billing purpose

1 - Web application interface

First, make sure that the ∞Directory has been installed properly and is running along with its dependent back-end services. The ∞Directory provides a web application interface that lets the administrator manage the 3D Juump Infinite network, data and user rights. This web application is accessible through the ∞Directory web server, by browsing the corresponding URL:

`https://directory_name:\${apache_https_port}/directory`



Please display the web administration interface with a supported browser: Microsoft Internet Explorer 11 and above, Google Chrome or Mozilla Firefox.

1.1 - Login

The *Login* page prompts the administrator to prove his identity through the defined delegation. If Single Sign-On (SSO) is supported by your *authenticator*, the login page is skipped.

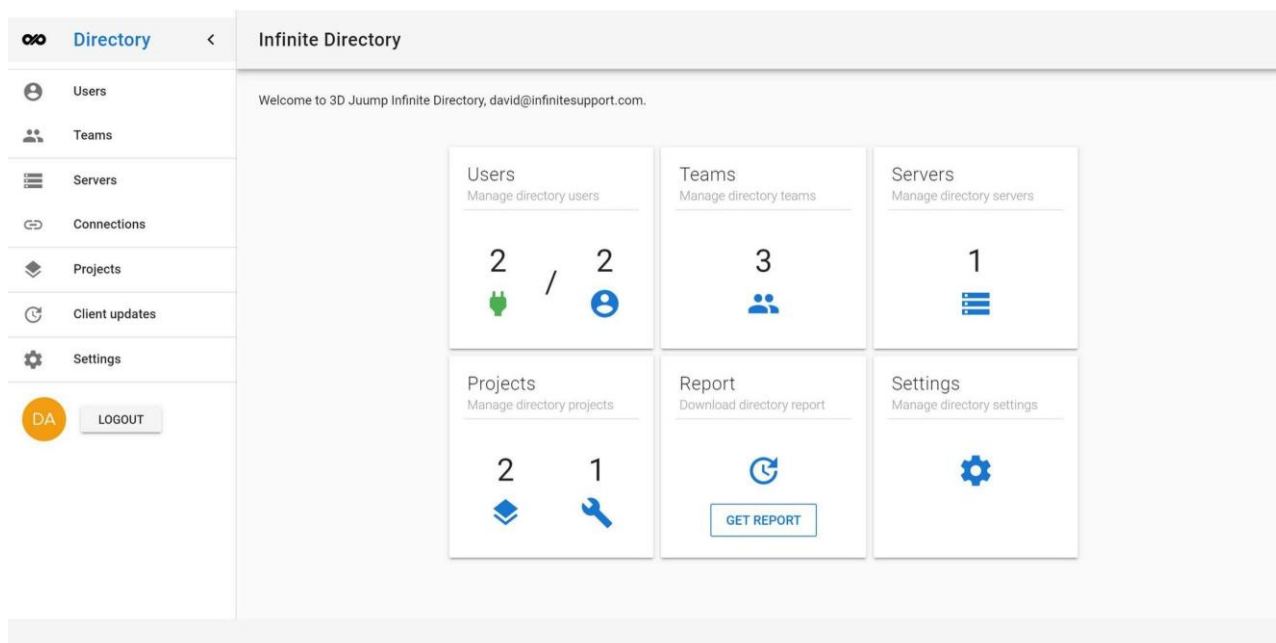


Login

Only users with administrator privilege can successfully login to the front-end. One exception is the superuser login: it is always possible to fallback to a special login procedure relying on the ∞Directory PostgreSQL superuser credentials. In particular, upon prime installation, the ∞Directory has no registered user, thus no registered administrator: use the fallback superuser login procedure to initiate the ∞Directory user database with its first administrator.

1.2 - Dashboard

Once logged in, the browser proceeds to a *Dashboard* summarizing the current state of 3D Juump Infinite.



Dashboard

It lists several statistics about 3D Juump Infinite users, teams, projects and servers.

A menu bar, located at the left of the *Dashboard*, gives access to dedicated administration panels. Note the top-left infinite icon: it is a shortcut for a quick access to the *Dashboard*.

1.3 - Team Management

1.3.1 - Users

The *User Management* page lets the administrator create, edit and delete users. It also lists the registered users.

The screenshot displays the 'User Management' page. On the left is a sidebar with navigation links: Directory, Users (selected), Teams, Servers, Connections, Projects, Client updates, and Settings. Below these is a 'LOGOUT' button. The main content area is titled 'User Management' and shows 'TOTAL USERS: 2'. It features a 'NEW USER' button, search fields for 'Search text' and 'Search tags, teams', and a table of users.

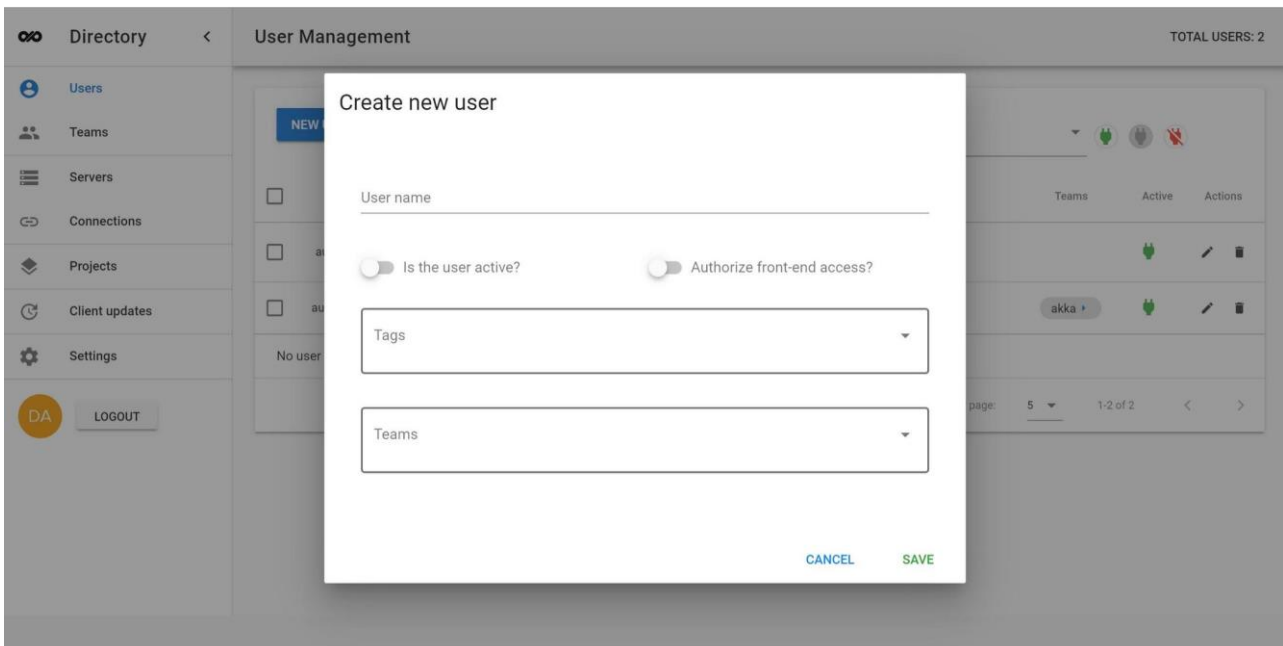
<input type="checkbox"/>	User ID	Infos	Personal tags	Teams	Active	Actions
<input type="checkbox"/>	auth0 5ccc073f90af3710f269d988	SY sylvain@infinitesupport.com sylvain@infinitesupport.com	view:Link&Go export2d:Link&Go			
<input type="checkbox"/>	auth0 5cdc1611db4d301107026c77	DA david@infinitesupport.com david@infinitesupport.com		akka		
No user selected						

At the bottom right of the table, it shows 'Rows per page: 5' and '1-2 of 2'.

Users

1.3.1.1 - Create Users

When the administrator needs to create a new user, he clicks on the *New User* button (top-left) and fills in the user id for this user (the one delivered by the *authenticator*). It is also possible to activate the user, to grant him administrator privilege (the ability to access the *front-end*), to decorate him with tags or to inscribe him in teams.

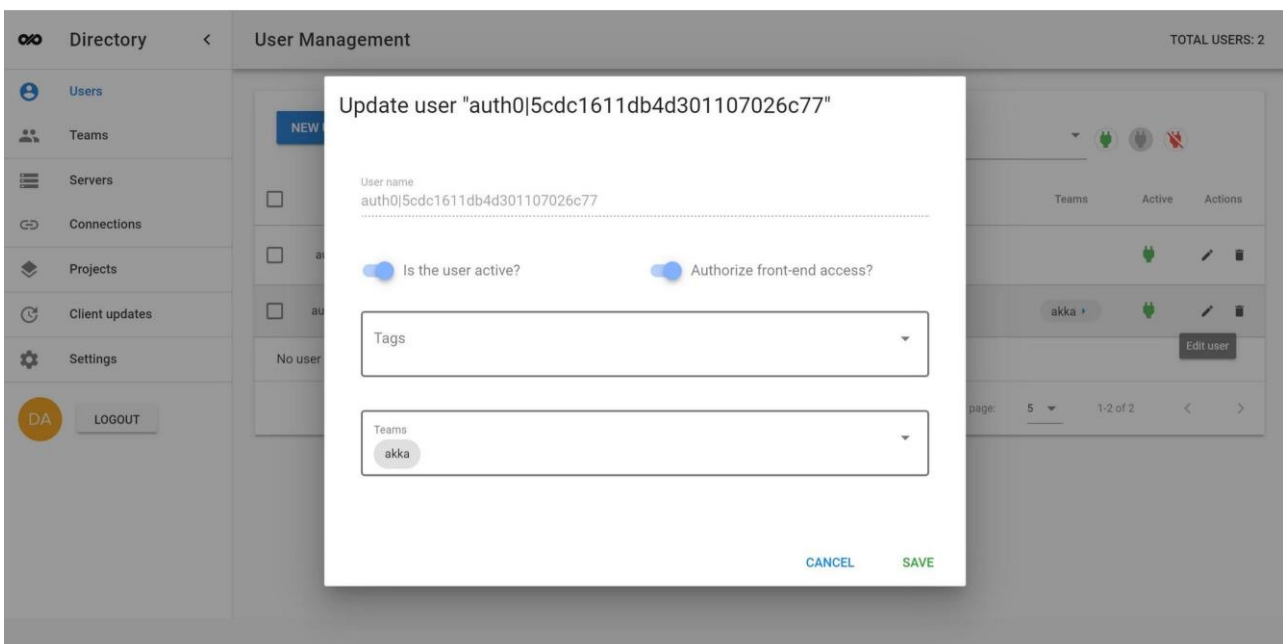


Create User

Another way to create users is to have them login from the 3D Juump Infinite client application. Once properly authenticated, the user id is automatically registered in the ∞Directory users database, though the corresponding user remains inactive, has no privilege, nor tags, nor teams.

1.3.1.2 - Edit Users

The administrator can edit a user by clicking on the pencil icon that appears on the corresponding line.



Edit User

Of course, it is impossible to change the user id of a given user as it links him to his *authenticator*-provided identity.

1.3.1.3 - Delete Users

To delete an existing user, the administrator can click on the trash-can icon that appears on the corresponding line.

1.3.1.4 - Batch User Creation

Batch user creation is not part of the *front-end* anymore and is only available through the *Directory API*.

1.3.1.5 - Send messages

It is not possible to send a message to selected users anymore: to adress messages to specific users, we suggest you rather use emails.

1.3.2 - Teams

The *Team Management* page lets the administrator create, edit and delete teams. It also lists the created teams.

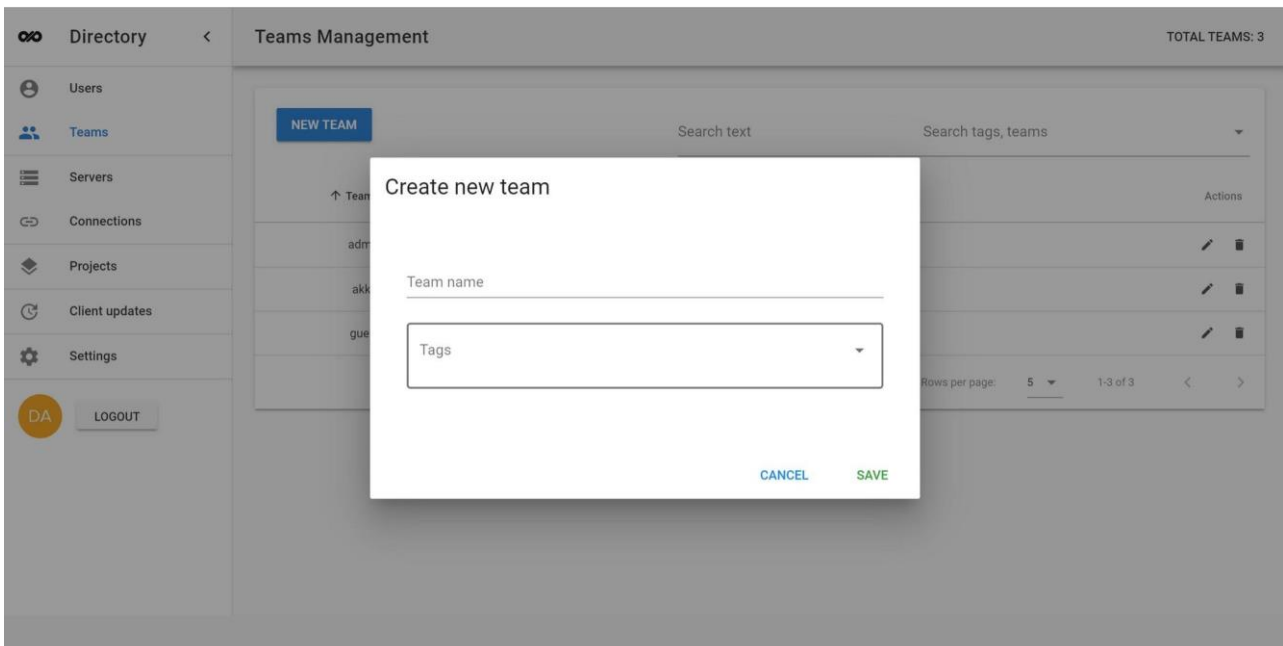
Teamname	Tags	Actions
admin	view:Link&Go export2d:Link&Go (+4 more)	[edit] [delete]
akka	view:Link&Go connect:server (+1 more)	[edit] [delete]
guest	connect:server connect:support (+1 more)	[edit] [delete]

Teams

Teams are useful to quickly set tags to a group of users.

1.3.2.1 - Create Teams

When the administrator needs to create a new team, he clicks on the *New Team* button (top-left), fills in the (unique) name for this team and sets its tags.

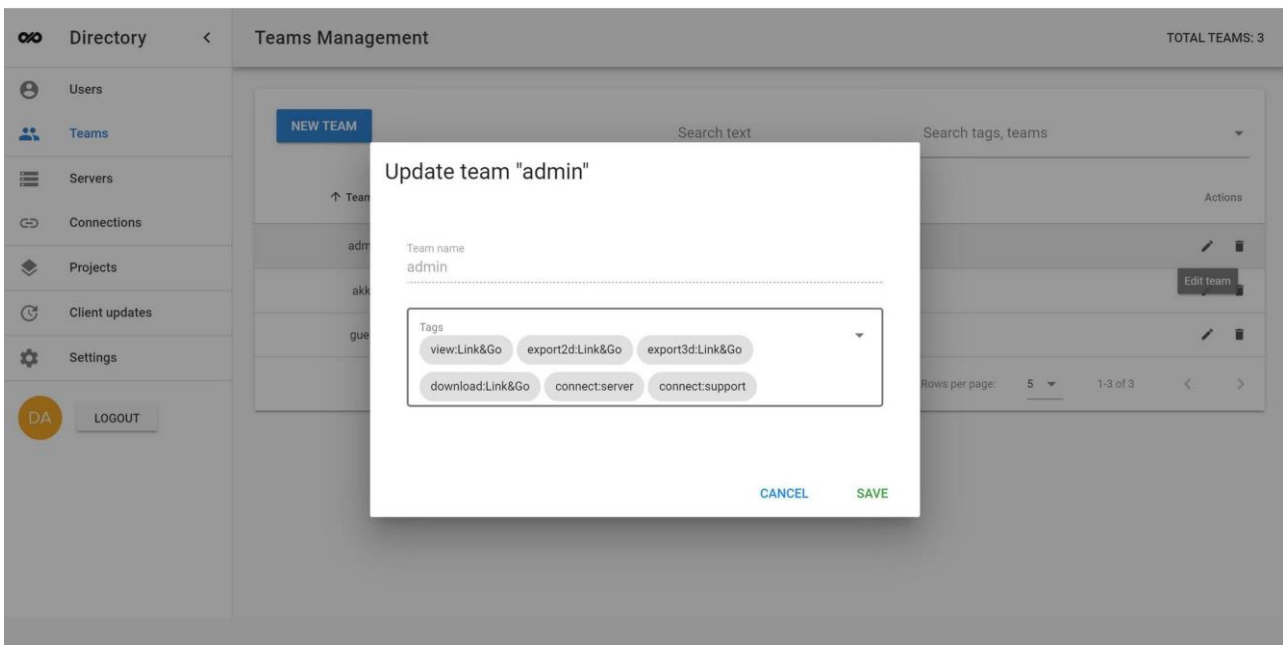


Create Team

Note: it is not possible to change a team's name once it is created.

1.3.2.2 - Edit Teams

The administrator can edit a team's tags directly by clicking on the pencil icon that appears on the corresponding line. It is possible to change the tags associated to the team.



Edit Team

1.3.2.3 - Deleting Teams

To delete an existing team, the administrator can click on the trash-can icon that appears on the corresponding line.

1.3.2.4 - Send messages

It is not possible to send a message to teams anymore.

1.4 - Network



1.4.1 - Servers

The *Servers Management* page lets the administrator register new servers (∞Server, ∞Proxy or ∞StaticProxy) to the 3D Juump Infinite network of this ∞Directory. It also lists the registered servers. As you might guess, it requires some installed ∞Servers; please go to ∞Server installation section if no ∞Server has been installed yet.

Servers

1.4.1.1 - Server Info

The list displays one line per server with several pieces of information (from left to right):

- The left-most column contains an icon representing the type of server (either a ∞Server  or a ∞Proxy ).
- A status icon indicates whether the server is properly running or not. On mouse-over, it also reports information regarding the server's sub-services status and versions.
- The *Label* column contains the name of the server as defined by the administrator upon [registration](#).
- The *URL* column contains the host address of the given ∞Server/∞Proxy.
- The *PG Port* columns contain the PostgreSQL port of the server.
- The *HTTPS Port* columns contain the HTTPS port of the server.
- The *Tags* column displays the tags of this server, that is to say the tags that a user must match to connect to this server.

- The *Active connections* column displays the number of sessions currently opened to this server.
- The *Accept new connections* column displays if the given ∞Server/∞Proxy may accept user connections.
- The *Actions* column gives access to several actions listed below.

1.4.1.2 - Register Servers

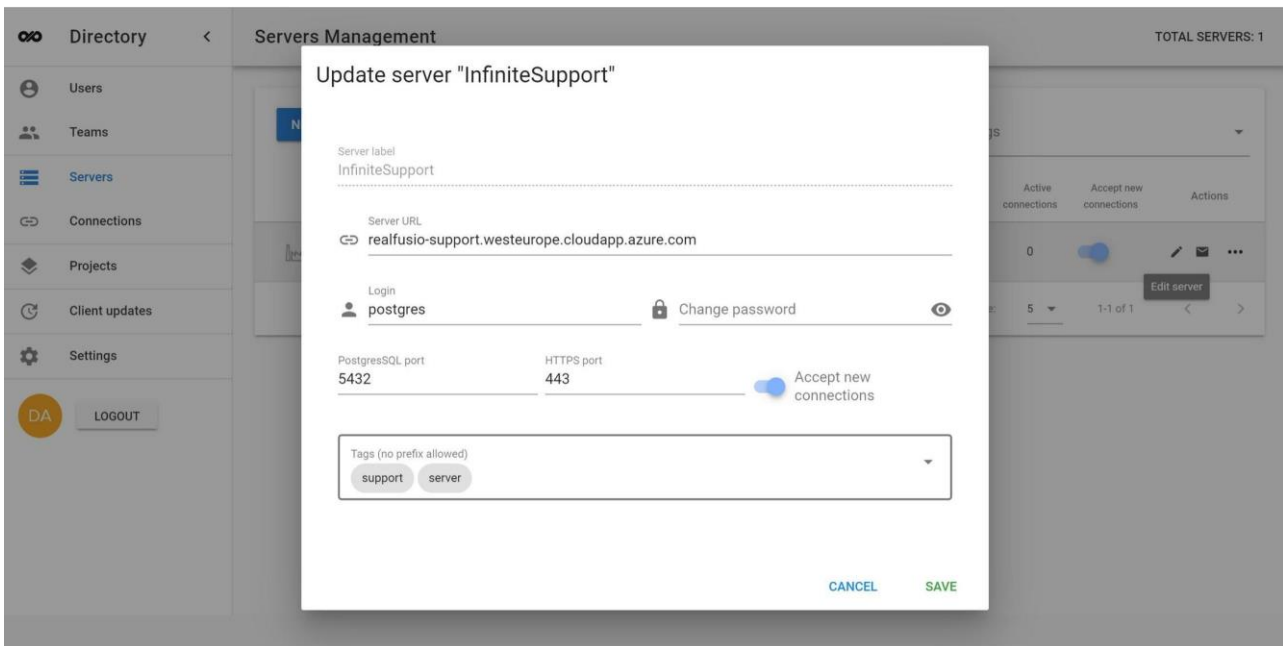
The administrator can register a new server by clicking on the *New Server* button (top-left) and filling in the form with:

Registering Servers

- The name of the server (for human-readable label displayed in the ∞Directory).
- The public address of the server (IPv4, IPv6, DNS or NetBIOS host name).
- The login to use to access the server PostgreSQL database as administrator (*`\${postgres_login}`*).
- The password to use to access the server PostgreSQL database as administrator (*`\${postgres_password}`*).
- The port of the PostgreSQL server (*`\${postgres_port}`*).
- The port of the HTTPS server (*`\${apache_https_port}`*).
- If the ∞Server/∞Proxy can accept user connections.
- The tags decorating the server and effectively limiting the user's connections.

1.4.1.3 - Update Servers

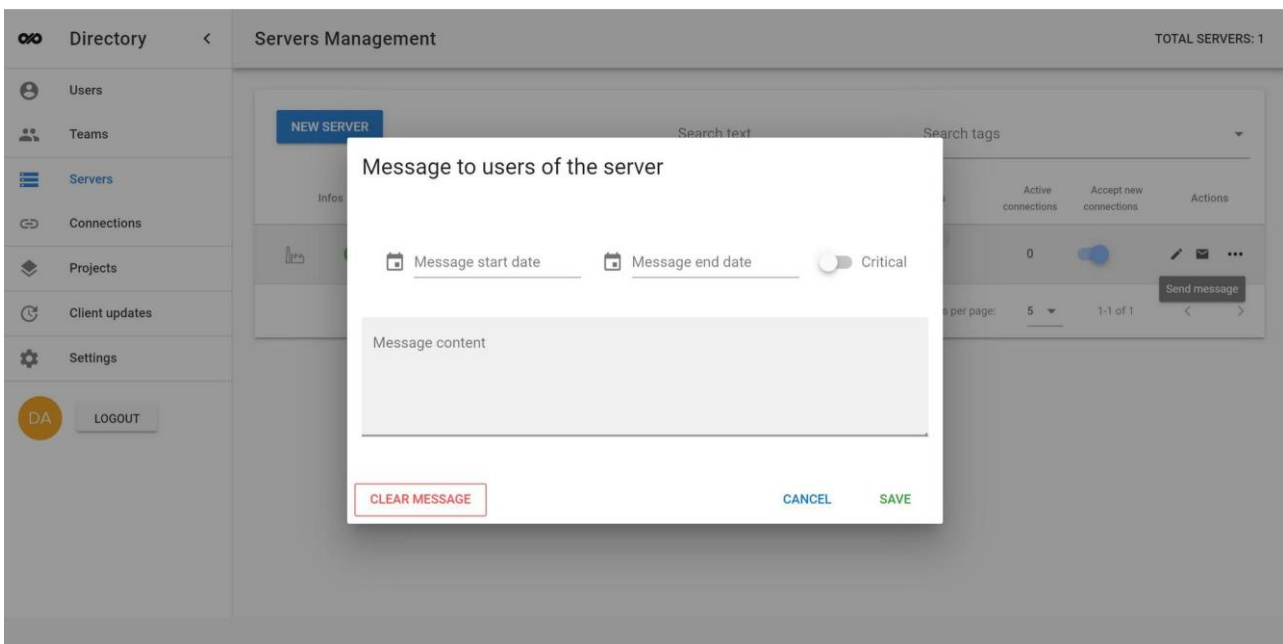
To update an existing server, the administrator can click on the pencil icon in the *Actions* column.



Update Server

1.4.1.4 - Send message

To send a message to users on a server, the administrator can click on the mail icon in the *Actions* column.

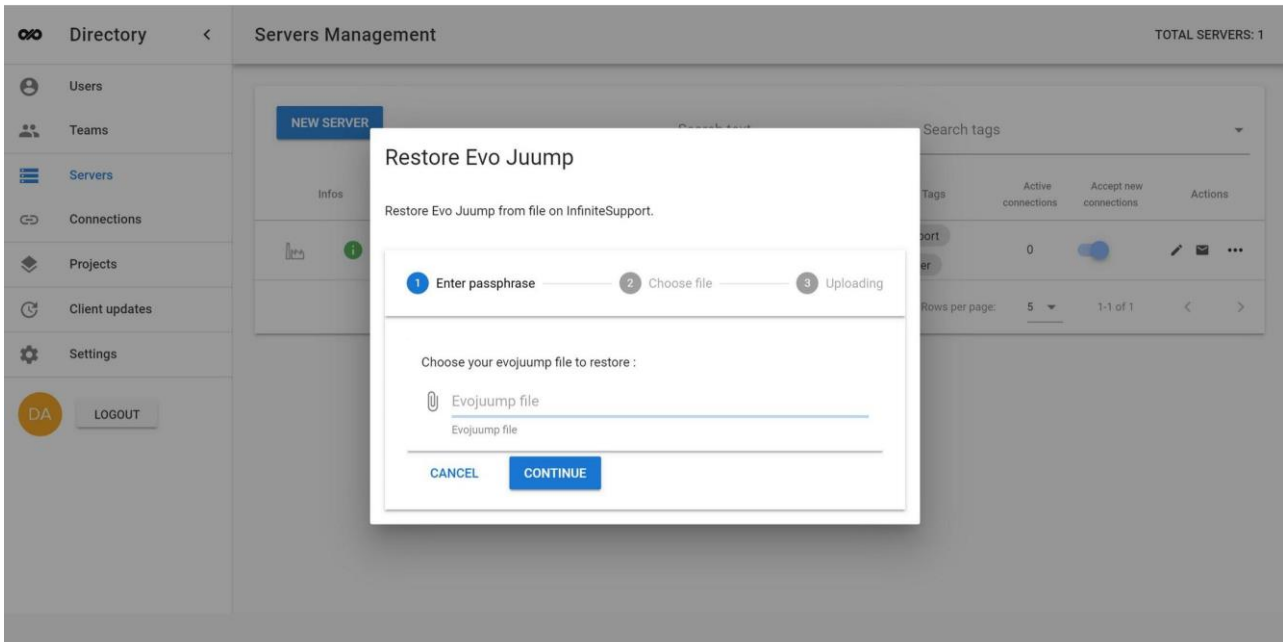


Send Message

The corresponding form lets the administrator define the message displayed on the server's billboard (when it is visible, what is its content, whether it is critical or standard...). The server's billboard is automatically shown to users connected or connecting to the server. The administrator can also clear the server's billboard.

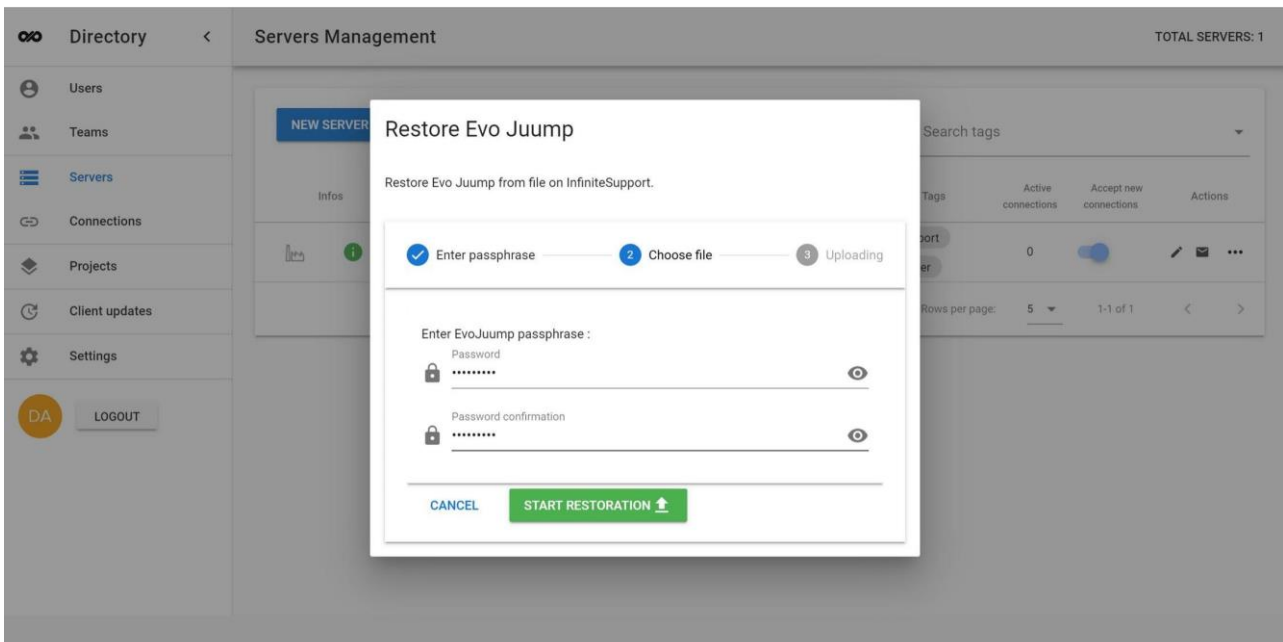
1.4.1.5 - Restore builds

The administrator can also restore a build previously packed. It is feasible on any ∞Server or ∞StaticProxy by selecting the corresponding action in the *Actions* column.



Choose file

First, select the evojump file from your drive.



Enter passphrase

Then provide the passphrase (if any).

Fill in the required information and proceed to the build installation.

1.4.1.6 - Clean-up Proxy

It is possible to clean-up unreferenced documents from a server/proxy by clicking on the corresponding action in the *Actions* column.

1.4.1.7 - Revoke all users

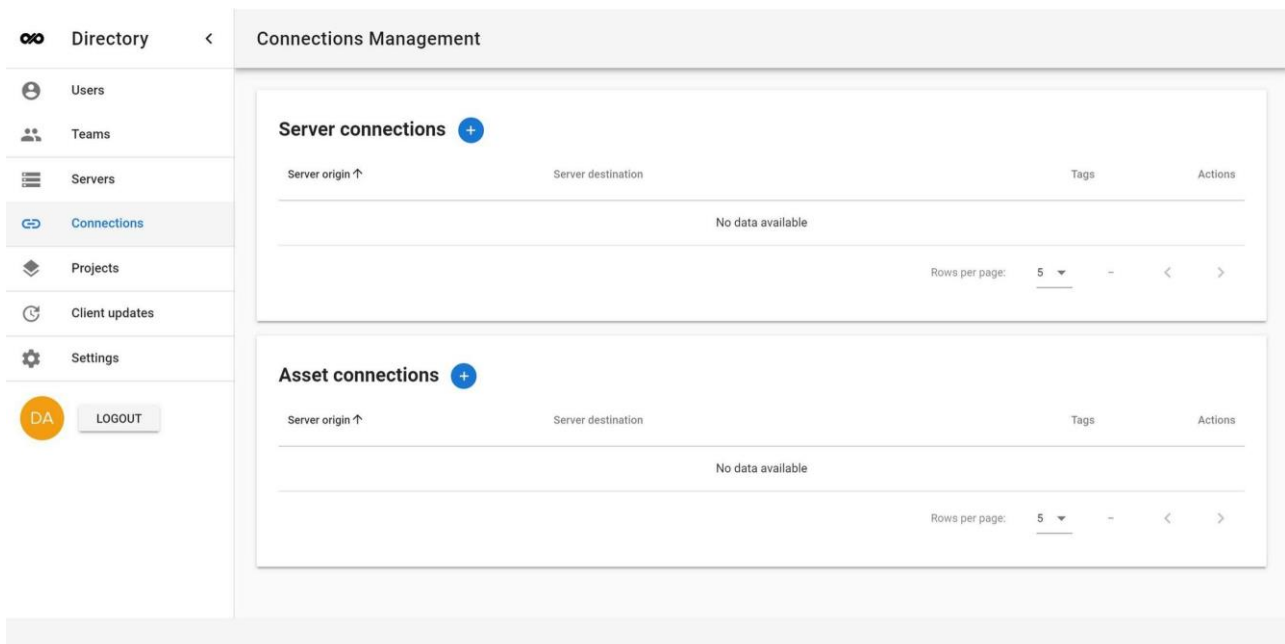
The administrator can forcefully revoke all connected users from a server by clicking on the corresponding action in the *Actions* column.

1.4.1.8 - Delete Servers

The administrator can delete an existing server by clicking on the corresponding action in the *Actions* column.

1.4.2 - Connections

The *Connections Management* page lets the administrator define the [DMU-flow](#), the manner in which 3D Juump Infinite connects ∞Proxies and ∞Servers to broadcast the DMU to the 3D Juump Infinite client applications. It also provides a list of the existing connections. Moreover, it lets the administrator set replication routes (for assets).



Connections

1.4.2.1 - Server connections

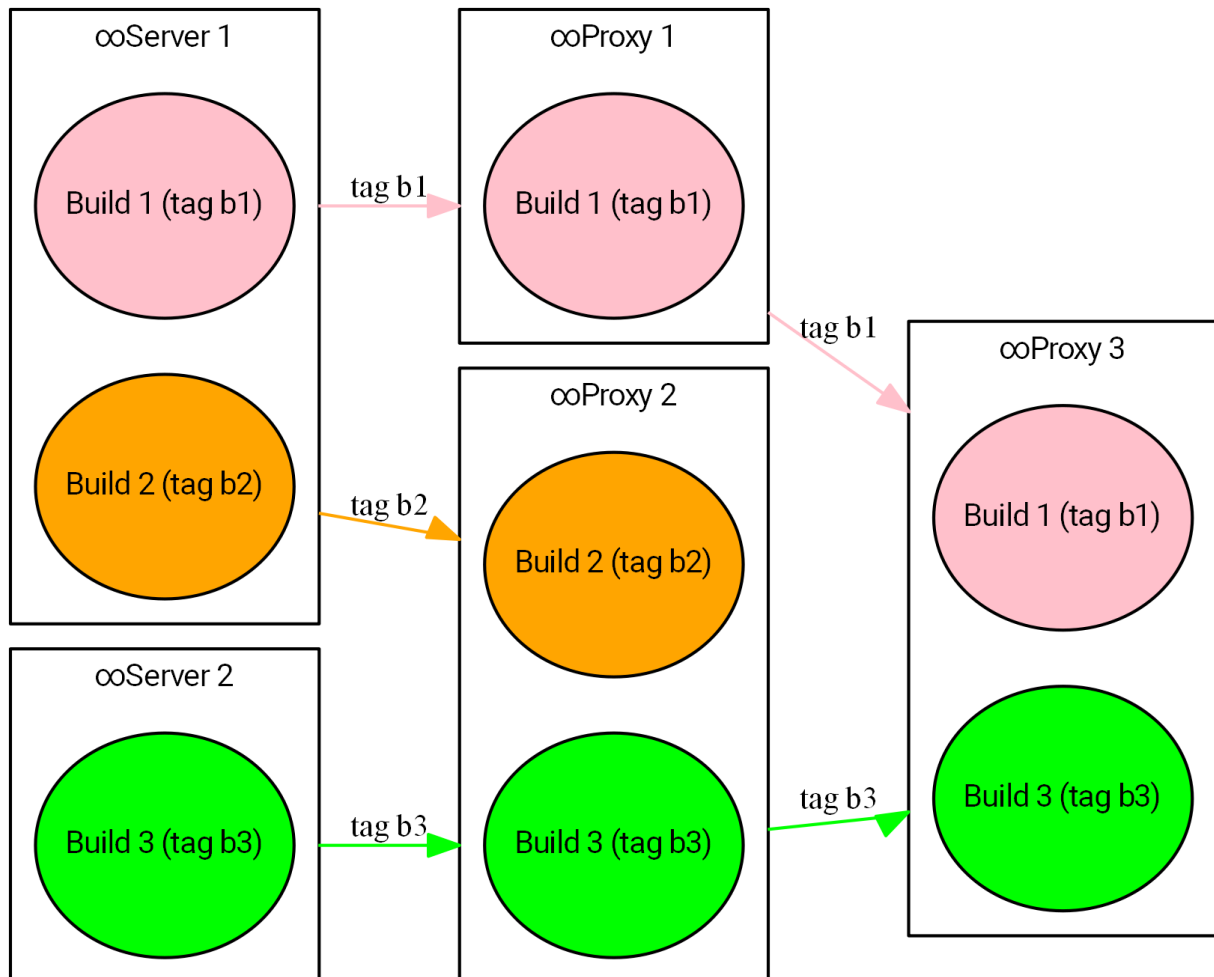
Each ∞Proxy replicates all the *builds* and *assets* from its *sources* (the ∞Servers or ∞Proxies that it takes its data from) given that the *connection* matches the tags of the *builds*.

In the connections list, from left to right:

- The *Server origin* column indicates the source ∞Server or ∞Proxy.
- The *Server destination* column indicates the destination ∞Proxy.

- The *Tags* column indicates the limitations to builds replication. If no tags are set, all builds are replicated. Else, only the builds matching the tags are replicated.

All the assets of the projects the matching builds belong to will also be synchronized.



Sources

In the example, two *ooServers* generate three projects and are replicated by three *ooProxies*:

- *ooServer 1* generates two builds tagged resp. *b1* and *b2*.
- *ooServer 2* generates one build tagged *b3*.
- *ooProxy 1* has *ooServer 1* as its only source and its connection is tagged *b1* only.
- *ooProxy 2* has two connections (with *ooServer 1* and *ooServer 2*) tagged resp. *b2* and *b3*.
- *ooProxy 3* has two connections (with *ooProxy 1* and *ooProxy 2*) tagged resp. *b1* and *b3*.

All the assets of the projects the replicated builds belong to will also be synchronized.

1.4.2.2 - Asset connections

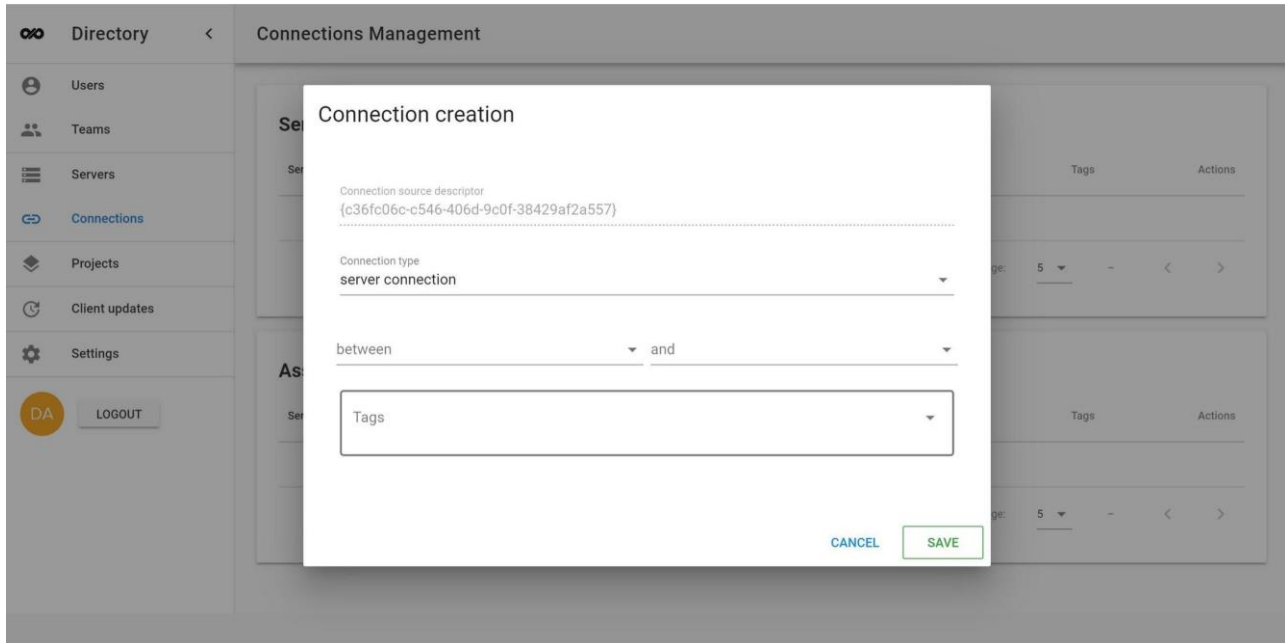
Asset connections are similar to Server connections but only replicate *assets* and no *builds*. The tags rules are slightly different. The *Tags* column indicates the limitations to asset synchronization. If no tags are set, all assets of all the projects from the source are replicated.

Else, the projects from the source whose name is included in the tags connection ("_prj_XXX") will be synchronized. Projects with no build will not have their assets synchronized.

1.4.2.3 - Add connection

The administrator can add a new connection by clicking on the + button and fill in the form with:

- The type of connection (either server or asset).
- The source ∞Server or ∞Proxy.
- The destination ∞Proxy.
- The tags to limit replication.



The screenshot shows the 'Connections Management' page in a web application. A modal titled 'Connection creation' is open, displaying the following fields:

- Connection source descriptor:** A text field containing the UUID {c36fc06c-c546-406d-9c0f-38429af2a557}.
- Connection type:** A dropdown menu with 'server connection' selected.
- between** and **and**: Two dropdown menus for selecting source and destination.
- Tags:** A text input field.

At the bottom right of the modal are 'CANCEL' and 'SAVE' buttons. The background shows a sidebar with navigation links: Directory, Users, Teams, Servers, Connections, Projects, Client updates, and Settings, along with a 'LOGOUT' button.

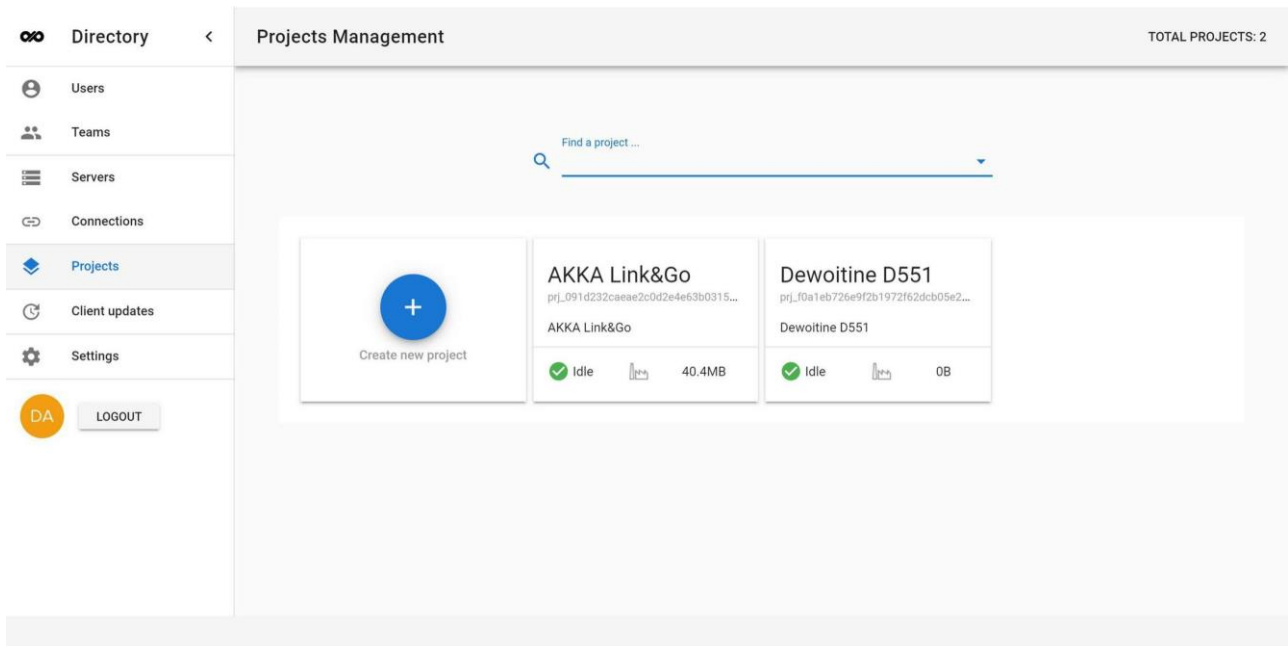
Adding Sources

1.4.2.4 - Delete connection

The administrator can remove a connection by clicking on the trash icon.

1.5 - Projects

The *Projects* page lets the administrator create new projects, check the status of existing projects and list existing builds.



Projects

1.5.1 - Create Projects

When the administrator needs to create a new project, he clicks on the **+** button. Then, a form pops up and requires entering the following information:

- The human readable name of the project.
- The ∞ Server which will host the project.
- The project's unit.

Note: the creation of a project may take several minutes.

The newly created project will appear in the project selection combo-box. After few minutes, please refresh the web page to display project creation status.

The screenshot shows the 'Create project' dialog box within the 'Projects Management' section of the application. The dialog has a title bar 'Create project'. It contains several input fields: 'Project name' (a text input), 'Generator' (a dropdown menu), 'Unit' (a dropdown menu), and 'Project ID' (a text input showing the value 'prj_a48f28bca0d4a475cea17dc31783deb6'). Below these fields, there is a section titled 'Or, restore a project.' followed by a text input field with the placeholder 'Please, upload a project description.' and a small icon of a document. At the bottom right of the dialog, there are two buttons: 'CANCEL' and 'SEND EVENT'. The background of the application shows a sidebar with navigation links: Directory, Users, Teams, Servers, Connections, Projects (highlighted), Client updates, and Settings. There is also a 'LOGOUT' button and a 'DA' icon. The top right corner of the application shows 'TOTAL PROJECTS: 2'.

Project Creation



Please refresh the web page to update the creation status display.



This project unique identifier might be an useful piece of information for the [Connector development](#).

Note: it is now possible to create a project from its *project descriptor*. It is required for the backup-restore procedure.

1.5.2 - Check a Project Status

The *Project Management* page gives an overview of all projects status. For more detailed information, click on the project's tile.

The screenshot shows the 'Project AKKA Link&Go' page. The left sidebar contains navigation links: Directory, Users, Teams, Servers, Connections, Projects (highlighted), Client updates, and Settings. The main content area shows the project name and ID, a 'TOTAL BUILDS: 1' counter, and a 'Status: Idle' indicator with a green checkmark. A 'BUILD' button is visible. Below this is a 'Build comment' section with a timestamp of '03 MAY 2019 16:34 UTC'. A table titled 'PROXIES' is displayed, showing one proxy named 'InfiniteSupport' with a status of 'HTTP' and a green checkmark.

Role	Label	Status	Action
	InfiniteSupport	HTTP	

Project page

A status bar displays the current [status](#) of the project, either:

- **Idle**: the project is in idle state, waiting for new data.
- **Updating**: the project is currently being fed with new data.
- **Building**: the project is currently compiling a new *build*.
- **Packing**: the project is currently packing a *build* for export.

The status can also denote an error in the update process.

The actions available in the project menu are described below.

1.5.3 - Request a Build

In the project menu, the **Build** action commands the ∞Server to generate a new *build* as soon as the project is *buildable* (i.e. it is in *idle* state and the connector has declared that the database is coherent).



Please refresh the web page to update the build status.

1.5.4 - Synchronize metadata

In the project menu, the **Synchronize metadata** action commands the ∞Server to update the metadata of existing *build* with new data from the *Connector*.

1.5.5 - Get descriptor

In the project menu, the **Get descriptor** action retrieves the *project descriptor* of this project for later restoration.

1.5.6 - Clean generation data

In the project menu, the *Clean generation data* action commands the ∞Server to clean-up generation data.

1.5.7 - List Builds

Under the status bar are listed all the existing *builds* for the selected project. For each *build*, the page contains:

- The name of the *build* (as defined in the *buildcomment* of the [project status document](#)).
- Its unique identifier (for internal use).
- Its creation date.

Several panels are available:

- The *Proxies* panel lists all the servers (∞Server or ∞Proxy) publishing this *build*.
- The *Info* panel lists the tags of the build and its complete description.
- The *Evojump* panel lets the administrator create/download a file image of a packed build.

1.5.8 - Delete a Build

In the *Proxies* panel, it is possible to delete the build from the corresponding ∞Server or ∞Proxies.

1.5.9 - Export a Build

To export a *build* for standalone use, the *build* must first be packed in a secured container - called *evojump*. This is done by clicking on the corresponding button in the *Evojump* panel. A dialog then asks for a passphrase to protect the *evojump*.

Once the *evojump* is packed, two button appears:

- a *delete* button (to remove the corresponding *evojump* from the ∞Server),
- a *download* button (to retrieve the corresponding *evojump* file).

1.5.10 - Delete Projects

In the project menu, the *Delete project* action lets the administrator totally delete a projects.

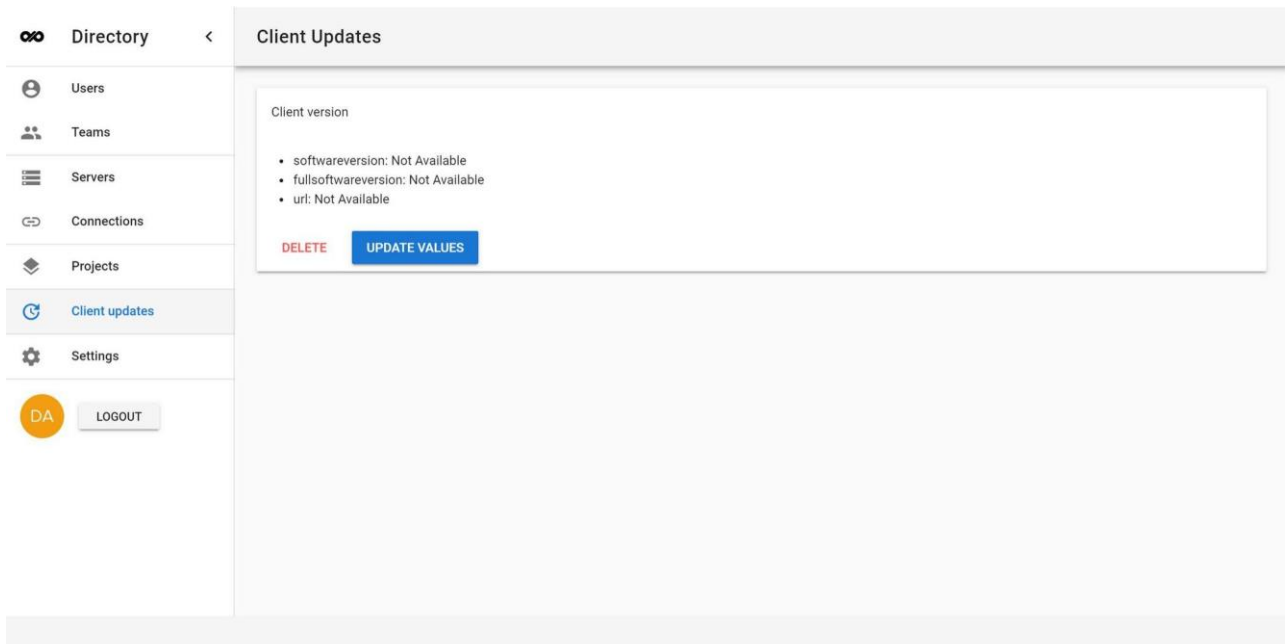
1.6 - Settings

Setting	Range	Description
<i>auth_allowed_callback_urls</i>	-	comma separated list of allowed redirect uris. Each uri corresponds to an application that will use the directoryapi. Typical value is https://mydirectory:443/directory,https://mydirectory:443/directoryapi/directorysession/onauthenticationdone
<i>client_hardwarereport</i>	-	enables ∞Client hardware report
<i>client_usagereport</i>	-	enables ∞Client usage report
<i>data_session_max_heartbeat_interval_sec</i>	600 / 7200	maximum duration between two heartbeats from ∞Client application before session revocation

<code>directory_session_lock_on_ip</code>	-	if true ∞Directory and data sessions will be locked on client ip. It is preconized to register servers by host name when using this security feature
<code>directory_session_max_client_idle_duration_sec</code>	3660 / 34560 0	if there is no user activity during this duration, the ∞Client will close the ∞Directory session
<code>directory_session_max_duration_sec</code>	3660 / 34560 0	maximum duration of the ∞Directory session, if elapsed the ∞Directory session will be closed
<code>directory_session_max_heartbeat_interval_sec</code>	60 / 14400	maximum interval between two ∞Directory session heartbeats, if elapsed the ∞Directory session will be closed
<code>directory_session_wait_for_auth_sec</code>	60 / 900	maximum duration of the authentication procedure, if elapsed the ∞Directory session will be closed
<code>max_dmu_rent_time_sec</code>	86400 / 80352 00	maximum dmu rent duration
<code>max_token_offline_time_sec</code>	86400 / 80352 00	maximum duration of lmx borrowed license
<code>proxy_access_token_maxage_sec</code>	30 / 600	maximum duration of ∞Proxy access tokens delivered by the ∞Directory
<code>sso_user_consent_interval_sec</code>	0 / 60480 0	maximum duration between two user consents. If 0 SSO will be disabled
<code>verbose</code>	-	enables verbose logging
<code>webhook_report_interval_sec</code>	60 / 60480 0	duration between two calls to <code>webhook_url</code>
<code>webhook_url</code>	-	a webhook url where ∞Directory and ∞Proxies will POST theirs errors

1.7 - Client Updates

The *Client Updates* page lets the administrator register new client versions to the ∞Directory. Clients connecting to this directory will be asked to download the new version.



Client Updates

1.7.1 - Register new Client Updates

Each new client version comes with at least :

- A **json** file (*3djuumpinfinite_win_\${version}.json*).
- The new installers for this version (*3D Juump Infinite X64-setup-standarduser-\${version}.exe*, *3D Juump Infinite X64-setup-admin-\${version}.exe*).

The json file follows the following structure :

```
{
  "applicationname" : "3D JuumpInfiniteClient",
  "systemname" : "WINDOWS",
  "softwareversion" : "${internal_version}",
  "fullsoftwareversion" : "${version}",
  "url" : "http://"
}
```

In order to provide a new update for all clients :

- Copy the requested installer (admin or standard) to a http server, note the url to download the given installer
- Edit the *3djuumpinfinite_win_\${version}.json* and replace the "url" field with the url of the installer
- Register the json file to the ∞Directory by clicking on the *Update Values* button.

2 - Error notification (Webhook)

A webhook url could be specified on ∞Directory [Settings](#) to be notified of log entries. Here is a description of the POST content :

```
{
  "numdebug": 0,
  "lastdebug" : [...],
  "numinformation": 0,
  "lastinformation" : [...],
  "numwarnings": 0,
  "lastwarnings" : [...],
  "numcriticals": 0,
  "lastcriticals" : [...]}

```

- The *num** is the number of log entries of this type since last webhook call
- The *last** is a list containing last three messages of this type

3 - Migration procedures

3.1 - Cluster migration

To migrate a cluster:

1. Export the descriptors of all projects (from the ∞Directory)
2. Pack and download any useful build from all projects (from the ∞Directory)
3. Delete all projects from the cluster
4. Stop all servers (∞Directory, ∞Servers and ∞Proxies)
5. Install the new ∞Directory software, then restart the ∞Directory server
6. Install the new ∞Server software, then restart all ∞Server servers
7. Install the new ∞Proxy software, then restart all ∞Proxy servers
8. Recreate all projects from their descriptors
9. Migrate the builds (see the corresponding chapter in the Integration Manual)
10. Upload and unpack the migrated builds

3.2 - Assets migration



Migrating from 3.0 and below requires that all assets are patched to reflect the new authentication mechanisms. The assets' security descriptors have to be updated to reflect the new team and user identifiers.

1. Before starting the procedure, make sure all asset databases are properly backedup
2. Retrieve the internal team identifiers from the old cluster
3. Migrate the cluster (see above)
4. Recreate the teams
5. Prepare a map of all users, matching the previous user identifier to the corresponding subid from the new *Authenticator*
6. Prepare a map of all teams, matching the previous team identifier to the corresponding team identifier from the new ∞Directory
7. Patch all assets, replacing previous user and team identifiers in the security descriptor.

Annexes

These annexes contain:

- the release notes,
- the security guidelines,
- the list of third-party licenses,
- the list of known limitations.

1 - Release notes

1.1 - 16/01/2024: version 3.1.16

Native Client

- Fixed LTA banner display

1.2 - 08/12/2023: version 3.1.15

Hub

- Added LTA license support

1.3 - 15/03/2021: version 3.1.14**Client**

- Fixed hang during bom / 3d export

1.4 - 21/02/2021: version 3.1.13**Client**

- Fixed bad visibility / color during screenshot export

1.5 - 10/11/2020: version 3.1.12**Client**

- change IDE Visual Studio to 2019 version
- upgrade Windows Openssl version to openssl-1.1.1d

Hub

- Fixed a hang if lmx could not retrieve host id

Directory Front end

- Fixed an issue where build date was not displayed if a proxy was in error

1.6 - 07/07/2020: version 3.1.11**Hub**

- Fixed a time-out when restoring large evojuumps
- Added a missing client version check when connecting to a directory

Back end

- Fixed a replication issue with large assets
- Fixed a CouchDb service start issue on Windows

1.7 - 18/06/2020: version 3.1.10**Backend**

- Fixed a large build publication issue

1.8 - 02/06/2020: version 3.1.9**Client**

- Added the support of http proxy auto-config (PAC)

Backend

- Fixed a metadata synchronization issue

1.9 - 10/01/2020: version 3.1.8**Backend**

- Fix upload of evojuump files with admin credentials

1.10 - 17/12/2019: version 3.1.7**Backend**

- Add support of OpenID Provider configuration url

1.11 - 10/12/2019: version 3.1.6**Backend**

- Fixes data retention issue on proxy
- Fixes on update postgres role

Api Web

- fixes message API

1.12 - 07/11/2019: version 3.1.5**Backend**

- improve configuration selection reactivity

Client

- fixes computing dynamic low definition
- fixes stop of HD loading

1.13 - 02/10/2019: version 3.1.4**Backend**

- Bug fixes

1.14 - 22/08/2019: version 3.1.3**Client**

- Use configurations when exporting presentations
- Bug fixes

Hub

- Enable cancel during evojuump import
- Bug fixes

Backend

- Fix the start of 3D Juump Infinite service
- Improve services stability

- Fix timeout on importing large evojuump
- Bug fixes

Frontend

- Daemon informations
- Fix timeout on importing large evojuump

1.15 - 30/07/2019: version 3.1.2

Backend

- Bug fixes

1.16 - 18/07/2019: version 3.1.1

Client

- Improve loader's usage memory
- Improve loading heuristics

Backend

- protection of the migration process

Web Client

- Introduce cutting plane manipulator
- update documentation

1.17 - 26/06/2019: version 3.1.0

Client

- new camera position attached to Annotations of Annotation Tasks
- Hierarchy multi-branch id-card: new id-card feature for product structure analysis
- request existing evojuump download directly from Hub
- direct connexion thanks to openIDConnect
- improvement of lowdef computation and rendering
- automatic disconnection in case of non-use
- performances improvements
- unification of the offline and standalone modes

Backend

- direct connexion thanks to openIDConnect
- webhook to catch notification in case of errors
- metadata ranges type management
- optimisation of generation workflow
- safety improvements

Frontend

- new Directory frontend
- management of user rights based on Tags
- new setting page
- limitation of sending messages by proxy
- usage report generation from the frontend
- new fronted API
- support MS IE11 browser

1.18 - 25/02/2019: version 3.0.3

Client

- Fixed: issue with color bucket and export bucket returning empty list for filter with compound including date or numeric filter
- Fixed: glitches while rendering a presentation task
- Fixed: incorrect search results when a configuration is selected
- Fixed: unsorted configuration list
- Fixed: search results sometimes fail to display per instance information (count & red dots)
- Fixed: per-title sort not working in task manager

Backend

- Security fix:
 - update postgres
 - update elasticsearch
 - update java
 - apache configuration
- Added support for *linkmetadata* in PSConverter

1.19 - 11/02/2019: version 3.0.2

Client

- IdCard:
 - Improvement of IdCard with big array
- Presentation Task
 - Fixed predefined profile
- GUI
 - Fixed save/load session bug
- Display
 - Fixed picking in empty zone

Backend

- Fixed migration process 2.2 to 3.0
- improved the robustness of the generation process and replication

1.20 - 12/09/2018: version 3.0.1**Client**

- Main:
 - Fixed enumerated attribute bug
 - Fixed local DMU manager bug

1.21 - 24/08/2018: version 3.0.0**Client**

- Overall:
 - add/modify filter type
 - improve launching time
 - improve picking time
 - improve search/filter calculation time
 - improve client reactivity time on big DMUs
- Hub:
 - allow the cancellation of the connection
- IdCard:
 - Improvement of the full IdCard display
- Presentation Task
 - add arrow concept at the slide level
 - possibility to move the legends and title position
- Annotation Task
 - Add filtering by type
 - Add filtering by text
 - Add resize annotation
- GUI
 - show match information when searching
- Display
 - adding the front of the DMU at the browsing cube level
 - adding measure offset
 - adding access to the hidden visible space

Backend

- change of the ElasticSearch version
- change of the Postgres version
- replace couch by postgres for the metadata storage
- psconverter redesign
- API revision for the connector

Frontend

- allow the cancellation of a build

2 - Security guidelines

3D Juump Infinite relies on several server-side third-party software components that, like any piece of software, are subject to security flaws. The following paragraphs list the security measures to be taken in order to enforce security of each of these third-party components, together with a list of known vulnerabilities and how the listed measures can circumvent them.



The following guidelines, vulnerabilities and countermeasures apply to 3D Juump Infinite version 3.1 at the date of 2019/02/05.

2.1 - PostgreSQL

2.1.1 - Recommended Version

3D Juump Infinite requires PostgreSQL version 10.x. Make sure to apply the last security patches regarding PostgreSQL 10.

2.1.2 - Configuration guidelines

Access to the PostgreSQL clusters must be protected. It implies keeping the PostgreSQL cluster superuser credentials secret and protecting the system from malevolent access. In particular, the `pg_hba.conf` file should be carefully crafted to limit unprotected access to the cluster from unsecured and potentially compromised computers.

2.1.3 - Known vulnerabilities

- [\[CVE-2018-16850\]](#) Protect local access to the PostgreSQL host.
- [\[CVE-2018-10925\]](#) Upgrade the PostgreSQL version to include the last security fix.
- [\[CVE-2019-10164\]](#) Upgrade the PostgreSQL version to 10.9 or upper to include the security fix.
- [\[CVE-2019-10129\]](#) PostgreSQL 10.x not impacted. Upgrade the PostgreSQL version to 11.3 or upper if you use PostgreSQL 11.x.
- [\[CVE-2019-10130\]](#) Upgrade the PostgreSQL version to 10.8 or upper to include the security fix.
- [\[Scarlett Johanssons\]](#) Protect superuser credentials.

2.2 - Elasticsearch

2.2.1 - Recommended Version

3D Juump Infinite requires Elasticsearch version 6.x. Make sure to apply the last security patches regarding Elasticsearch 6.

ElasticSearch runs in a Java VM. There is no constraints on the Java version except for those listed in the Elasticsearch release note.

2.2.2 - Configuration guidelines

ElasticSearch is not directly exposed: it resides behind a reverse proxy. Only the required API end-points should be made available. In particular, access to sensible end-points like `/_snapshot` should be prohibited at the Web-server configuration level.

2.2.3 - Known vulnerabilities

- [\[CVE-2019-7611\]](#) Upgrade the Elasticsearch version to 6.6.1 or upper to include the security fix.
- [\[CVE-2019-7614\]](#) There is no workaround for this issue but 3d Juump Infinite will not be impacted.
- [\[Hijack\]](#) Protect PostgreSQL superuser credentials as they are used to give access to Elasticsearch.
- [\[Java\]](#) Upgrade your Java VM to include the last security patch.

2.3 - Apache CouchDB

2.3.1 - Recommended Version

3D Juump Infinite requires Apache CouchDB version 1.6.1.

2.3.2 - Configuration guidelines

CouchDB is not directly exposed: it resides behind a reverse proxy. Only the required API end-points should be made available. In particular, access to sensible end-points like `/_config` should be prohibited at the Web-server configuration level.

2.3.3 - Known vulnerabilities

- [\[CVE-2018-11769\]](#) Deny access to the `/_config` end-point.
- [\[CVE-2017-12635\]](#) Deny access to the `/_users` end-point.
- [\[CVE-2017-12636\]](#) Deny access to the `/_config` end-point.

2.4 - Apache Lounge Web-server

2.4.1 - Recommended Version

3D Juump Infinite requires Apache Lounge Web-server version 2.4.38, though any version sharing the same major version should be compatible.

2.4.2 - Known vulnerabilities

- [\[CVE-2019-10098\]](#) Upgrade the Apache Lounge Web-server version to 2.4.40 or upper to include the security fix.
- [\[CVE-2019-10092\]](#) Upgrade the Apache Lounge Web-server version to 2.4.40 or upper to include the security fix.
- [\[CVE-2019-10097\]](#) Upgrade the Apache Lounge Web-server version to 2.4.40 or upper to include the security fix.

2.5 - OpenSSL

2.5.1 - Recommended Version

3D Juump Infinite requires OpenSSL. the version of openssl may change depending on the system used. 3d juump infinite supports versions 1.0.2 1.1.0 and 1.1.1 of openssl.

2.5.2 - Known vulnerabilities

- [\[CVE-2019-1552\]](#) Upgrade the openssl version at least to 1.0.2t 1.1.0l or 1.1.1d to include the security fix.
- [\[CVE-2019-1547\]](#) Upgrade the openssl version at least to 1.0.2t 1.1.0l or 1.1.1d to include the security fix.
- [\[CVE-2019-1549\]](#) Upgrade the openssl version to 1.1.1d or upper to include the security fix.
- [\[CVE-2019-1563\]](#) Upgrade the openssl version at least to 1.0.2t 1.1.0l or 1.1.1d to include the security fix.

3 - Third-party software licenses

The details of licenses is available on the 3D Juump Infinite Third Party License manual provided with the software.

4 - Range of use

This annex summarizes the range of use of the software.

4.1 - Minimum requirements

The ∞Directory, ∞Server, ∞Proxy and ∞StaticProxy software run on any of the following operating systems:

- Microsoft Windows 7 and above, 64-bit version
- Linux Debian stretch, AMD64
- Linux Ubuntu 16.04 LTS (xenial) AMD64

Minimum hardware requirements are:

- Quad-core processor
- For the ∞Directory, ∞Proxy and ∞StaticProxy : 8GB of RAM
- For the ∞Server : 4GB of RAM by CPU Core (A Quad-core processor should have at least 16 GB of RAM).
- 1GB disk space for binaries + sufficient disk space for data (depending on your data sources)
- High-speed hard disk drive highly recommended

The ∞Client runs on Windows 7 (or higher) x64. Hardware accelerated 3D rendering requires an OpenGL 3.1 and GLSL 140 compatible graphics card.

For the application to run, the local computer must meet some minimal requirements:

- Windows 7/8/10 64-bit version
- 2GB RAM
- 4GB disk space for binaries and caches

4.2 - Supported input formats

3D Juump Infinite is able to process the following formats:

- ACIS (.sat) - all => R21
- ASC Medusa 3D (.asc)
- CADDs (explicit parts) & CAMU (._pd, ._ps) - 4 & 5
- FBX (.fbx)
- I-DEAS (.arc,.unv) - all => NX5
- IGES (.igs) 5.2 & 5.3
- Inventor (.ipt,.iam) - all => 2017
- CATIA V4 (.model,.dlv,.exp,.session) - all 4.xx
- CATIA V5 (.CATPART,.cgr,.CATProduct) - R10 - R26
- CATIA V6 (.3Dxml) - 2011x => 2013x
- 3D Experience (.3Dxml) - 2014 => 2015x
- JT-Format (JtOpen) (.jt) 7.0 => 10.2
- Matra Euclid 3 (.e3i) - 3.2
- Nastran (.nastran)
- NX Unigraphics (.prt) - 11 => NX11
- OBJ (.obj, .mtl)
- Parasolid XT-Format (.x_t) - all => 28
- ProEngineer (.asc,.prt, .neu) - part files: 13 => Creo 4 (F000) / neutral files: 13 => WF5
- Rhino 3D (.3dm)
- ROBCAD (.rf)
- Solidworks (.sldprt,.sldasm) - 99 => 2017
- STEP AP203, AP214 and AP242 (.stp) - 203/214/242
- Straessle EUKLID (.edx)
- STL (.stl)
- VDA (.vda)
- VRML (.wrl, .wrz, .vrm) - 97
- 3DS (.3ds)

4.2.1 - Geometry

3D Juump Infinite only accepts surface information. All other geometric informations are ignored (in particular, vector and point data are not supported).

4.2.2 - Metadata

3D Juump Infinite is able to extract product structure, including external references. It also extracts textual, numeric and datum metadata, plus any combination of lists and maps of the above.

4.2.3 - Annotation

3D Juump Infinite is able to retrieve both FTA and PMI. It is limited in the number of options it supports regarding the FTA and PMI representation. Supported data includes:

- text content,
- fonts,
- colors,
- shapes (limited to square and flags),

Symbols (NOA) are not automatically read from input files and require a manual customization to circumvent the lack of universal symbol catalog.

All glyphs visible in a given Annotation View must fit in a 2048x2048 pixels image.

4.3 - Supported output formats

4.3.1 - Geometry

Supported output formats for geometry export are:

- OBJ
- VRML
- FBX
- JT
- STEP¹³+JT
- STEP+WRL
- WRL+WRL
- WRZ+WRZ

Exported geometries are tessellated. Annotations are not exported.

4.3.2 - Image

Supported output formats for image export are:

- PNG
- JPEG
- TIFF

Transparency support depends on the output format.

4.3.3 - Metadata

Supported output formats for metadata export are:

- CSV (comma separated)
- CSV (semi-colon separated)

¹³ STEP AP242 Part 21

- XML
- JSON

4.3.4 - Presentation

Supported output formats for presentation export are:

- Markdown
- HTML
- HTML with DZSlide embedded viewer
- ODP
- JSON

Note: Markdown and ODP do not support rich-text, thus when slide comments containing rich-text are found, they are exported as raw-text instead.

4.4 - Limits

Known limits include:

- Minimum number of assemblies¹⁴: 1
- Minimum number of single parts¹⁵: 1
- Maximum number of parts¹⁶: 536 870 912
- Maximum number of links¹⁷: 536 870 912
- Maximum number of ids¹⁸: 4 294 967 294
- Maximum number of geometry instances¹⁹: 16 777 215
- Maximum number of distinct materials: 32 768
- Maximum number of annotation views: 1 073 741 823
- Maximum number of annotation types: 1 024
- Maximum number of rasters: 4 096
- Maximum number of annotations visible at the same time: 262 143
- Maximum number of conf declared by the connector: 10 000

¹⁴ Structure document with children

¹⁵ Structure document with geometric representation

¹⁶ PartMetadata documents

¹⁷ LinkMetadata documents

¹⁸ Distinct *id* fields

¹⁹ One source model instantiated at one world position

5 - Export control classification

The Software, which integrates dual use information security items of American origin (ECCN 5D992.c <10%), is subject to the US Export Administrative Regulations (EAR) 15 C.F.R. part 730 et seq. for the country Group E:1 and E:2 which are, at the date of the License Terms and Conditions: Iran, North Korea, Sudan, Syria and Cuba. In particular, the User shall not use, export or re-export the Software in those countries and with end users or for end uses in breach of the US export control regulations.

The Software has been the subject of a declaration of operations relating to a means of cryptology to the ANSSI (Declaration N ° 17070363). However, the Software does not come under Regulation (EC) N ° 428/2009 of May 5, 2009, setting up a Community regime for the control of exports, transfer, brokering and transit of dual-use items, as confirmed by the Direction Générales des Entreprises / Services des Biens à Double-Usage Goods in his mail N ° FR 80404.