# BEYOND PLATFORM

## Architecture Dossier

Les diagrammes de ce dossier sont en majorités réalisés à partir de l’outil de modélisation UML PLANTUML (http://plantuml.com/)

*Nota Bene :* Cette version du document est un export of de la documentation en ligne.

La dernière version est disponible ici: <https://github.com/sxd-platform/byd-all-documentation/tree/master/02.byd%20-Platform/1.ArchitectureDossier>

Les chapitres suivants forment le dossier d’architecture :

* [Introduction](./0000.Introduction.md)
* [Table des matières](./0001.TableOfContent.md)
* [Glossaire](./0002.Glossary.md)
* [Contexte du système](./0100.SystemContext.md)
* [Modèle de cas d’utilisation](./0200.UseCaseModel.md)
* [Exigences](./0300.Requirements.md)
* [Modèle de composants](./0400.ComponentsModel.md)
* [Architecture Overview](./0500.ArchitectureOverview.md)
* [Décisions d’Architecture](./0600.ArchitectureDecisions.md)
* [Modèle opérationnel de déploiement](./0700.DeploymentModel.md)
* [Modèle de donnnées](./0800.DataModel.md)
* [Liste des COTS](./0900.COTSList.md)

|  |  |
| --- | --- |
| Rôle | NAME - Email |
| Auteurs | Frederic FADDA frederic.fadda@fr.ibm.com / Sylvain WILBERT - s\_wilbert@fr.ibm.com |
| Relecteurs |  |
| Approbateurs |  |

Dossier d’Architecture - BEYOND PLATFORM - Novembre 2019

# Table des matières

* [Introduction](./0000.Introduction.md)
* [Table des matières](./0001.TableOfContent.md)
* [Glossaire](./0002.Glossary.md)
* [Contexte du système](./0100.SystemContext.md)
* [Modèle de cas d’utilisation](./0200.UseCaseModel.md)
* [Exigences](./0300.Requirements.md)
* [Modèle de composants](./0400.ComponentsModel.md)
* [Architecture : vue d’ensemble](./0500.ArchitectureOverview.md)
* [Décisions d’Architecture](./0600.ArchitectureDecisions.md)
* [Modèle opérationnel de déploiement](./0700.DeploymentModel.md)
* [Modèle de donnnées](./0800.DataModel.md)
* [Liste des COTS](./0900.COTSList.md)

Dossier d’Architecture - BEYOND PLATFORM - Novembre 2019

<[Introduction](./0000.Introduction.md) | [Table des matières](./0001.TableOfContent.md) | [Contexte du système](./0100.SystemContext.md) >

# Glossaire

## Acronymes

La liste des acronymes utilisés dans le dossier d’architecture est :

|  |  |
| --- | --- |
| **ACRONYME** | **DESCRIPTION** |
| **API** | Application Programming Interface ; interface exposant un service |
| **COTS** | Commercial Off The Shelf: par opposition à tout développement spécifique, Solution logicielle disponible sur le ‘marché’ |
| **CRUD** | Create, Read, Update, Delete. Fonctions de base pour le stockage persistant de données |
| **IdP** | Identity Provider ; fournisseur d’identité pour l’authentification des utilisateurs d’une système |
| **IHM** | Interface Homme - Machine |
| **NoSQL** | Not Only SQL : système de gestion de base de données non relationnelles |
| **REST** | Representational State Transfer, style d’architecture logicielle permettant de décrire les services exposés par une API |
| **SGBDR** | Système de Gestion de Base de Données Relationnelles |

## Termes

Les termes suivants sont utilisés dans le dossier d’architecture :

|  |  |
| --- | --- |
| **TERME** | **DESCRIPTION** |
| **Container** | Instance de virtualisation d’un système d’exploitation (par exemple Docker) |
| **Microservice** | Une architecture par microservices est un style d’architecture logicielle où la complexité apparente des applications est diminuée par la décomposition des processus en plusieurs entités indépendantes et faiblement couplées. Ils sont souvent exposés au travers d’API REST, reposant sur le format JSON et le protocole Http(s) |

Dossier d’Architecture - BEYOND PLATFORM - Novembre 2019

< [Introduction](./0000.Introduction.md) | [Table des matières](./0001.TableOfContent.md) | [Contexte du système](./0100.SystemContext.md) >

<[Glossaire](./0002.Glossary.md) | [Table des matières](./0001.TableOfContent.md) | [Modèle de cas d’utilisation](./0200.UseCaseModel.md) >

# Diagramme de contexte du système

La plateforme BEYOND sera dénommée **SYSTEME** dans le reste du document.

*It is composed of several components linked over REST API in order to promote lose coupling among components of the system though finely integrated.*

Diagramme de contexte

Diagramme de contexte

Le **SYSTEME** a pour vocation de permettre l’acquisition, l’hébergement la visualisation et l’analyse de données liées au cycle de vie d’infrastructures du domaine de la construction.

Pour ce faire, Le **SYSTEME** s’appuie sur un ensemble de solutions applicatives (BEYOND-ASSET, BEYOND-REALITY, BEYOND-DIGITALSITE, BEYOND-MONITORING…) auxquelles il fournira : - Des services transverses de gestion des données (DATA FACTORY), des utilisateurs et des droits d’utilisation ; la liste des services est en cours de construction dans le backlog de développment du SYSTEME. - Des composants réutilisables.

Le **SYSTEME** interagit avec les acteurs suivants :

* Des **Administrateurs d’utilisateurs et de rôles** qui gèrent les l’accès au SYSTEME et aux applications reposant sur le SYSTEME.
* *Liste non exhaustive*

Le **SYSTEME** interagit avec les systèmes externes suivants :

* Des **IdP** de clients entreprises pour la délagation de l’authentification.
* *Liste non exhaustive*

Dossier d’Architecture - BEYOND PLATFORM - Novembre 2019

<[Glossaire](./0002.Glossary.md) | [Table des matières](./0001.TableOfContent.md) | [Modèle de cas d’utilisation](./0200.UseCaseModel.md) >

< [Exigences](./0300.Requirements.md) | [Table des matières](./0001.TableOfContent.md) | [Architecture : vue d’ensemble](./0500.ArchitectureOverview.md) >

# Modèle de composants

Les composants sont des éléments indépendants, interchangeables du **SYSTEME**. Ils respectent et réalisent une ou plusieurs interfaces fournies et requises, qui déterminent le comportement des composants.

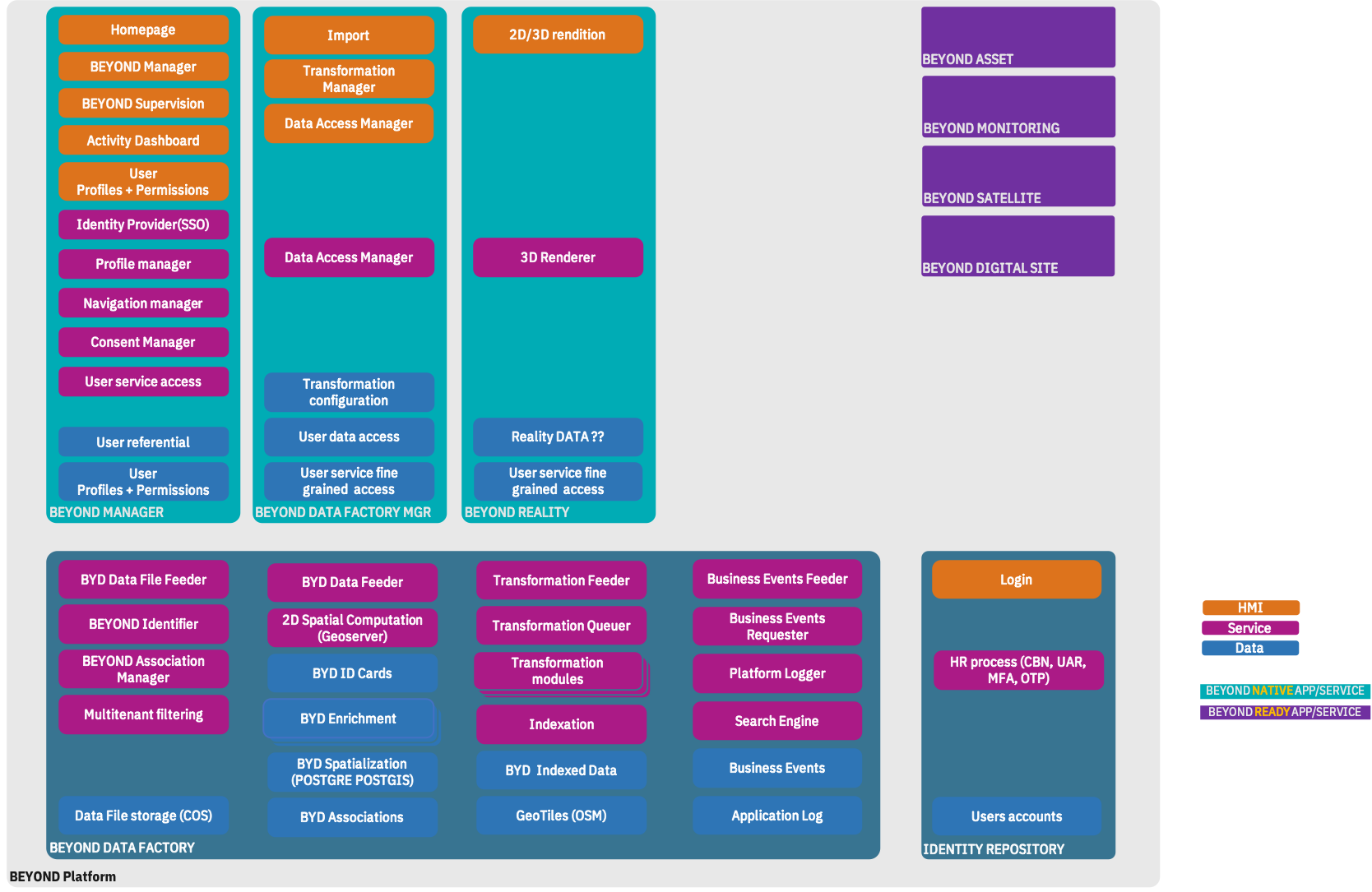
Un composant du **SYSTEME** doit stocker ses données selon les concepts et principes d’architecture de la plate-forme : l’utilisation d’un identifiant unique Beyond-ID est obligatoire, la gestion des revisions importantes en terme de traçabilité.

Le **SYSTEME** peut être décomposé en sous-systèmes (BEYOND DATA FACTORY, BEYOND MANAGER, BEYOND DATA FACTORY MANAGER, IDENTITY REPOSITORY…), un ou plusieurs composants constituent un sous-système, selon le degré de granularité adopté.

Dans une approche “cloud native”, les sous-systèmes sont implémentés sous forme de **microservices**. Ces microservices exposent (par défaut) des interfaces de type services REST pour promouvoir le couplage faible. La description de ces services respecte la spécification openAPI 2.0 (SWAGGER) ; cette description est accessible via un fichier au format YAML.

Les solutions applicatives (BEYOND-ASSET, BEYOND-REALITY, BEYOND-DIGITALSITE, BEYOND-MONITORING…) reposant sur le **SYSTEME** doivent respecter les principes ci-dessus dans le cadre de leur utilisation de services et composants du **SYSTEME** et de leur exposition de services et composants pour les autres sous-systèmes ou solutions applicatives de la plate-forme.

## Vue Statique



Modèle de composants - vue statique

### Packages et composants

Le **SYSTEME** peut être représenté à l’aide d’un diagramme de packages (les sous-systèmes) montrant les dépendance entre les packages.

#### Packages

Modèle de composants - packages

Modèle de composants - packages

Les packages suivants peuvent être définis :

|  |  |
| --- | --- |
| Package | Description |

#### Composants

Les composants suivants peuvent être définis :

*A détailler sur la base des éléments des images ci-dessus*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Package** | Composant | Domaine | Rôle | Responsabilitées |

## Vue Dynamique

### Diagrammes de processus

*A détailler*

#### Authentification (SSO)

ActivityDiagram - Authentification

ActivityDiagram - Authentification

#### Intégration de données

ActivityDiagram-DataUpload

ActivityDiagram-DataUpload

Dossier d’Architecture - BEYOND PLATFORM - Novembre 2019

< [Exigences](./0300.Requirements.md) | [Table des matières](./0001.TableOfContent.md) | [Architecture : vue d’ensemble](./0500.ArchitectureOverview.md) >

# Modèle de données

## Data Collections

### Parameters

* **objectID** : Identifiant du document
* **revisionID**: Identifiant de la version de l’objet
* **parameterID**: Identifiant du parametre
* **categoryID**: Categorie du paramètre.
* **creationDateTime**: Horodatage de la création du document
* **creationBy**: Identifiant du créateur
* **status**: Statut courant de l’objet

Entité “Valeur du paramètre” sous forme de liste

* **valueid**: identifiant des valeur Entité “Labels” sous forme de liste pour une valueid donnée:
* **language**: Langue
* **label** : Libellé
* **isDefault**: Est-ce le libellé par defaut (de création de l’objet)

# BEYOND PLATFORM - Dossier d’architecture

Le présent document détaille les éléments d’architecture de BEYOND PLATFORM, plateforme de services génériques de l’écosystème **BEYOND**.

## Démarrage

Pour récupérer localement cette documentation, cloner le repository “byd-all-documentation” via GIT. Se référer à : https://github.com/sxd-platform/byd-all-documentation

## Version

|  |  |  |
| --- | --- | --- |
| Version | Description | Auteur |
| **0.1.0** | Version Initiale | FFA |