

3DEXPERIENCE

Configuration & Development

Overview

R2024x



About this course

Course objectives

Upon completion of this course, you will be have an overview about:

- 3DEXPERIENCE platform architecture
- Access Model implementation
- Data Model Configuration
- Presentation Layer Development
- Business Logic Configuration

Target audience

- Solution Architect

Prerequisites

- Knowledge of 3DEXPERIENCE platform Architecture Overview course



Disclaimer

The content of this training material has been created with great care. Dassault Systèmes takes no responsibility for the topicality, correctness, completeness or quality of the data and information. Liability claims against Dassault Systèmes , which refer to material or immaterial nature caused by use or disuse of the data and information provided through the use of incorrect and incomplete data and information are basically excluded.

Dassault Systèmes reserves the right to change parts of this training material and information, or the entire offer without prior notice, add to, delete or cease publication temporarily or permanently.

All data and information are offered free and without obligation.

Caution / Warning

This document is based on a variety of internal DS contents for 3DEXPERIENCE platform R2024x FD01 (FP 2405), sometimes validated in technical platforms.

In some cases, this document contains forward looking statements based on current expectations, forecasts and assumptions that involve risks and uncertainties.

Upon further communication this document is classified to be DS Confidential. It may not be distributed without DS authorization.

Changes in Course with Respect to R2023x

- ❑ Some of the most important value additions made to the R2024x release of this course, with respect to the R2023x course, are as below:
- ❑ Enhanced the following topics
 - Widget Configuration
 - 3DPassport User Management
 - Collaborative Spaces Control Center App

TABLE OF CONTENT (1/2)

Preface

1. 3DEXPERIENCE platform Introduction
 - a. 3DEXPERIENCE platform Architecture
 - b. Data Types
2. Access Management
 - a. Organization Modeling & Secure Data Accesses
 - b. 3DPassport Control Center App
 - c. User Group
3. Data Layer Configuration Overview
 - a. Administrative and Business Objects
 - b. System Objects
 - c. Introduction to Data Model
 - d. Collaborative Spaces Control Center App
 - e. Data Model Customization App
 - f. Enterprise IP Schema Specialization App

TABLE OF CONTENT (2/2)

4. Presentation Layer Configuration Overview
 - a. Widget Configuration
 - b. OOTB Widget Configuration
 - c. Internationalization
5. Integration Layer Configuration Overview
 - a. EKL
 - b. Automatic Business Logic – Precommit Openings
 - c. CAA
 - d. Web Service
 - e. Event Publishing
6. Legacy Utilities
 - a. Presentation Layer Configuration
 - b. Integration Layer Configuration

Preface

- The primary objective of this course is to introduce and provide a basic outline of the scopes for **3DEXPERIENCE** platform configuration and development, both for Native Apps and Web Apps
- This course will not provide in-depth technical or functional details.
- If the trainee is interested to know more about the various topics that have been covered in this course, then it is recommended to go through the following courses that are available on **3DEXPERIENCE** University
 - **3DEXPERIENCE Configuration & Development Fundamentals**
 - **3DEXPERIENCE Widget Development Fundamentals**
 - **Discover Enterprise Knowledge Language**

3DEXPERIENCE platform Introduction

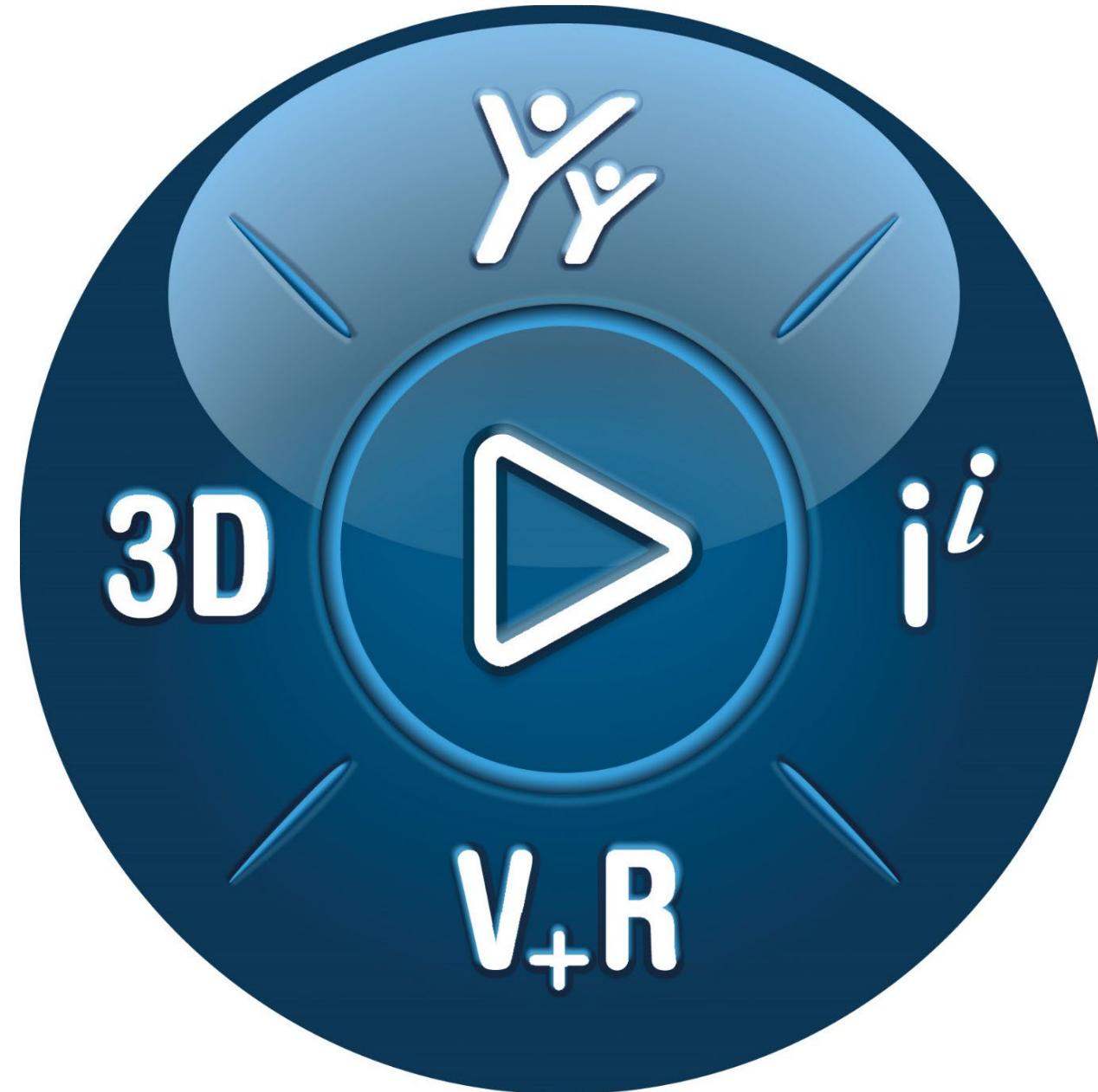


—

After completing this lesson you will be able to:

- Know about 3DEXPERIENCE platform architecture
- Be familiar with the data type definitions





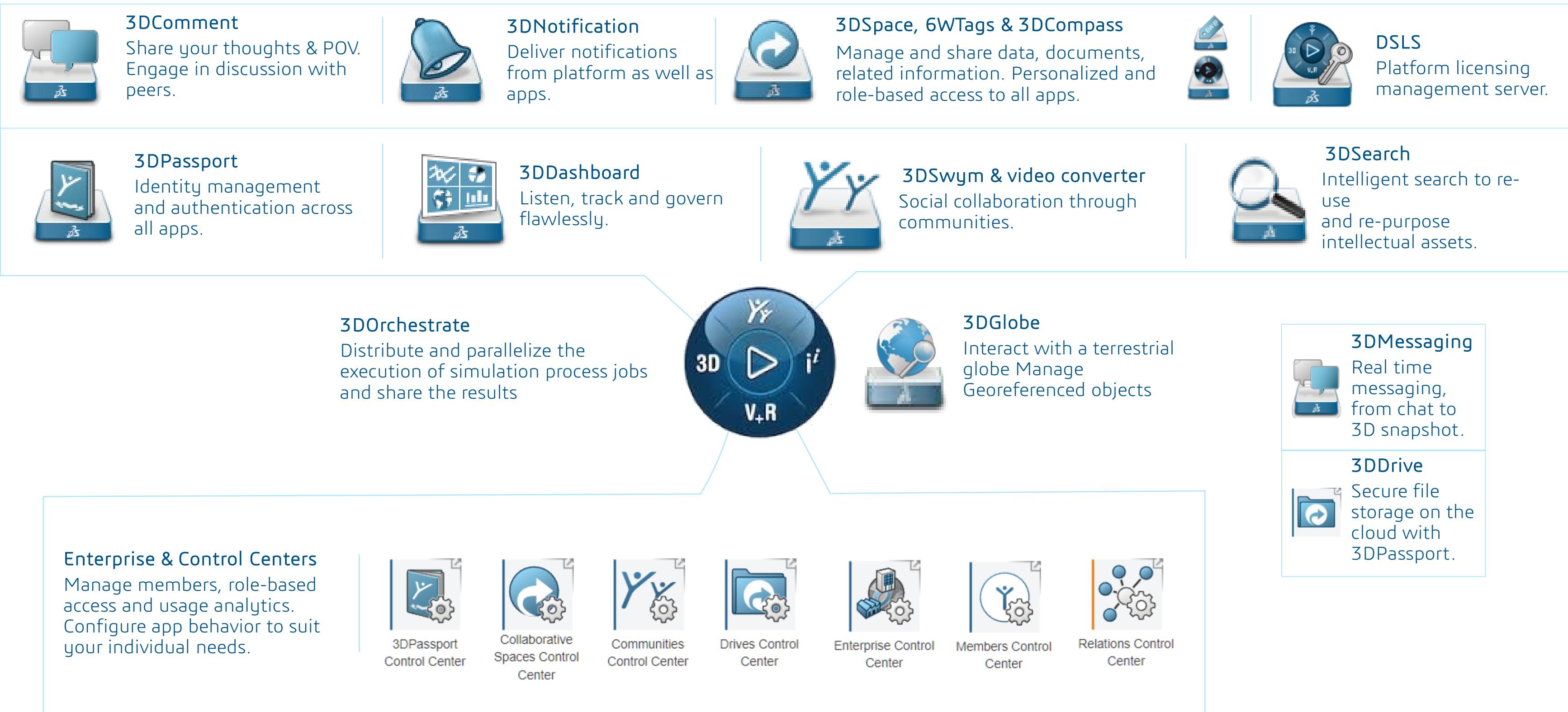
3DEXPERIENCE®

Introduction

In this chapter, we will know about architecture and components of 3DEXPERIENCE platform.

We will get an overview about platform Configuration and Development capabilities.

3DEXPERIENCE platform Services and Apps



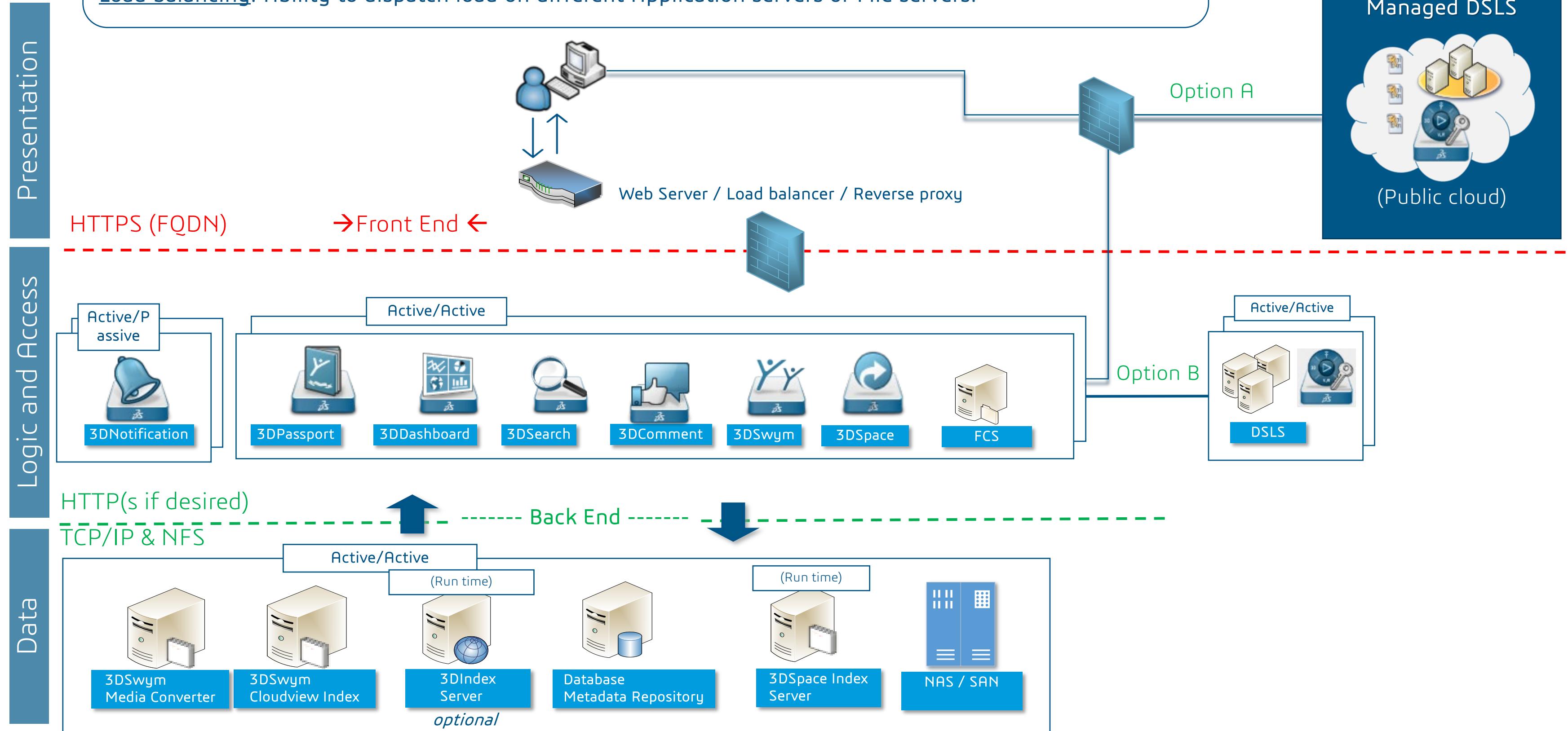
3DEXPERIENCE platform Architecture – On Premises



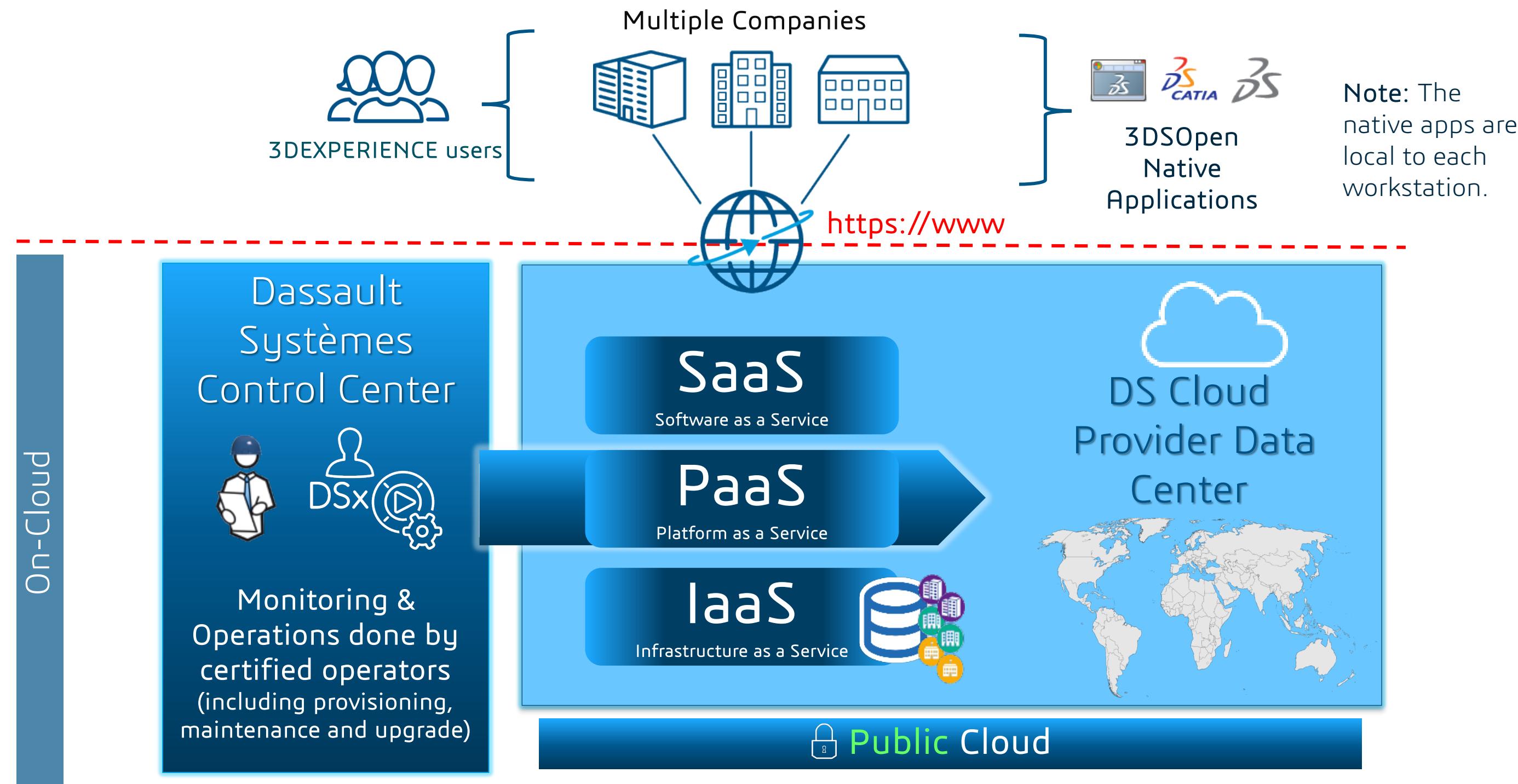
Modularity: Is the degree to which a system's components may be separated and recombined.

Fail-over: In case of failure of an application server transaction is dispatched on another application server.

Load balancing: Ability to dispatch load on different Application servers or File servers.



3DEXPERIENCE platform Architecture – On Cloud



3DEXPERIENCE Configuration and Development Capabilities Overview



3DEXPERIENCE environment can be customized by several means, each with their own scope, prerequisites and limitations:

| Area of Customization | Customizable Components | Collaborative Spaces Control Center | Data Model Customization | Enterprise IP Schema Specialization | VS Code | Data Setup |
|-----------------------|-------------------------|-------------------------------------|--------------------------|-------------------------------------|---------|------------|
| Data Model | Type | ✓ | ✓ | ✓ | ✗ | ✗ |
| | Attribute | ✓ | ✓ | ✓ | ✗ | ✗ |
| | Relationship | ✗ | ✗ | ✗ | ✗ | ✗ |
| | Extension | ✓ | ✓ | ✓ | ✗ | ✗ |
| User Interface | Native App UI | ✗ | ✗ | ✗ | ✗ | ✗ |
| | Custom Widget | ✗ | ✗ | ✗ | ✓ | ✗ |
| Business Logic | Business Rule | ✗ | ✓ | ✓ | ✗ | ✓ |

- ❑ *Visual Studio Code* is standard code IDE. Developer may use other IDEs of their choice.
- ❑ *Data Model Customization* app has an option named **Business Rule Editor** for working with Business Rules
- ❑ *Enterprise IP Schema Specialization* app (**Private Cloud** only) can be used to import server business rules of a pre-commit trigger Resource Set

Different Possibilities of Configuration and Customization – On Premise



| What you can do | Web Apps | Native Apps | Dashboard Apps |
|--|---|--|---|
| Data Model Modification – Creation of new Types and Attributes | <input type="checkbox"/> Data Model Customization App <input type="checkbox"/> Collaborative Spaces Control Center App | <input type="checkbox"/> Data Model Customization App <input type="checkbox"/> Collaborative Spaces Control Center App | <input type="checkbox"/> Data Model Customization App <input type="checkbox"/> Collaborative Spaces Control Center App |
| Data Model Modification – Relationships, Policies, P&O, etc. | <input type="checkbox"/> Members Control Center <input type="checkbox"/> Collaborative Spaces Control Center App | <input type="checkbox"/> Members Control Center <input type="checkbox"/> Collaborative Spaces Control Center App | <input type="checkbox"/> Members Control Center <input type="checkbox"/> Collaborative Spaces Control Center App |
| Access to the data | <input type="checkbox"/> P&O | <input type="checkbox"/> P&O <input type="checkbox"/> Security Masks | <input type="checkbox"/> P&O |
| Configuration of the UI | <input type="checkbox"/> Configurable Components <input type="checkbox"/> MQL/TCL Scripts | <input type="checkbox"/> Business Rules <input type="checkbox"/> EKL <input type="checkbox"/> CAA <input type="checkbox"/> VB | <input type="checkbox"/> HTML 5 <input type="checkbox"/> JavaScript <input type="checkbox"/> CSS |
| New business Logic | <input type="checkbox"/> JPO <input type="checkbox"/> Triggers | | <input type="checkbox"/> REST Web Services |

Customized Types should preferably be created as sub-Types of OOTB Types only, to avoid migration and upgrade issues in the future

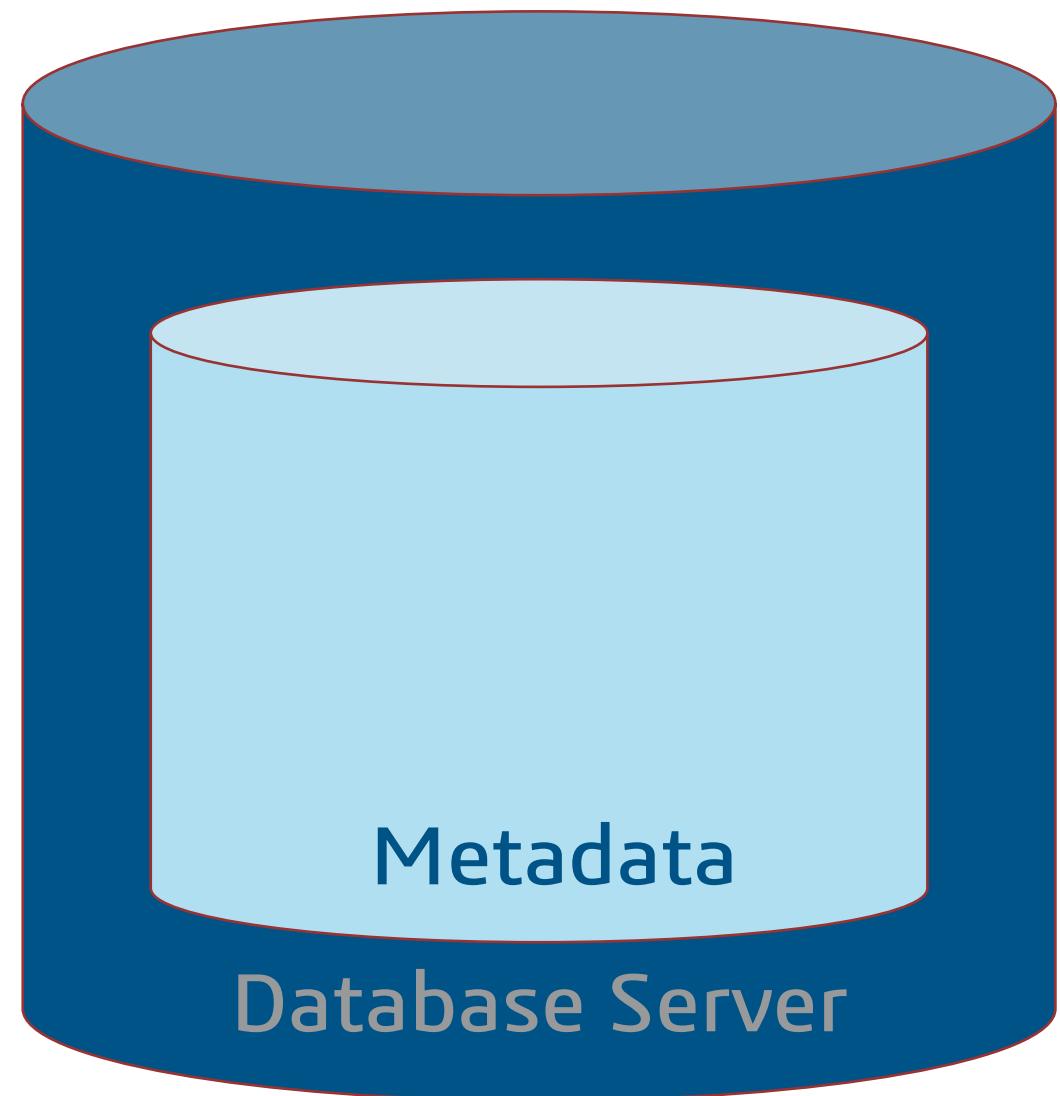
Different Possibilities of Configuration and Customization – On Cloud



| What you can do | Web Apps | Native Apps | Dashboard Apps |
|--|--|---|--|
| Data Model Modification - Creation of new Types and Attributes | <input type="checkbox"/> Enterprise IP Schema Specialization App <input type="checkbox"/> Collaborative Spaces Control Center App | <input type="checkbox"/> Enterprise IP Schema Specialization App <input type="checkbox"/> Collaborative Spaces Control Center App | <input type="checkbox"/> Enterprise IP Schema Specialization App <input type="checkbox"/> Collaborative Spaces Control Center App |
| Data Model Modification - Relationships, Policies, P&O, etc. | <input type="checkbox"/> Members Control Center <input type="checkbox"/> Collaborative Spaces Control Center App | <input type="checkbox"/> Members Control Center <input type="checkbox"/> Collaborative Spaces Control Center App | <input type="checkbox"/> Members Control Center <input type="checkbox"/> Collaborative Spaces Control Center App |
| Access to the data | <input type="checkbox"/> P&O | <input type="checkbox"/> P&O | <input type="checkbox"/> P&O |
| Configuration of the UI | <input type="checkbox"/> NA | <input type="checkbox"/> Business Rules <input type="checkbox"/> EKL <input type="checkbox"/> CAA (CA) <input type="checkbox"/> VB | <input type="checkbox"/> HTML 5 <input type="checkbox"/> JavaScript <input type="checkbox"/> CSS |
| Customization of the UI | <input type="checkbox"/> NA | | |

- Customized Types should preferably be created as sub-Types of OOTB Types only, to avoid migration and upgrade issues in the future
- CAA is available in Control Availability mode
- Enterprise IP Schema Specialization App is available only on Private and Dedicated Cloud and not available on Public Cloud

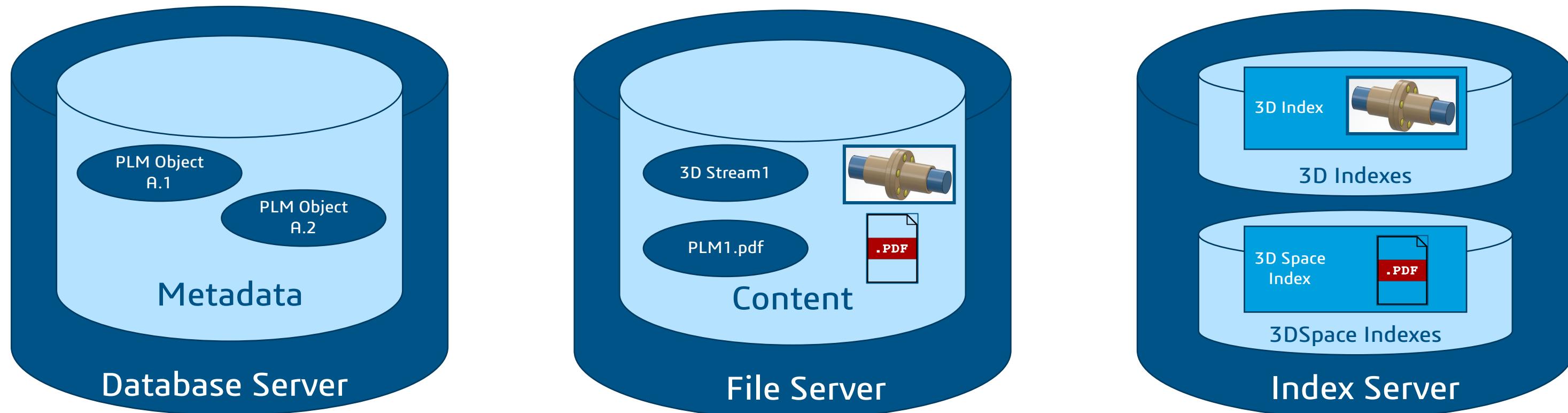
Data Types



Introduction

In this chapter, we will learn about different form of data managed in **3DEXPERIENCE** platform

- ❑ **Metadata** for objects and relationships as well as business logic
 - Webtop Data Model: Attribute, Type, Interface, etc.
 - Native Client Data Model: Reference, Instance, Port, Connection, RepReference, RepInstance, etc.
 - 3DSwym: Communities, Blogs, Wikis
- ❑ **Metadata Content**: Files, including 3D representations, streams, office documents, etc.
 - Documents, Static 2D/3D Thumbnails
 - Streams for Representations and Ports (3D Shapes)
 - CATParts, CATMaterials, etc.
 - Videos and other 3DSwym media
- ❑ **Indexes** to accelerate access to data and geometry
- ❑ Each of these data types should have dedicated storage for backup, recovery and performance





Here are the key takeaways from this lesson:

- The 3DEXPERIENCE platform provides a unique visual interface, as well as a clear and unified access to key services for all 3DEXPERIENCE platform apps. It also applies a graphic interface that is compliant with touch devices.

- Metadata is stored in **Database**, whereas physical files are stored in **File Store**

Access Management



—
After completing this lesson you will:

- Know about P&O elements and Baseline Roles
- Know about 3DPassport Control Center App
- Know about User Group utility



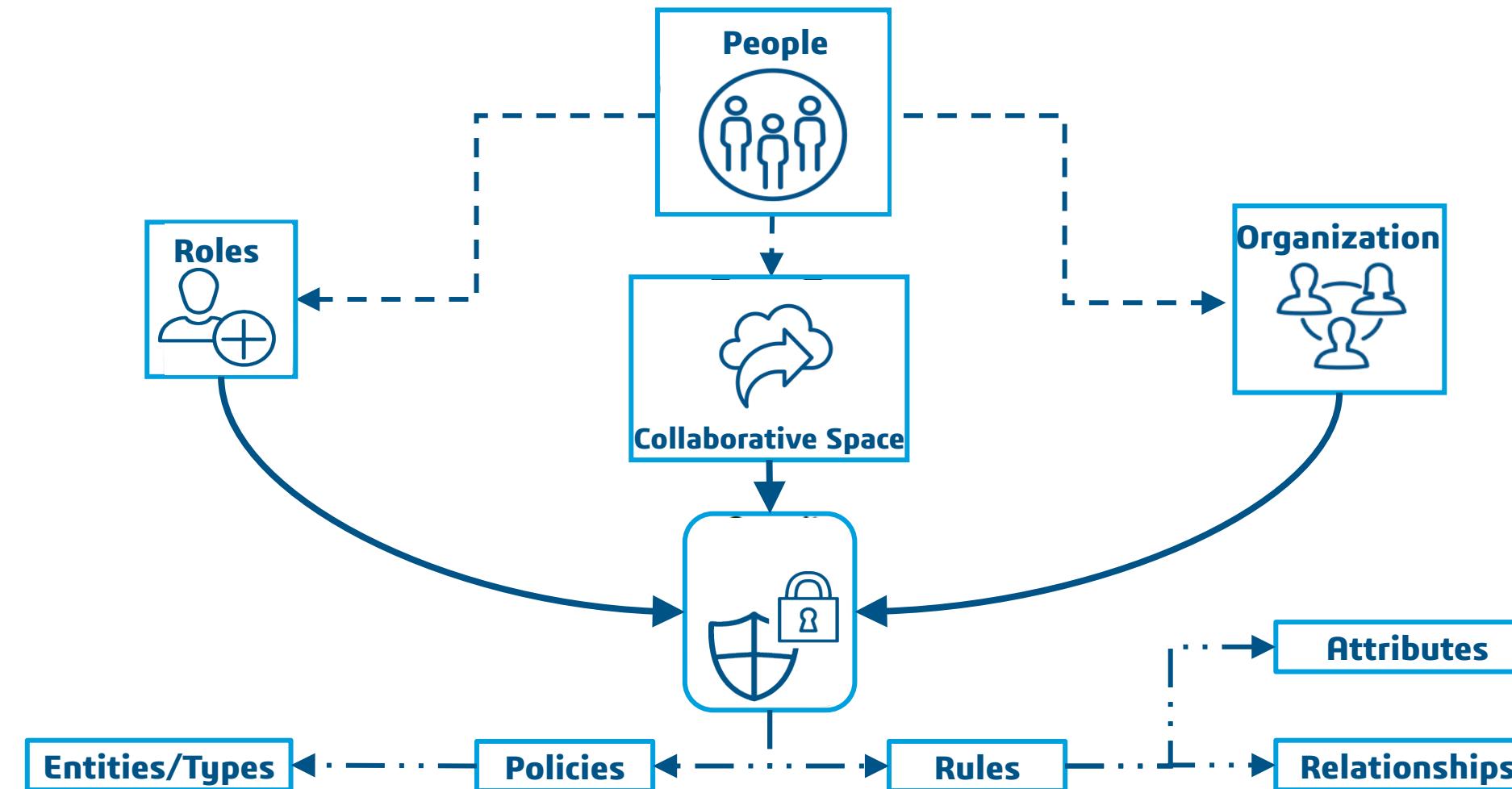
Organization Modeling & Secure Data Accesses



Introduction

In this chapter, we will learn about Organization Modeling & Secure Data Accesses

- ❑ Organization Modeling & Secure Data Accesses is the main security mechanism to manage authorizations for 3DSpace
 - This access model is also popularly known as “People and Organization” (P&O)
- ❑ P&O consists of five key concepts at its foundation:
 - Roles, People, Organization, Collaborative Spaces and Security Contexts.





- A Person object represents a logical user in the **3DEXPERIENCE** platform.
- It has a User-ID and a Password to login into the platform.
 - It can be synchronized/integrated with an LDAP server using the 3DPassport.
- A Person is always related to at least one (or multiple) security context(s), which is a combination of :
 - an Organization
 - a Role
 - a Collaborative Space (PnOProject)
- A Person is assigned product licenses ("Named User" licenses).
 - License assignments are managed in the Member Management dashboard by Administrators.
 - The licenses usually appear to the end user as Roles in the compass.

Admin Platform - ADMIN_PLATFORM

admin_platform@smtp.mytraining.com
2-Factor Authentication: Not Active / Last connection: < 10 days
Administrator

Roles Additional Apps Settings

View roles granted View all Search Roles

3DSwymer
Collaborative Industry Innovator
Engineering BOM Manager

3DEXPERIENCE | ENOVIA Collaboration

ME COMPANY WORLD

My Roles

3DSwymer
Collaborative Industry Innovator
Engineering BOM Manager
Platform Manager

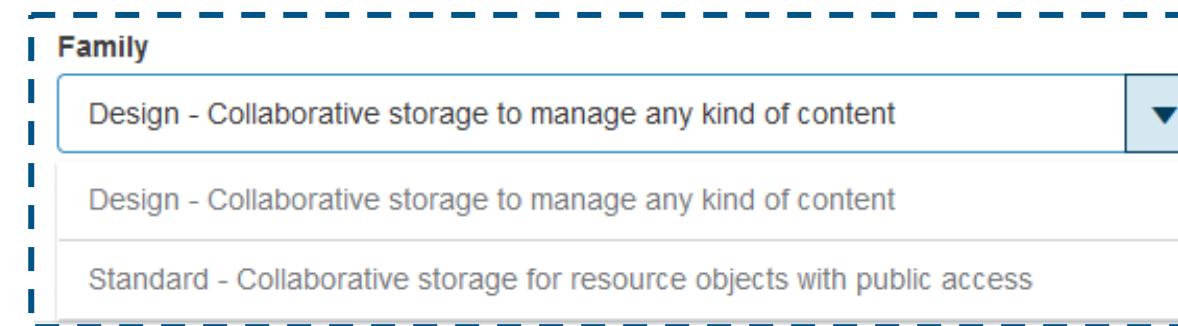
- These *Roles* are not to be confused with the PnO Roles described in later slides.



- A **Collaborative Space** represents a shared space where people can work together on content/data.
- Collaborative Spaces are mainly used for access management.
 - Used for making up Security Contexts, and stamping objects' Ownership vectors
- Collaborative Spaces organize content to manage who can access that content, and when users from other collaborative spaces can access the content.
- In R2013x , it was called '**Project**' (a word inherited from VPM/CAD world) or **PnOProject**.
 - In R2014/x and later it became Collaborative Space in the GUI but in the data model PnOProject is still used as the business type.
- Collaborative Spaces, or the type *PnOProject* are not to be confused with Program Central's **Project Space** type.
 - Collaborative Spaces do not propose tasks/planning features, and typical features relevant to project management domain.

- Based on Content Visibility, there are three kinds of Collaborative Spaces:
 - Public Design Collaborative Space
 - Protected Design Collaborative Space
 - Private Design Collaborative Space

- Collaborative Space families can be used as templates to create other collaborative spaces
 - Standard (StandardTeam): Applicable for Public and Protected content only
 - Design (DesignTeam): Applicable for Public, Protected and Private content



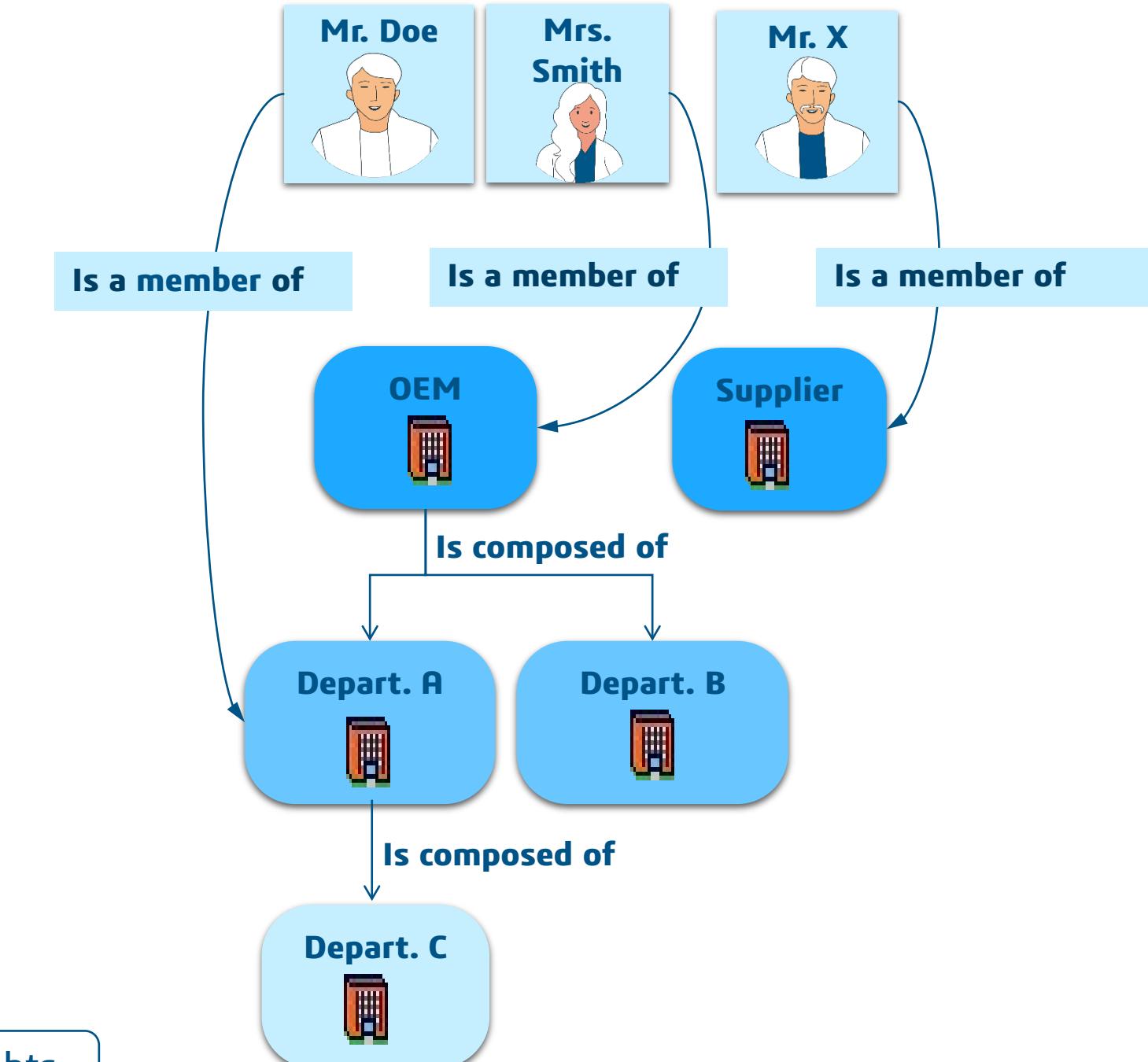
- From the list of collaborative spaces, you can edit the visibility to a collaborative space.

- You can make Private Collaborative Space as Protected collaborative space or Public Collaborative Space and Public Collaborative Space as Protected Collaborative Space or Private Collaborative Space.

- From **Manage Collaborative Spaces** tab you can increase or reduce the visibility of Collaborative Space.



- ❑ Organization object's purpose is to define a group of persons belonging to a same operational entity.
- ❑ There are four types of organizations in the platform:
 - Company
 - Subsidiary
 - Business Unit
 - Department
- ❑ Organizations can be connected in a hierarchical manner.
 - A child organization is considered as part of its parent organization but not the opposite.
 - Example: A context on Depart. A will give you the access rights as a similar context on OEM, in addition to the more specialized access privileges for Depart. A



The *member of* Relationship and type of organization has no influence on the security rights.

A context in a given organization will provide access to data belonging to the same organization.

Organization represents the disciplines of a company which does not change frequently. It does not represent other organizational structures, such as HR structure, that may change frequently



- ❑ Roles represents the area of work, or access level of a person.
- ❑ Depending on the access level, three separate categories of roles are available.
 - *Applicative Roles*
 - VPLMViewer (Reader)
 - VPLMExperimenter (Contributor)
 - VPLMCreator (Author)
 - VPLMProjectLeader (Leader)
 - VPLMSecuredCrossAccess (Public Reader)
 - *Administrative Roles*
 - VPLMProjectAdministrator (Owner)
 - VPLMAdmin (Administrator)
 - *Restricted Roles*
 - 3DSRestrictedAuthor
 - 3DSRestrictedContributor
 - 3DSRestrictedLeader
 - 3DSRestrictedOwner
 - 3DSRestrictedReader

Add members of your 3DEXPERIENCE Platform to this collaborative space.

Enable restricted responsibilities
 Assign credentials with a given organization

| | |
|----------------|--------------------------------|
| Public Reader | <i>Enter a name or a login</i> |
| Reader | <i>Enter a name or a login</i> |
| Contributor | <i>Enter a name or a login</i> |
| Author | <i>Enter a name or a login</i> |
| Leader | <i>Enter a name or a login</i> |
| Leader & Owner | <i>Enter a name or a login</i> |
| Owner | <i>Enter a name or a login</i> |

Add members Cancel

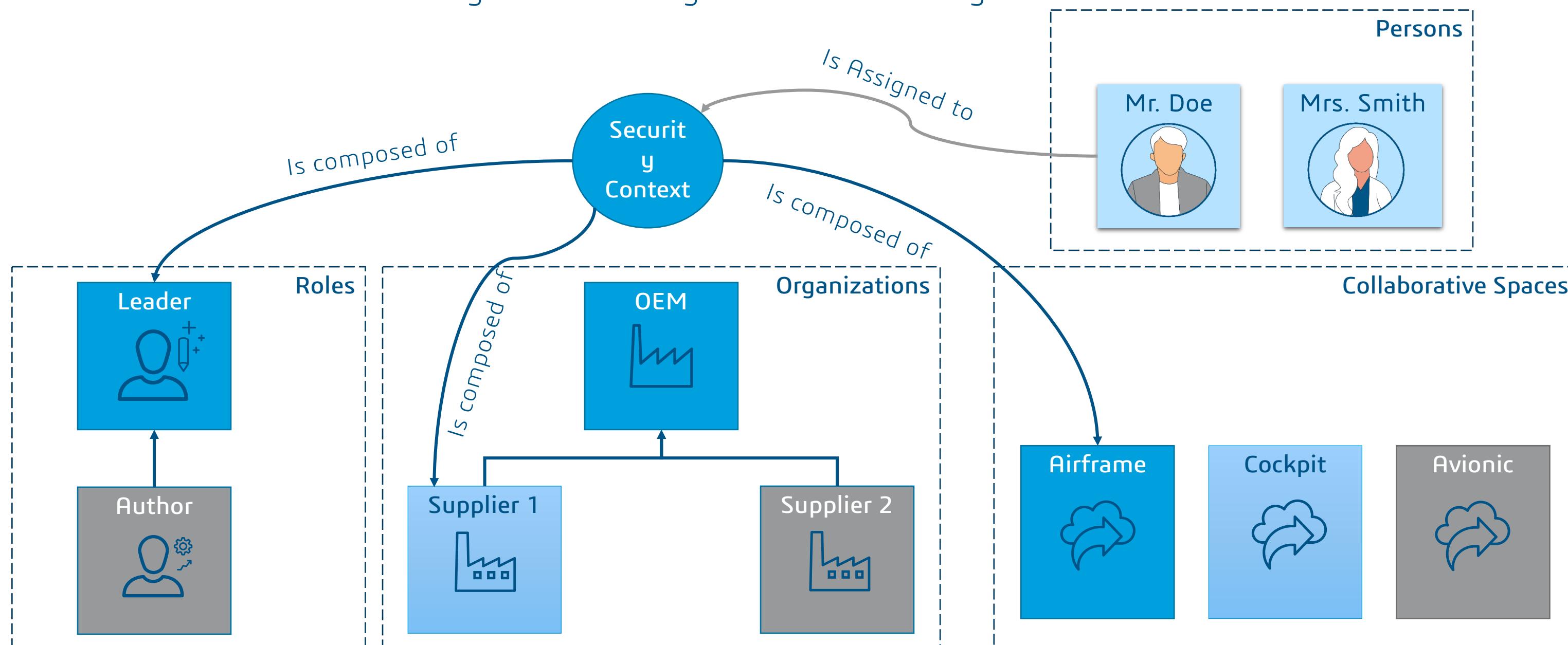
Restricted Roles are assigned to partners to allow employees of other companies access limited to content in the Collaborative Space and organization of the user's credentials



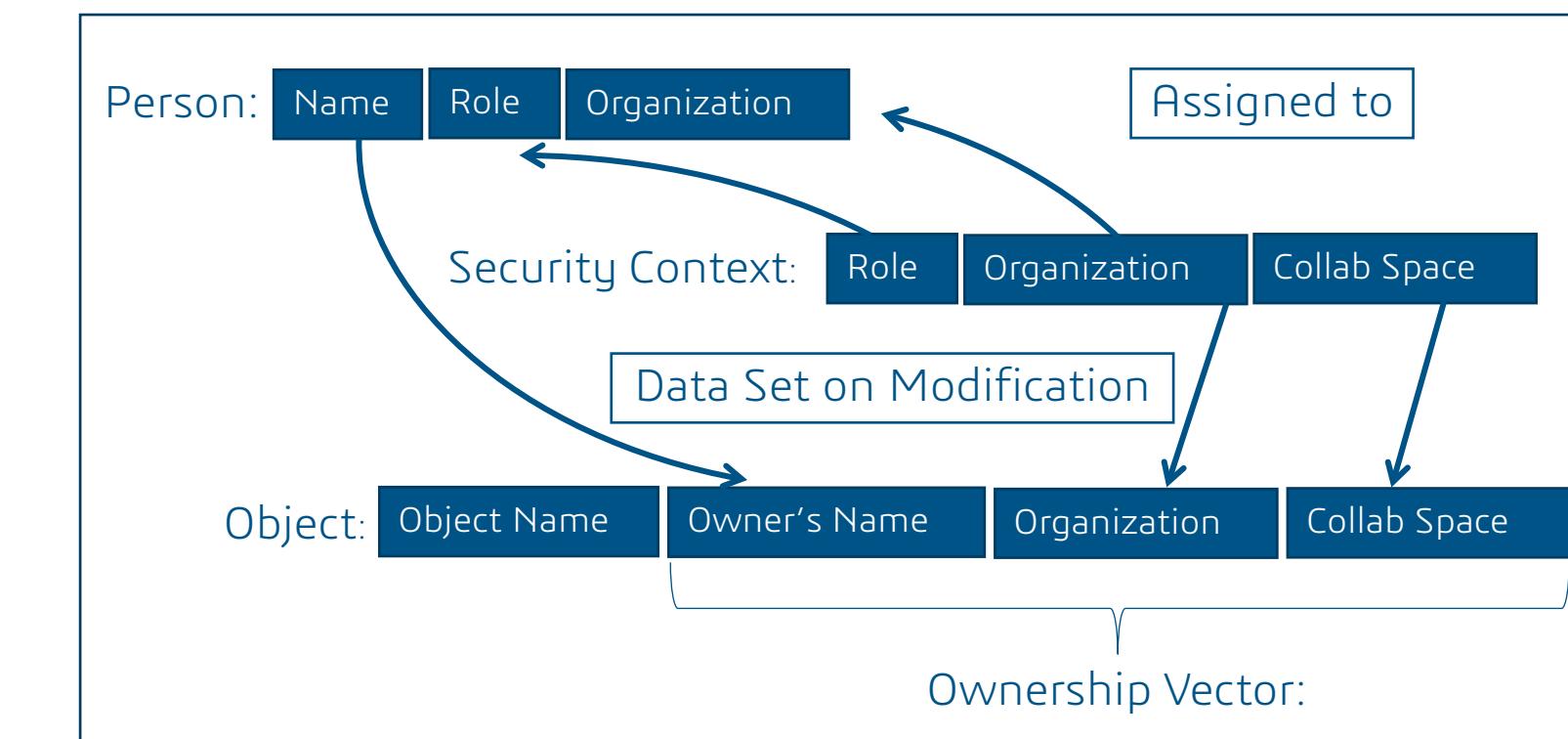
□ 3DSpace Baseline Roles

- **Reader:**
 - Allows a user Read-access to both Public and Private Content within a Collaborative Space he is given access to.
- **Contributor:**
 - Allows a user to Create Evaluation content like DMU review, interference Simulations, structural analysis, kinematics analysis, manufacturing simulations on any Content within a Collaborative Space he is given access to.(to (regardless of the user ownership of the objects)
- **Author:**
 - Allows a user Write/Modify-access to any Content he owns within a Collaborative Space he is given access to.
- **Leader:**
 - Allows a user Write/Modify-access to any Content as well as Promote access to any Change Controlled Content within a Collaborative Space he is given access to.
- **Owner:**
 - User/Collaborative Space that is responsible for the Content.
- **Administrator:**
 - Create Administrative Resource content such as Material Domain, Resource Set, etc.
 - Manage company resources such as collaborative spaces, organizations, and persons, etc.
- **Public Reader:**
 - Allows a user Read-access to only Public content within a Collaborative Space.
- **Restricted Access Roles:**
 - The user assigned with restricted roles can access the objects except the restricted content that are created by the Reader, Contributor, Author, Leader, and Owner roles of the same collaborative space and organization

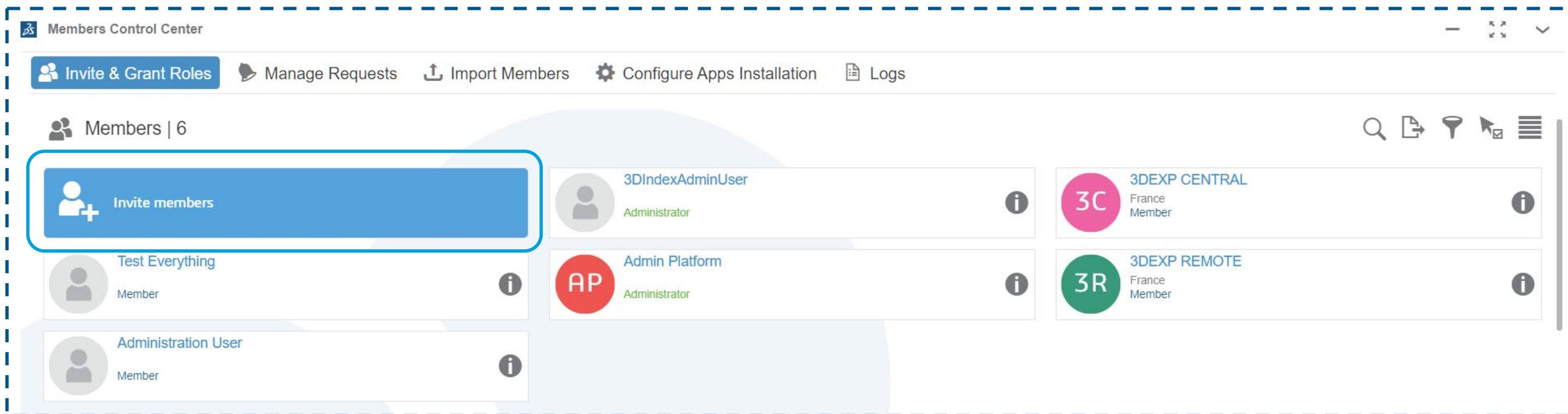
- A **Security Context** is defined by the combination of one Organization, one Role and one Collaborative Space (PnOProject).
- A *Security Context* is assigned to a user/person.
 - A user can have several Security Contexts assigned simultaneously.



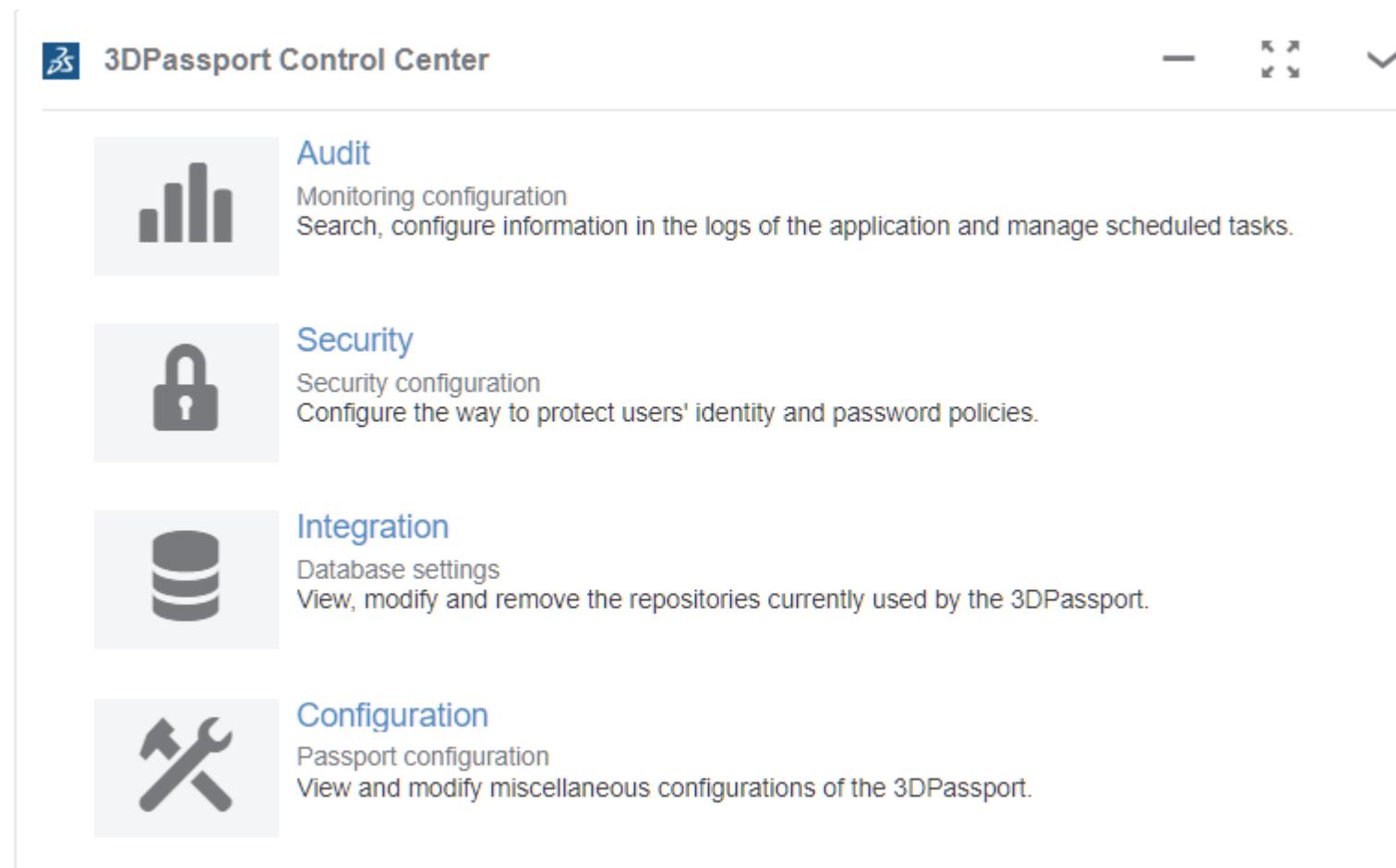
- A user can have only one Active Security Context at a time.
 - The active Security Context is chosen from the list of contexts assigned to the user.
 - Active security context is sometimes referred to as **Preferred Context** or **Default Context**.
- The active security context can be chosen at login, it can be changed from the login menu later if necessary (2014x and onward).
 - The context which is active when the user logs out will be the active one when logging back in.
 - Active Security Context can also be set using the VPLMPosImport batch (On Premise only).
 - More information regarding setting Preferred Context using VPLMPosImport can be found in User Assistance (On Premise only).
- The *Active Security Context* is used to set the initial data ownership, when a user creates data. The initial data ownership is also referred to as the **Primary Ownership Vector (POV)**.
 - The POV of an object is defined by the combination of 3 properties:
 - A User (Owner)
 - A Collaborative Space (project)
 - An Organization
 - Access to an object are based on data ownership and user's Security Contexts



- ❑ For a Person to have access to the data stored in the database, he/she must be set up as a valid user.
- ❑ The process can be done in different ways.
 - Users can be added via **Members Control Center** app (preferred way).
 - The users can be added via a bulk loading procedure (*VPLMPosImport* tool – *On Premise* only)
- ❑ Regardless of the method used, the individuals must have a valid username and password.
- ❑ Users must exist as an Active Employee of an Active Company to use the Application.
- ❑ The Person must be created in 3DPassport for successful login.



3DPassport Control Center App



Introduction

In this chapter, we will get an overview about 3DPassport Control Center app which is used to manage 3DPassport Service



- ❑ 3DPassport service provides authentication and single-sign-on (SSO) service to all **3DEXPERIENCE** platform services (except for FCS).
 - ❑ 3DPassport SSO **cannot** be replaced in any way by an enterprise SSO solution.
 - Authentication can only be delegated.
 - ❑ 3DPassport administration can be done using **3DPassport Control Center** widget.
 - ❑ New users can be created in 3DPassport using below options
 - *Web console* – Create your **3DEXPERIENCE** ID option on 3DPassport Login page
 - *Batch Utility* – PassportUserImport.bat can be used by Administrator.
 - *Web Service* – Using the OOTB User Account API we can create, update, delete or recover user accounts.
- ❑ Users can be created through web console by Administrator or user themselves can create their own profile.
 - ❑ Batch Option is only available for Administrator.
 - ❑ Web Service option is only meant for Administrator.



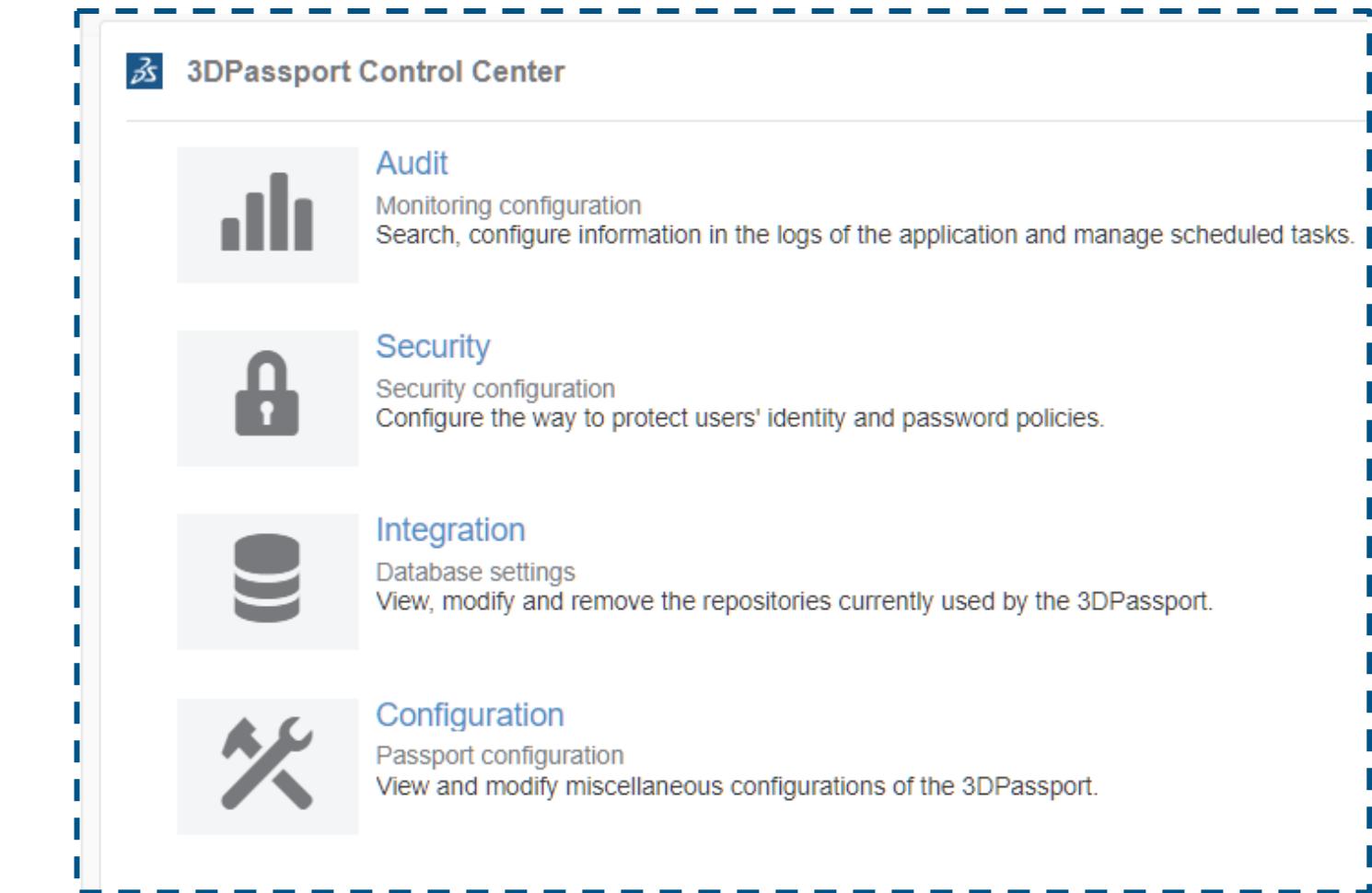
- ❑ 3DPassport administration can be done using **3DPassport Control Center** widget.

- ❑ *3DPassport Control Center* consists of below sections:

- Audit
- Security
- Integration
- Configuration

- ❑ **3DPassport Control Center** can be accessed via the following:

- **3DPassport Control Center Widget**
- **Administration Tools** button on 3DPassport landing page
- **HTTPS URL:**
<https://<PAS>/3dpassport/admin-tools/v2/login>
- **HTTP URL** (bypassing Apache web server):
<http://<PAS>:<port>/3dpassport/admin-tools/v2/login>



All Users with Administrator rights in 3DSpace can administrate 3DPassport.

User Group

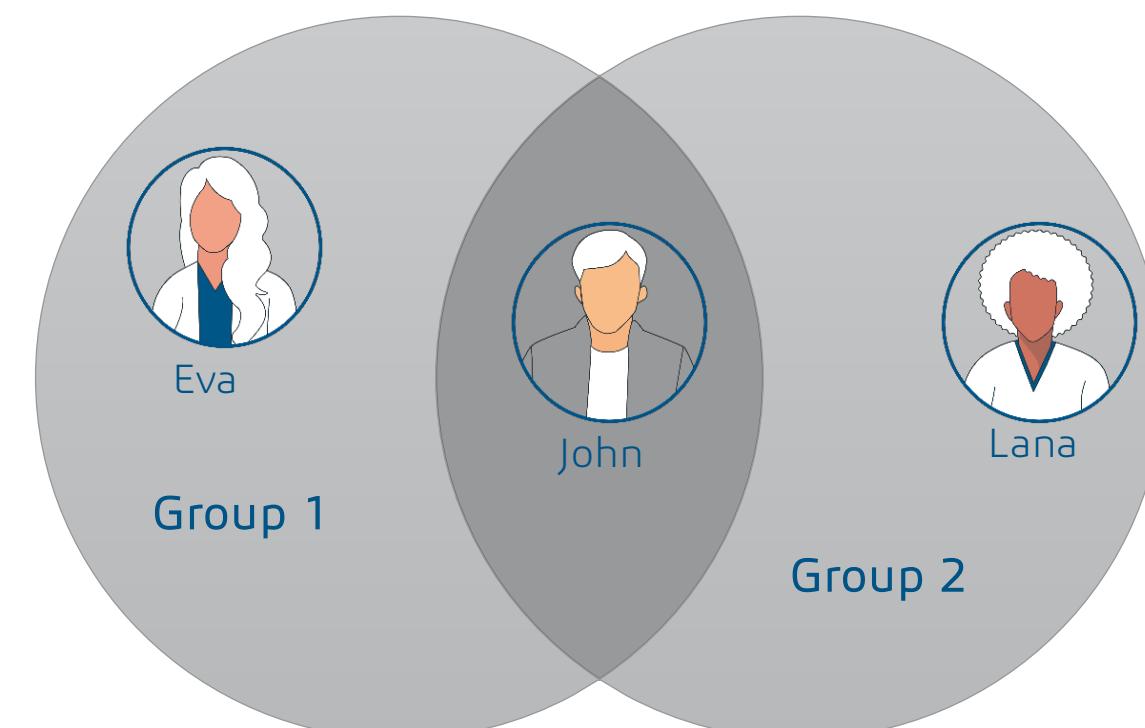
The screenshot shows a user interface for managing user groups. At the top, there is a navigation bar with a back arrow and a search bar containing the text 'User Groups'. Below this, a list item for 'CDF Group' is shown, featuring a blue icon of three people, the title 'CDF Group', the owner 'Admin Platform', and '0 Member'. Below the list item are three tabs: 'Members', 'Properties' (which is selected and highlighted in blue), and 'Responsibilities'. A pencil icon indicates that changes can be made to the properties. Below the tabs, detailed information about the group is listed in a table:

| | |
|-------------|---|
| Title | CDF Group |
| Description | Group of Person from CDF Department |
| Owner | Admin Platform |
| Name | auto_11681878009195 |
| Group URI | <uuid:ab5e6dd4-7449-42d3-8e76-92695ce2b5ba> |

Introduction

In this chapter, we will get an overview about User Group utility.

- ❑ User Group is a group of users defined within a **3DEXPERIENCE** platform instance.
- ❑ User Groups are an integral part of P&O.
- ❑ One User can be members of several Groups.
- ❑ Groups are managed at platform level to be leveraged by other services of the platform
- ❑ **3DSpace** service of **3DEXPERIENCE** platform supports the concept of Group of users to ease access control deployment
- ❑ For **onPremise** implementation, you can work with user group by using:
 - **User Group** utility
 - **VPLMPosImport** Batch
- ❑ For **onCloud** implementation, the **User Groups** dashboard app may be used to work with User-Groups.

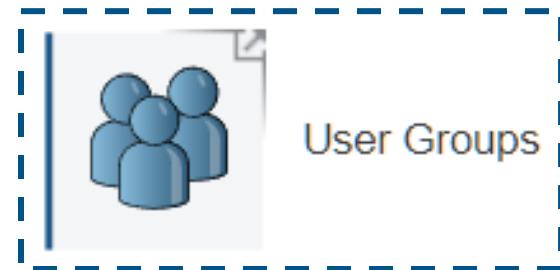


All users can view all *Public* User Groups

User Group – On-Cloud



- ❑ User Groups dashboard app allows you to manage a centralized list of groups that can be used by any user-group compatible app available in the **3DEXPERIENCE** platform.
- ❑ User Groups provides the following capabilities:
 - List the user groups you can use. By default, only the groups you own are displayed but you can view other groups also.
 - Create user groups that can be used by any member of the platform in their apps and edit these groups.
 - If you are a platform administrator:
 - update user groups owned by others.
 - create and update user groups through a file import.
- ❑ You can create **Private groups** to restrain group visibility to the selected users.
- ❑ Only platform administrators, group viewers, owners and managers can view and manage private group.
- ❑ By default, the group is private. Private groups cannot be seen by the members of the group.
- ❑ You can export one or more groups as a csv file that you can reuse to import and create new or update existing ones.



Create Group

Group Type

Group with Private Visibility

EN

Name *

Training

Description

Description

Share this private group with its members

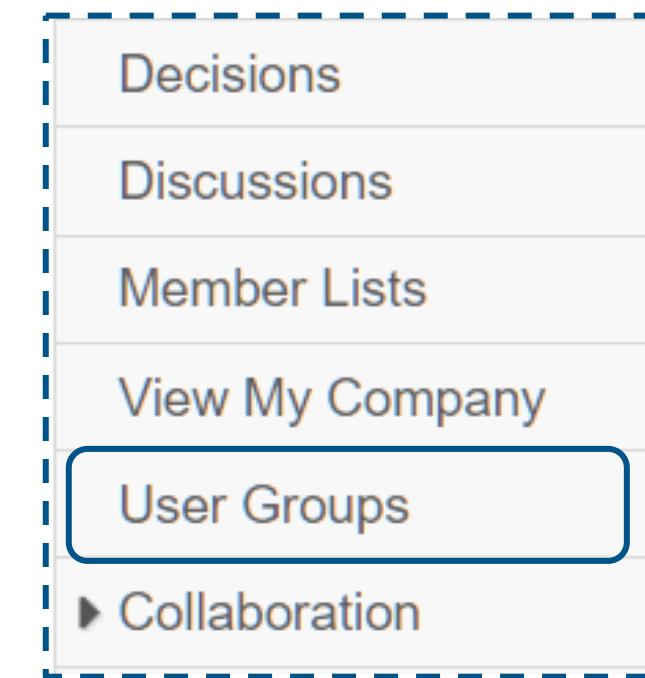
Viewer

Create Cancel

User Group – On-Premise



- To access User Group, launch **Collaboration and Approvals** app and go to *Navigation Pane* → *User Groups*
- Using User Group Utility one can :
 - Create User Group
 - Add the members to User Group
 - Remove the members from User Group
 - Modify the property of User Group
 - Define Access Rights
 - Export the User Group
- User groups can be used as task assignees in a route.





Here are the key takeaways from this lesson:

- Organization Modeling & Secure Data Accesses (P&O) is the main security mechanism to manage authorizations for 3DSpace
- P&O consists of five key concepts at its foundation:
 - *Roles, People, Organization, Collaborative Spaces and Security Contexts.*
- 3DPassport Control Center consists of below tools to control behavior and configuration of 3DPassport:
 - *Audit*
 - *Security*
 - *Integration*
 - *Configuration*
- User Group is a group of users defined within a 3DEXPERIENCE platform instance. User Groups are an integral part of P&O.

Data Layer Configuration Overview



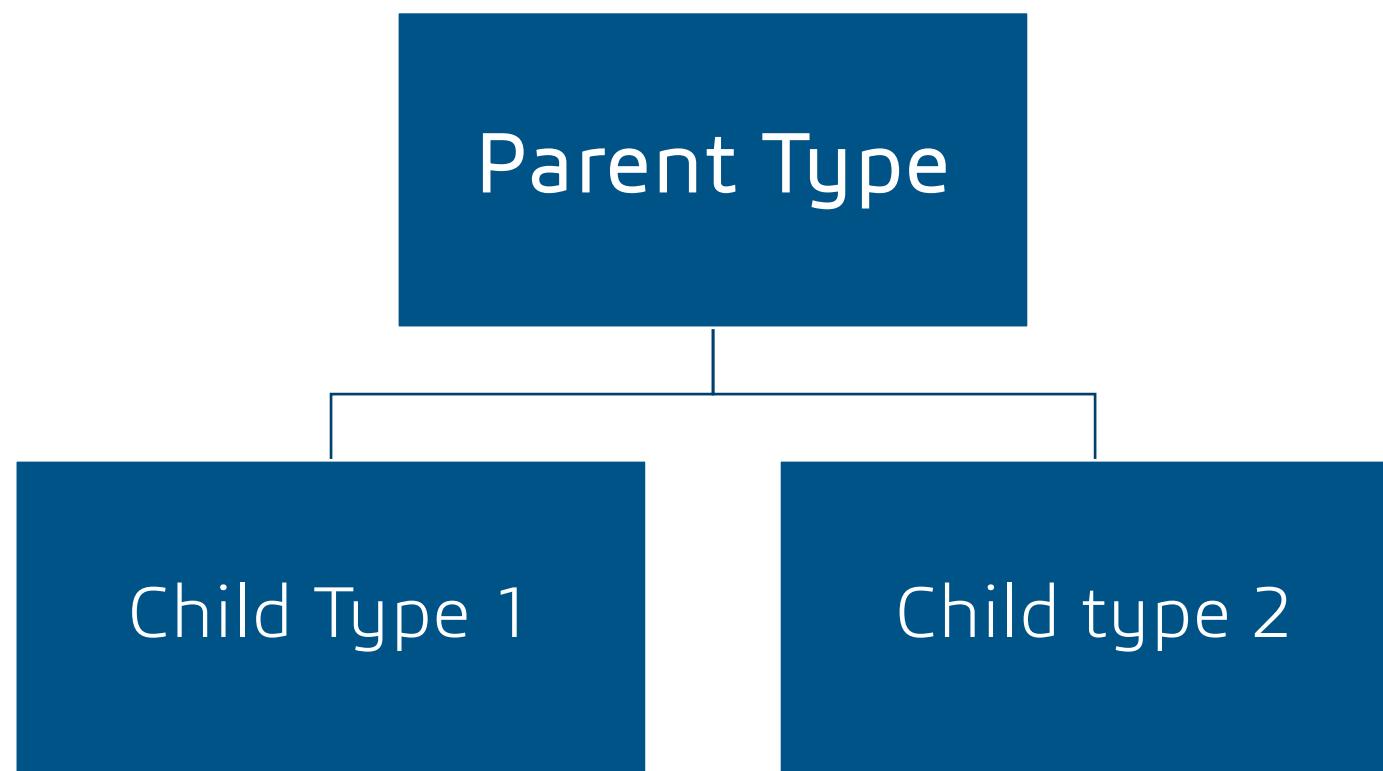
—

After completing this lesson you will:

- Know about Data Model
- Know about Collaborative Space Control Center Dashboard App
- Know about the Data Model Customization App for On-Premise Environment
- Know about Enterprise IP Schema Specialization app for On-Cloud environment



Administrative and Business Objects



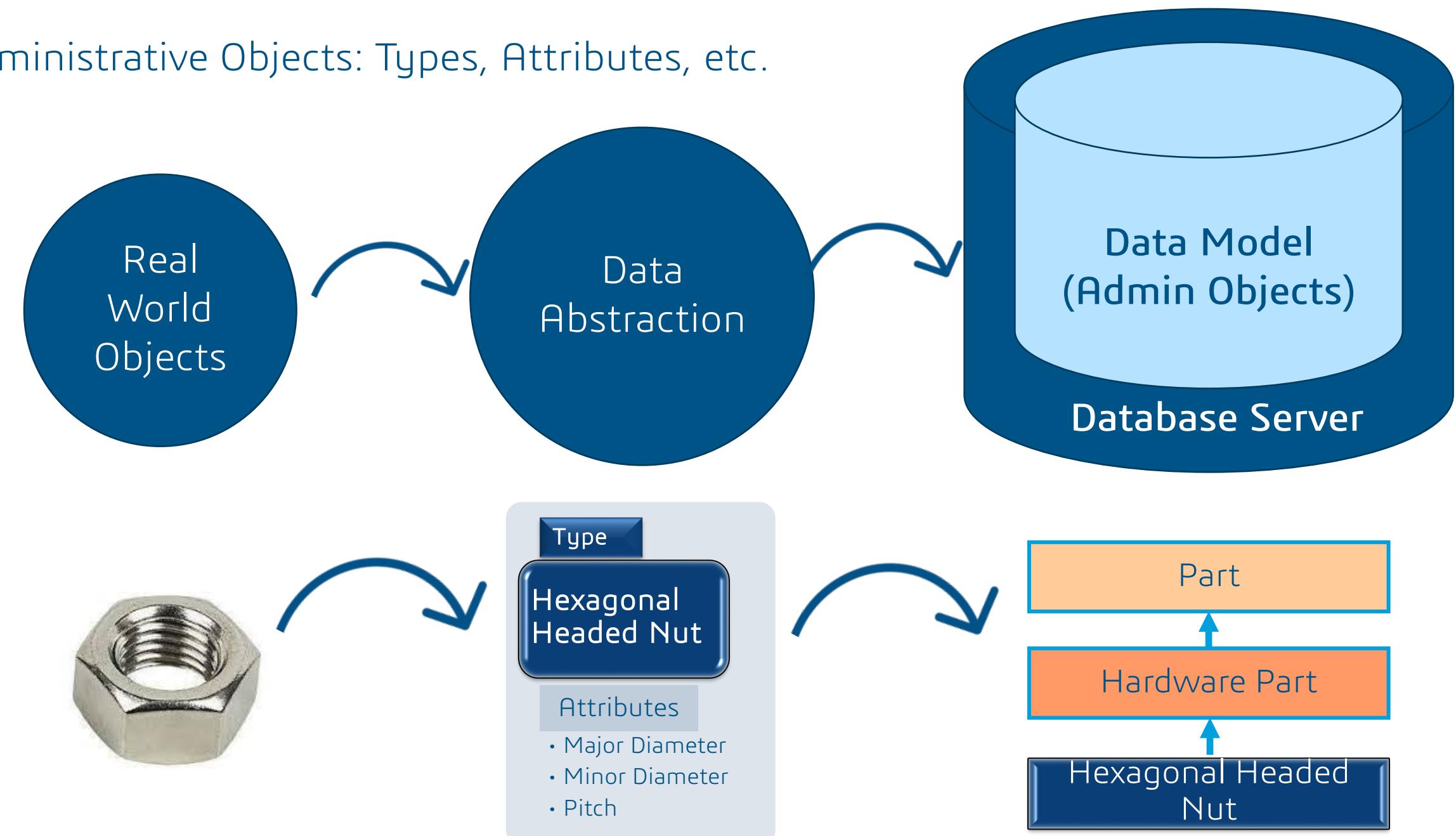
Introduction

In this chapter, we will have brief introduction to the concept of Administrative and Business Objects.

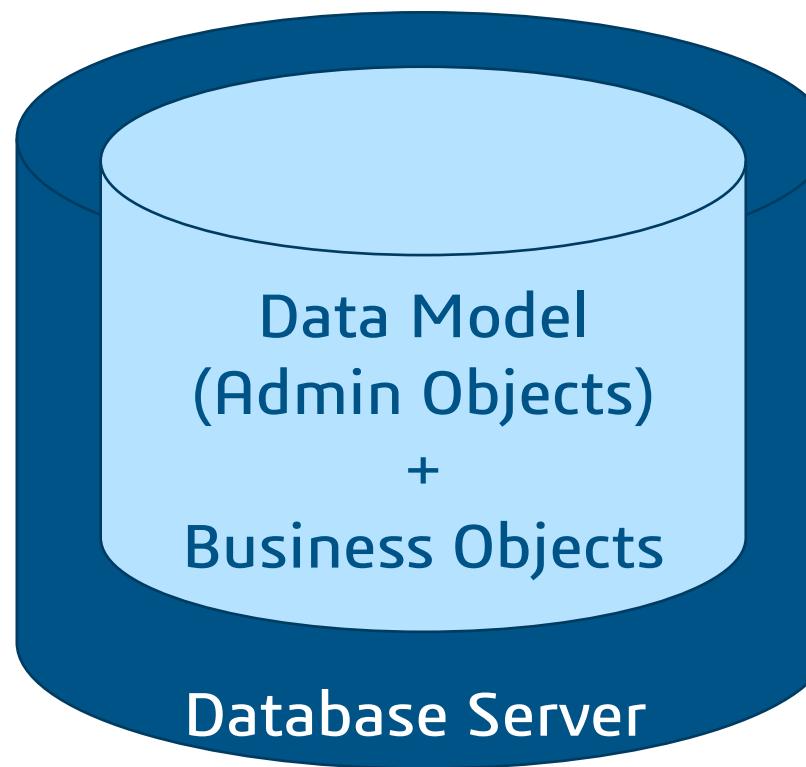
Administrative Objects



- Administrative Objects are responsible for the whole **3DEXPERIENCE** platform schema
- Administrative Objects define the **3DEXPERIENCE** platform Data Model explaining **how** information should be stored in database.
- Examples of Administrative Objects: Types, Attributes, etc.



- ❑ Business Objects are instances created from Administrative Objects
- ❑ Business Objects contain the information about customer data and business processes.
- ❑ Examples of Business Objects: Projects, Tasks, etc.
- ❑ Business Objects are displayed on the 3DEXPERIENCE platform Web as well as Native User Interfaces.



| ENOVIA - Product Structure Editor - 3 Products | | | | | | | |
|--|----------------------------|-------|-----------------------------|------------------|----------|----------------|-------------|
| | Title | Title | Name | Type | Revision | Maturity State | Description |
| <input type="checkbox"/> | + Physical Product00000112 | | prd-R1132101564589-00000112 | Roller Bearing | A | In Work | |
| <input type="checkbox"/> | + Physical Product00000113 | | prd-R1132101564589-00000113 | Roller Bearing | A | In Work | |
| <input type="checkbox"/> | + Physical Product00000116 | | prd-R1132101564589-00000116 | Physical Product | A | In Work | |

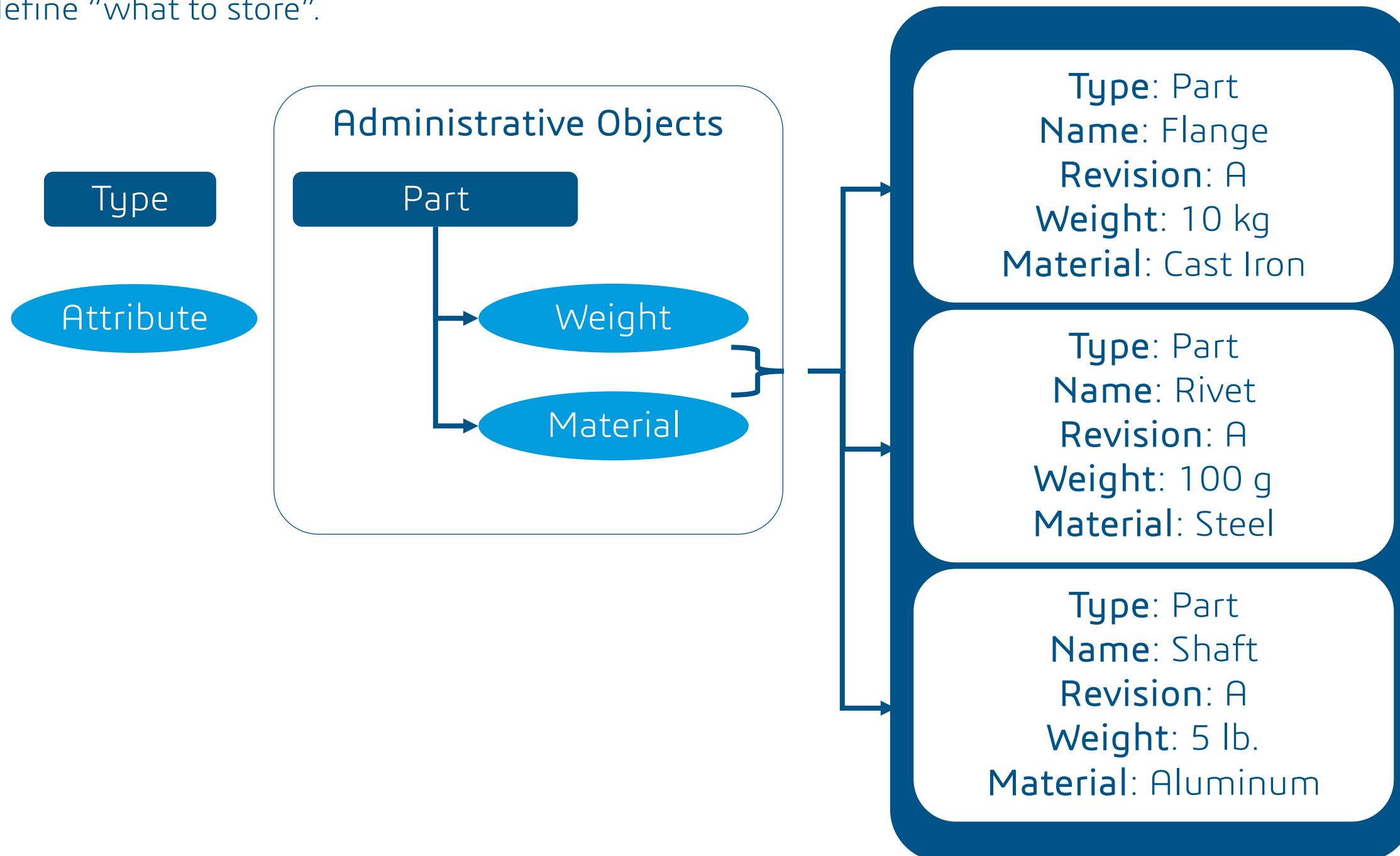
Difference between Administrative and Business Objects



□ Business Objects are instances of *Administrative Objects*.

□ This means that:

- Administrative Objects define “how to store information”
- Business Objects define “what to store”.





- There are various tools available for working with Administrative Objects and Business Objects.
- Please note that access to these mentioned tools are not provided to general end-users. Only users with special privileges are allowed to use these tools.
 - End users are intended to use only the front-end application (Native Applications or Web Applications)

| Tool for Object creation | Administrative Objects | Business Objects |
|--|------------------------|------------------|
| Collaborative Spaces Control Center | Yes | No |
| Data Model Customization (for now – only Types, Extensions and Attributes) | Yes | No |
| Enterprise IP Schema Specialization (for now – only Types, Extensions and Attributes) | Yes | No |

System Objects



Introduction

In this chapter, we will know about System Objects.



- The 3DEXPERIENCE platform uses different mechanisms to store the data
- To define those mechanisms, *System Administrators* will work with define the following kinds of Administrative Objects (*On Premise* only):

| System Object | Description |
|---------------|---|
| Vaults | A storage location for 3DEXPERIENCE platform Administrative and Business Objects |
| Stores | A storage location for files that are checked into Business Object (Physical Storage) |
| Locations | An alternate host and path for a Captured Store |
| Sites | A set of Locations |
| Indices | A set of Attributes and optional Basic properties that are stored in a single table to improve query and expand performance |

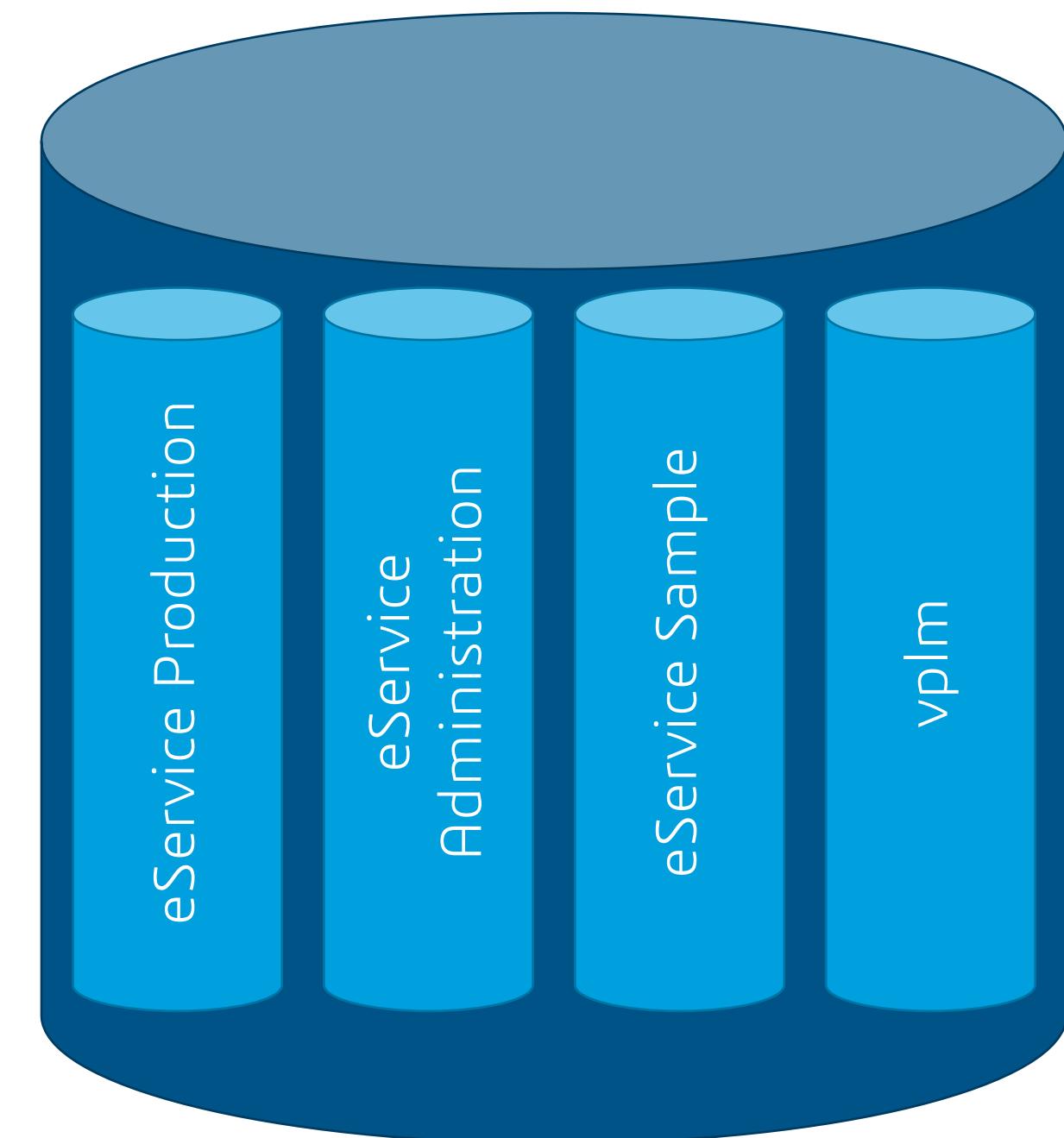
Comparison: Store versus Vault



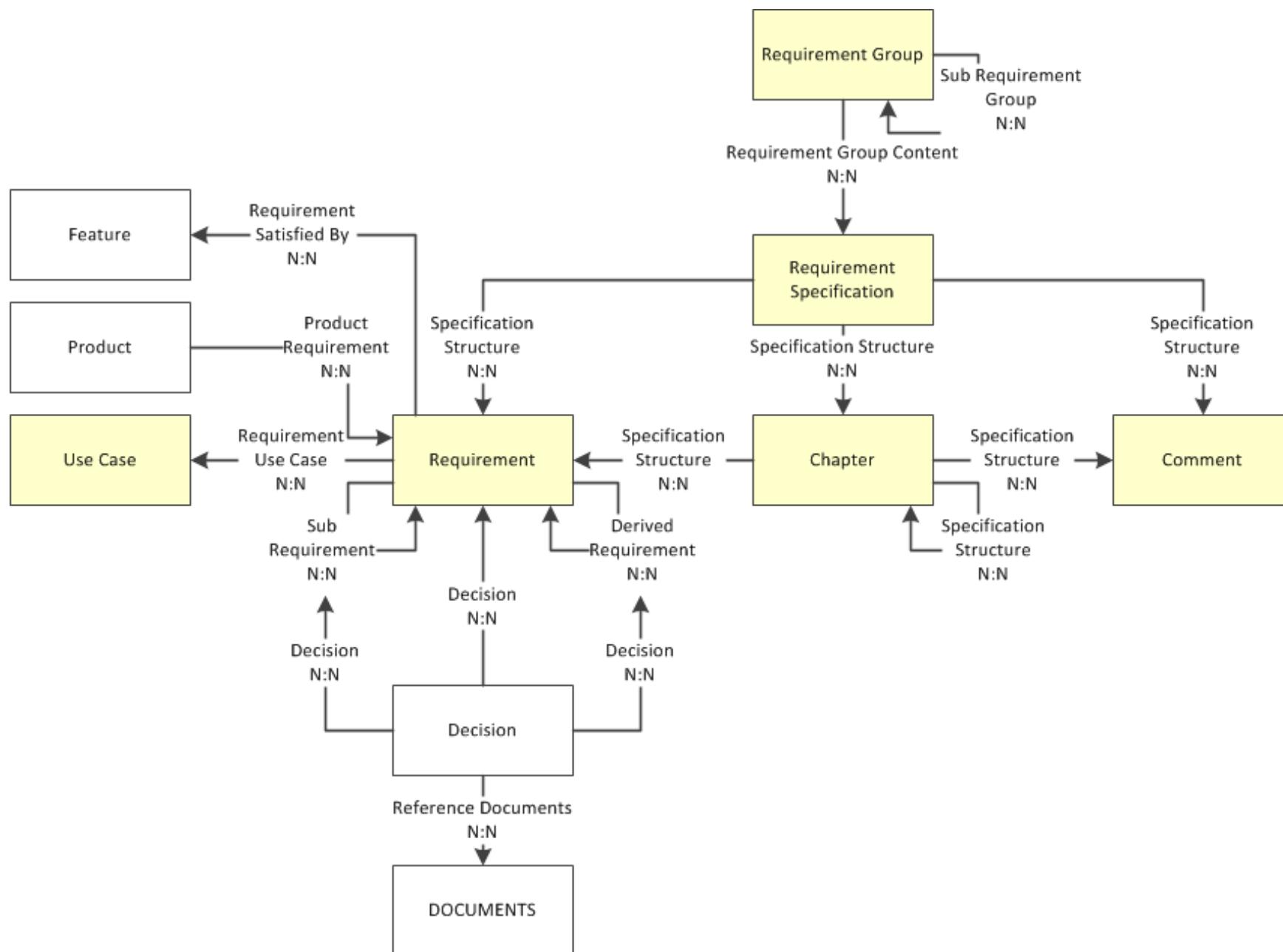
- ❑ A Store is the physical storage location of files.
- ❑ Store contains the files that are checked-in to Business Objects.



- ❑ A Vault is the logical organization of Business Objects and Administrative Objects.
- ❑ Vault contains metadata (information about objects).



Introduction to Data Model

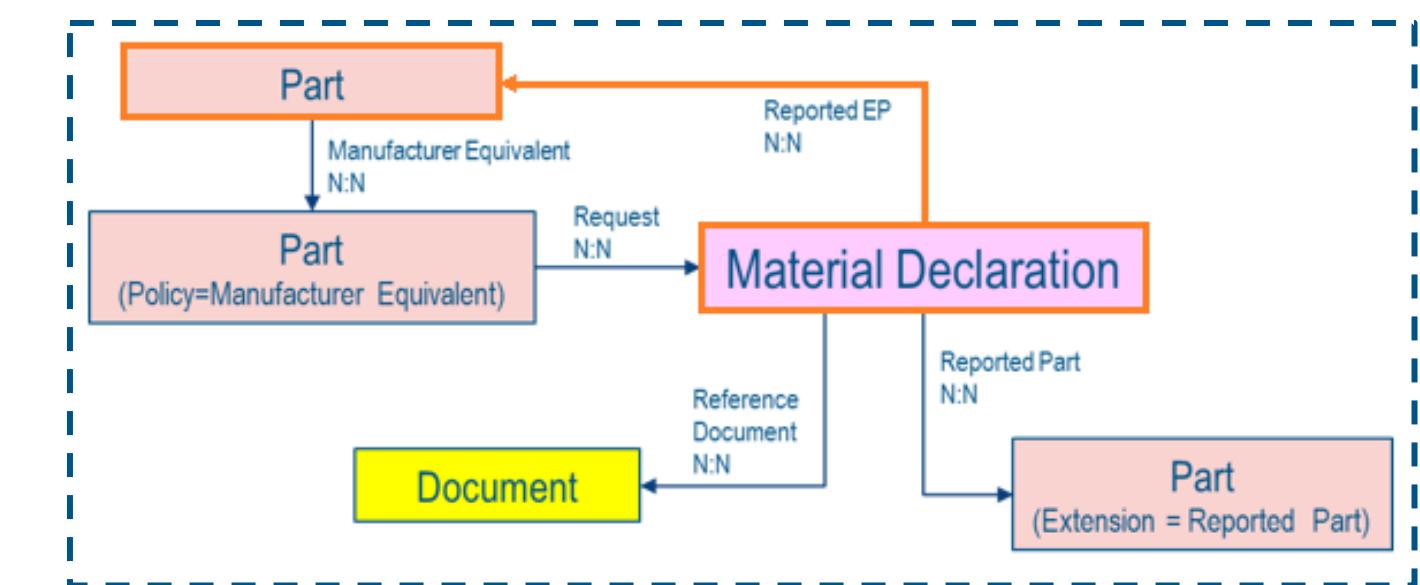


Introduction

In this chapter, we will discuss about Data Model



- Data Model defines how objects are related to each other and how they interact with each other.
- A Data Model illustrates object types used by an application and the relationships that connect them.
- Each Data Type can have any number of **attributes** defined for it.
- The graphic is an example which shows the data model for **Material Declaration**.
- Each rectangular box represents a type of business object.
- In this example,
 - Material Declaration and Part are *Types*, whose instances are connected with *Reported EP Relationship*.
 - Material Declaration Type is on *from* side and Part Type on *to* side.
 - For Reported EP relationship cardinality is *Many-to-Many*

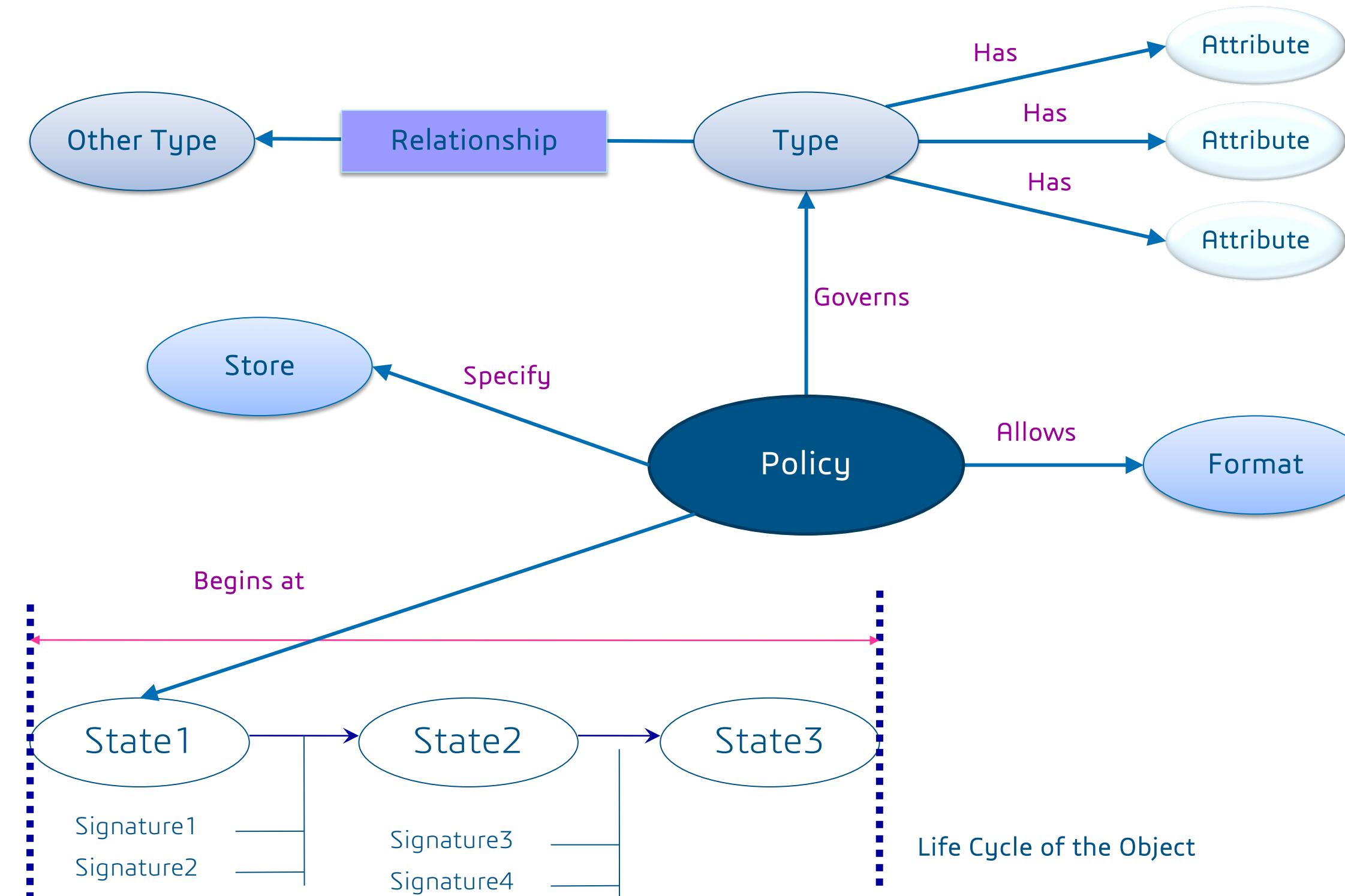


Entity Relationship (ER) Data Model (1/3)



- The Entity Relationship (ER) Data Model is the core model of 3DEXPERIENCE platform Web Apps (3DSpace)
- Entity Relationship (ER) Data Model is a graphical representation of entities and their relationships to each other.
 - ER Data Model is also often referred to as an ER diagram
- The ER Data Model relies on the notion of **Entities** and **Relationships** that can be defined between two **Entities**
- An entity is a piece of data-an object or concept about which data is stored.
- The two main objects in the Data Model are :
 - Types, to define the entities and these entities can have Attributes that define its properties.
 - Relationships, defined between Types.
- The Policy is a mandatory object which defines the behavior of a Type. It defines:
 - Revision sequence
 - Lifecycle
 - Access control
 - Signatures

Entity Relationship (ER) Data Model (2/3)





□ Type

- A **Type** defines *a kind of* business object and the collection of **Attributes** characterizes it.

□ Attribute

- An **Attribute** is any characteristic that can be assigned to an object or a Relationship.

□ Extension

- An **Extension** is a group of **Attributes** that can be added to Business Objects as well as connections to provide additional classification capabilities to objects.

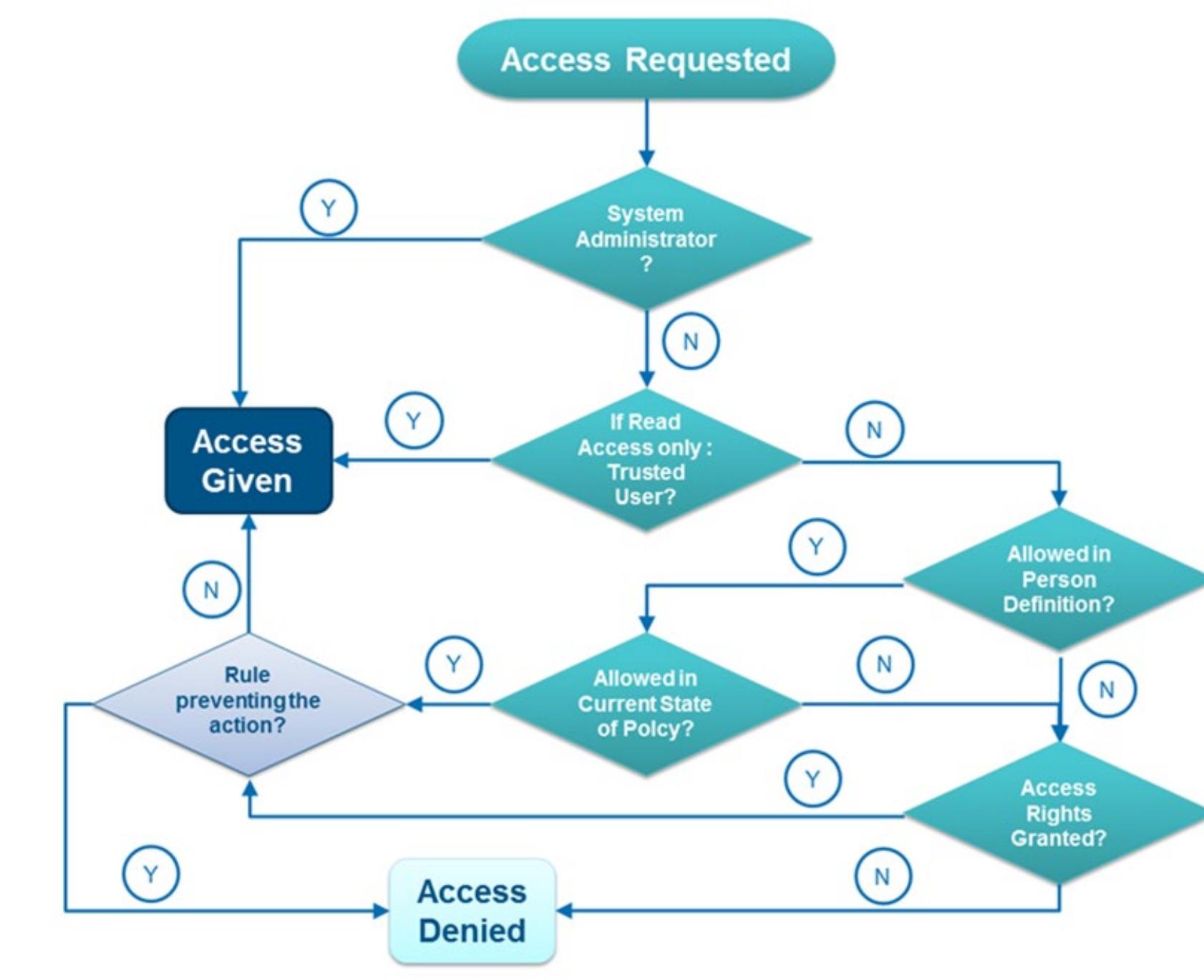
□ Relationship

- A **Relationship** is created to allow the connection between business objects and/or relationship instances.

□ Policy

- A **Policy** controls a business object.
- It specifies the rules that govern access, approvals, lifecycle, revisioning, etc.

- The Access Flow is defined as following, with first checks being applied on the User, and then on Policy.
- At last, Rules, which can restrict the access to specific Attributes, Program, and Relationships, are also applied

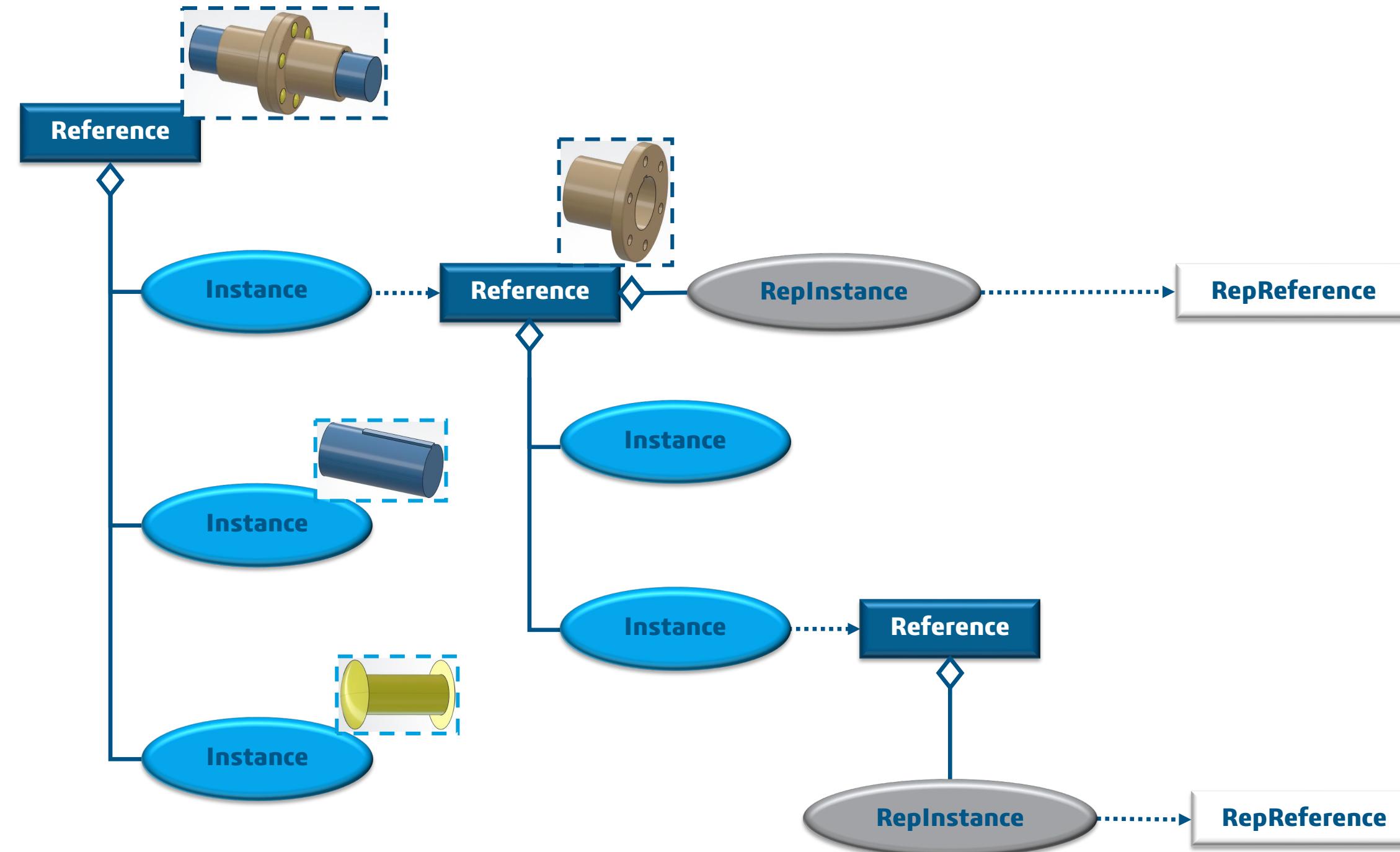


Instance Reference Port Connection (IRPC) Data Model (1/2)



- The **Instance Reference Port Connection (IRPC) Data Model** is the core model of **3DEXPERIENCE** platform Native Apps (CATIA, etc.)
- The IRPC Data Model relies on the notion of *Instance*, *Reference*, *Port* and *Connection*.
- IRPC Objects can be organized in structures, by creating parent-child relationships.
- A parent-child relationship represents the usage of an object in a structure, under another parent object.
- A **Reference** provides the definition of a structure object (a parent object in the model).
- An **Instance** is the usage of a reference in another reference.
- A **Connection** models semantic relationships between components i.e. is an object composed under a reference that can have multiple links.
- A **Port** is an object composed under a reference that can have one link.

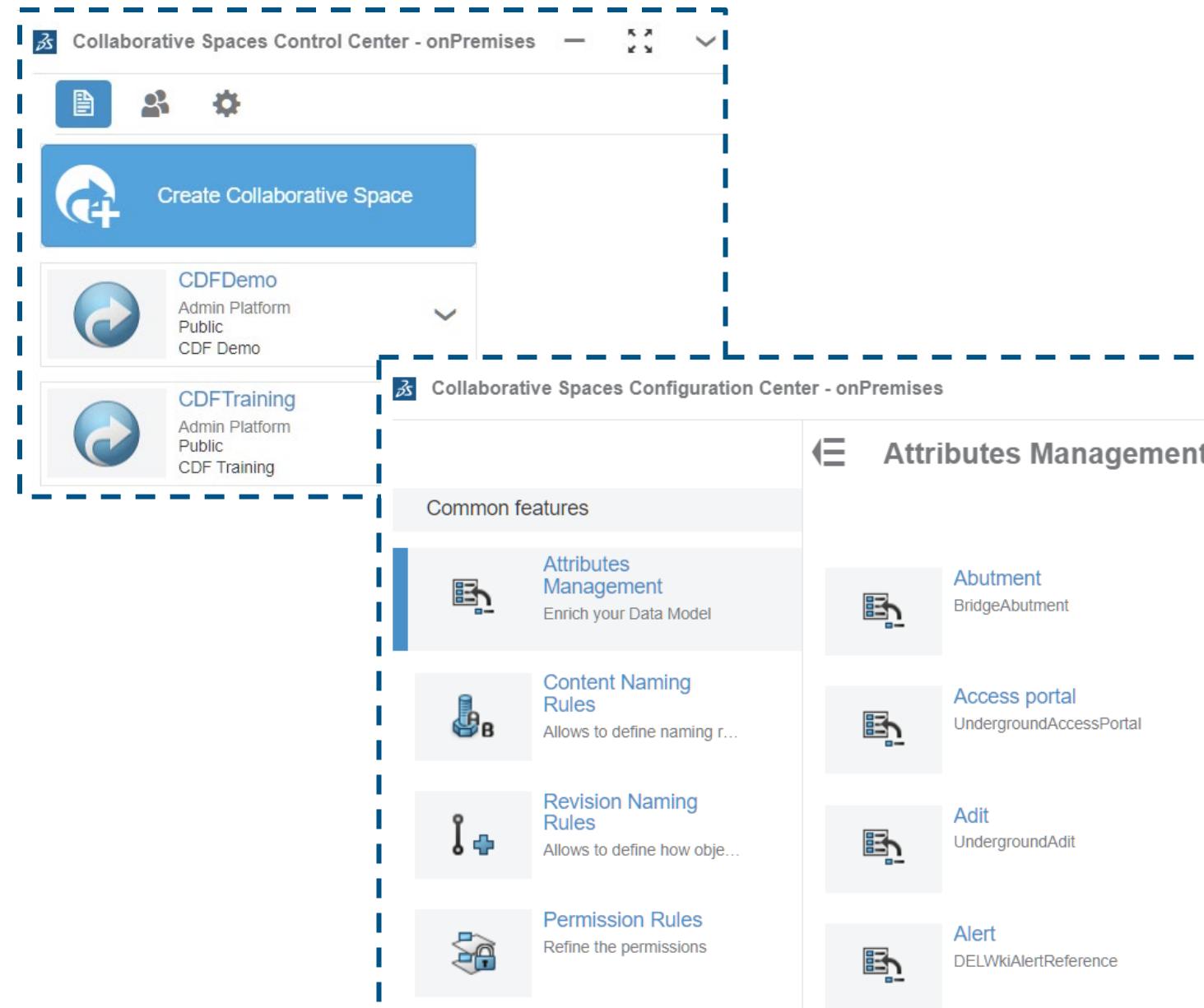
Instance Reference Port Connection (IRPC) Data Model (2/2)



In application we will be using the derived modeler types as indicated below:

| | | |
|--------------|---|----------------|
| Reference | → | VPMReference |
| Instance | → | VPMInstance |
| RepReference | → | 3DShape |
| RepInstance | → | VPMRepInstance |

Collaborative Spaces Control Center App



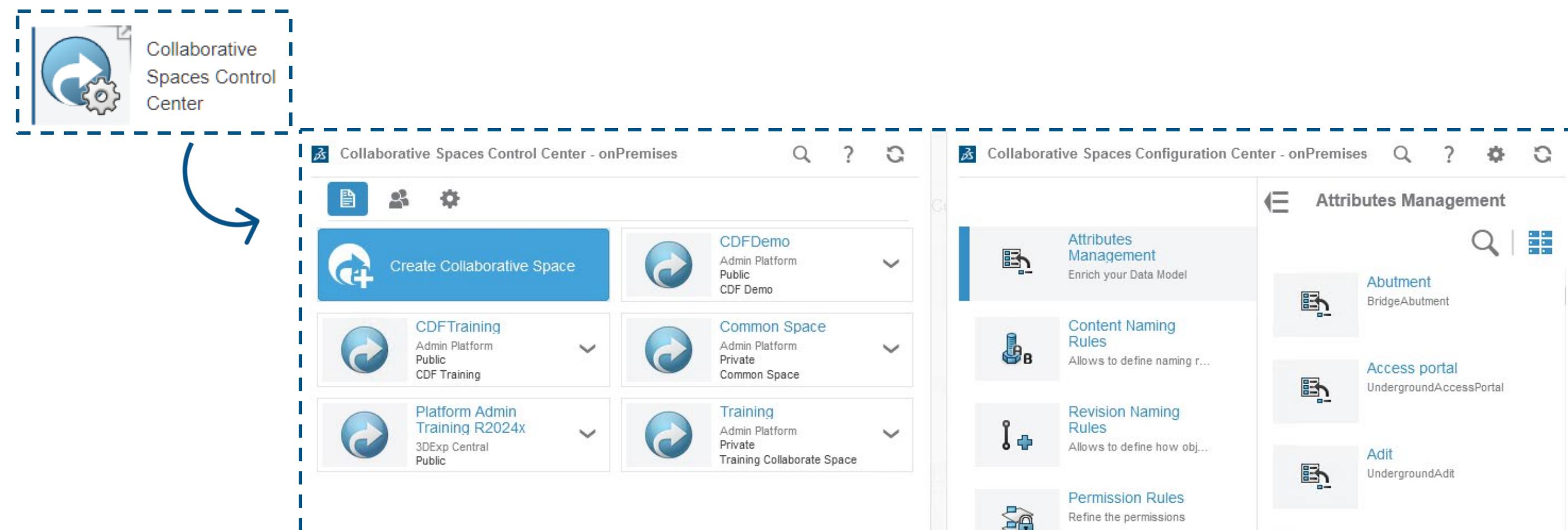
Introduction

In this chapter, we will get an overview about Collaborative Spaces Control Center App.

Collaborative Spaces Control Center App – Introduction



- The Collaborative Spaces Control Center app is the one-stop solution for all configuration functions. The App includes the following widgets:
 - **Collaborative Spaces Control Center:** Platform Managers can use the Collaborative Spaces Control Center widget to list Collaborative Spaces, create new ones, transfer ownership, define who can create Collaborative Spaces and configure settings.
 - **Collaborative Spaces Configuration Center:** Platform Managers can use the Collaborative Spaces Configuration Center to configure content behavior such as naming and revision rules, lifecycles, access rules, automation, and attributes



- For detailed information refer *User Assistance* at this [link](#).

Collaborative Spaces Control Center



The *Collaborative Spaces Control Center* widget has the following tabs:

- **Manage Collaborative Spaces:** Create new Collaborative Spaces or list existing ones, transfer ownership or edit visibility of collaborative spaces and manage roles.
- **Manage Permissions:** Defining who can create Collaborative Spaces, view the credentials assigned to a person
- **Manage Settings:** Configure generic settings for Collaborative Spaces Settings, as mentioned below
 - Access to *Common Space* collaborative space
 - Privilege of all new users to create their own collaborative spaces
 - Select the roles which would be allowed to be assigned to members of Collaborative Space
 - Access of Owner of Collaborative Space to change visibility of the same

The screenshot shows the 'Collaborative Spaces Control Center - onPremises' interface. At the top, there are three tabs: 'Manage Collaborative Spaces' (selected), 'Manage Permissions', and 'Manage Settings'. Below the tabs, there are several cards:

- Create Collaborative Space**: A blue card with a circular icon containing a plus sign and a 'C'.
- Common Space**: A card showing 'Admin Platform' as the platform, 'Private' as the visibility, and 'Common Space' as the name.
- Coexistence And Migration**: A card showing 'Admin Platform' and 'Public' as visibility.
- Platform Admin Training R2023x**: A card showing '3DEXP CENTRAL' and 'Public' as visibility.

The screenshot shows the 'Manage Settings' tab in the 'Collaborative Spaces Control Center - onPremises' interface. It includes sections for 'New users deployment' (with toggles for giving access to common space, enabling creation, and creating personal spaces for xApps users), 'Responsibility availability' (with toggles for Reader, Public reader, Restricted responsibilities, and Associate Leader and Owner), and 'Manage visibility' (with a toggle for owner access responsibility).

The screenshot shows the 'Manage Permissions' tab in the 'Collaborative Spaces Control Center - onPremises' interface. It displays a list of users and their creation permissions:

| User | Creation Status | Notes |
|---------------------|--------------------------|-------------------------------------|
| 3C | Not allowed for creation | 3dexpcentral@smtp.mytraining.com |
| AP | Allowed for creation | admin_platform@smtp.mytraining.co m |
| Administration User | Not allowed for creation | Unknown |
| EP | Not allowed for creation | erik_peters@3dp.com |
| JP | Not allowed for creation | john.peter@3dp.com |
| Joseph Bond | Not allowed for creation | Joseph@3dp.com |
| Test Everything | Not allowed for creation | warren@Test2 Customer.com |

A callout box highlights the 'Allow Creation' button for the 'AP' user, and another callout box shows the 'Display credentials' button.

Collaborative Spaces Configuration Center



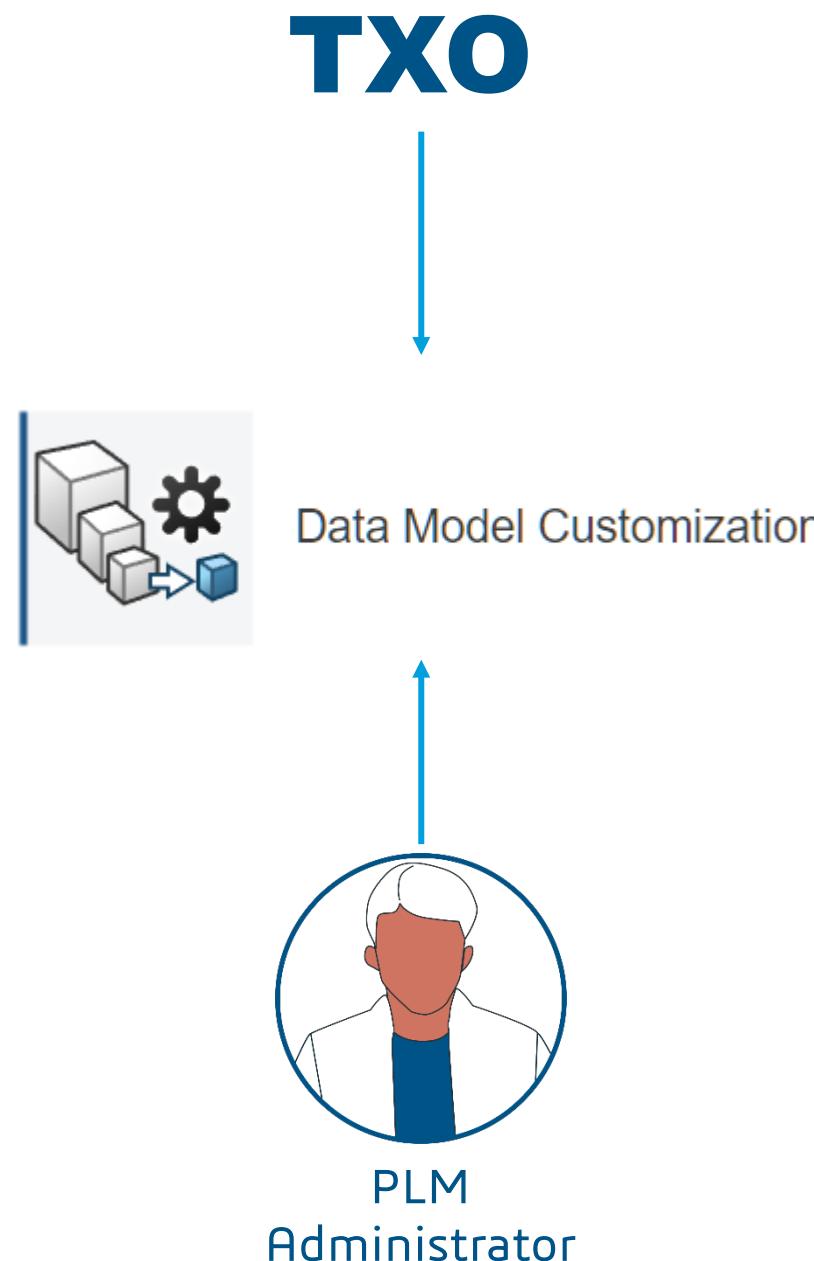
Collaborative Spaces Configuration Center widget has the below configuration options:

- Attribute Management:** Extending OOTB Types with new Attributes.
- Content Naming Rules:** Define AutoName series for OOTB Types.
- Revision Naming Rules:** Select applicable Revision Sequence for certain OOTB Types.
- Permission Rules:** Select read and write rules that determine who can open which content based on users permission levels and credentials.
- Maturity Graph:** Configure certain lifecycle Policies
- Configuration Deployment:** Ability to Import and Export Configuration, Reload Cache, Update index, Download CATNIs, etc.
- Other available modules of the Collaborative Spaces Configuration Center widget are governed by web application installed.

The screenshot shows the 'Attributes Management' section of the 'Attributes Management' module. It includes a search bar and a back arrow. On the left, there's a sidebar with 'Common features' and 'Application-specific features' sections. The main area lists several items with icons and descriptions:

| Icon | Name | Description |
|-------------------------------|--------------------------|------------------------------|
| | Attributes Management | Enrich your Data Model |
| | Content Naming Rules | Allows to define naming r... |
| | Revision Naming Rules | Allows to define how obje... |
| | Permission Rules | Refine the permissions |
| | Maturity Graph | Configure Maturity Graphs |
| | Configuration Deployment | Import and Export Config... |
| Application-specific features | | All Manufacturing |

Data Model Customization App



Introduction

In this chapter, we will know about Data Model Customization App used for customization of Data Model On Premises.

Data Model Customization (1/2)



- ❑ Data Model Customization app (or TXO) is an *Out Of The Box* (OOTB) set of tools for customizing the 3DEXPERIENCE platform Data Model.

- ❑ The features of Data Model Customization application are as below:
 - It is an unified and common framework for all PLM applications. It unifies the Data Model customization techniques for all 3DEXPERIENCE applications

 - It goes along with a new generation of Data Model customization consoles:
 - Simple and easy to use
 - Instantaneous testing and deployment

 - New capacities for Dassault Systèmes Data Models are offered to end-users, such as:
 - Depth in hierarchy
 - Discipline as Type

- ❑ The former type customization mechanism was:
 - Less restrictive
 - No declaration of which types can be specialized and extended.
 - Less protective
 - DS Types can be modified with the Business Modeler
 - This resulted in upgradeability issues.

Data Model Customization (2/2)



- ❑ TXO is now the recommended way to customize data models
 - It fully replaces Data Model Customizer (DMC) for Native Applications.
- ❑ TXO does not allow to create data model from scratch
 - It extends/specializes only data models provided by DS applications.
- ❑ TXO does not concern other areas of customization
 - Security, Web forms, etc.
- ❑ In order to work with TXO, you will need Administrator role
- ❑ The following table summarizes the capabilities of the tool:

| | Customization Capability | Who can Access Tools |
|---------------------------|--|----------------------|
| Data Model Specialization | <ul style="list-style-type: none">❑ Specialization Types, with their Attributes❑ Specialization Extensions, with their Attributes❑ Customer Extensions, with their Attributes❑ Unique Keys to define constrained Attributes | PLM Administrator |
| Data Model Administration | <ul style="list-style-type: none">❑ Deployment Extension, with their Attributes❑ Unique Keys to define constrained Attributes | |

- ❑ Security context **VPLMAadmin.Company Name.Default** is required to login as PLM Administrator

Installer, License and Role



- To access *Data Model Customization* tools from 3DSpace, *DataModelCustomizationFoundation* must be installed on top of 3DSpace.
- Data Model Customization tools are available through **Data Model Customization App** in the **Compass**
- Data Model Customization App** contains the below Commands:
 - Manage Packages
 - Specialize Data Model
 - Administrate Data Model
 - Manage NLS
 - Download CATNIs
 - Manage Business Rules
- User should be assigned **Customization Administrator** role (TXO) to use **Data Model Customization App**
- With the *Data Model Customization App*, packages can be exported and imported.

The screenshot shows the 'Customization Administrator' interface in the 3DEXPERIENCE platform. At the top, there are several icons and sections: 'Change Execution', 'Collaborative IP Management', 'Data Setup', and 'Know-how Apps Resources'. Below this, the main area is titled '3DEXPERIENCE | Data Model Customization'. On the left, a sidebar lists 'Data & Behavior Customization' options: 'Manage Packages', 'Specialize Data Model', 'Administrat Data Model', 'Manage NLS', 'Download CATNIs', and 'Manage Business Rules'. On the right, a table titled 'List of Specialization and Deploy' shows a list of packages with columns for 'Full Name' and other details. The package 'ClassificationDefaultPack' is highlighted in blue.

Values of the Data Model Customization Application



- Data Model Customization app has a Native interpretation from 3DEXPERIENCE platform
- Data Model Customization is loosely coupled with other personalization area (behavior, security & user interface)
- Other advantages that can be directly observed are:
 - More efficient
 - Native access to specialized Types, Attributes and Extension Attributes in configured forms
 - No need to modify Forms, no need to define access rights on Attributes
 - Safer & cost reduction
 - No modification of DS Types, suppresses the risk and cost of upgradability issues
 - More stable
 - Control of what is natively supported by DS applications and guaranteed by DS
 - Greater opportunity
 - Independence between *Data Model Administration* and *Data Model Specializations* processes.
 - Provides more opportunity to PLM Administrator for integrating Knowledge.

Glossary – Common Terminologies for Typing



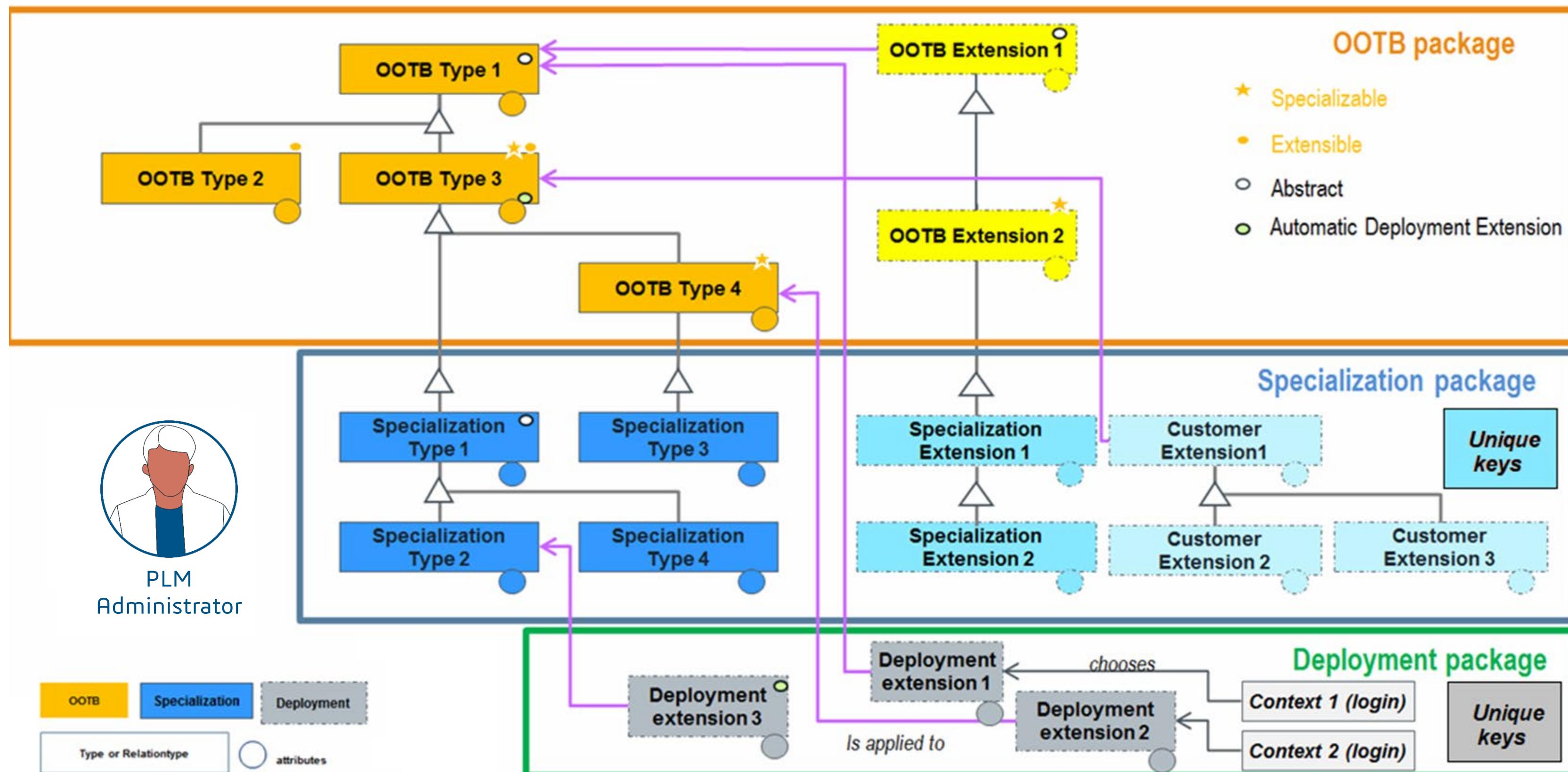
- **Package:** It is a group of Types and Extension Types, grouped together by domain
 - **DS Package:** Contains DS delivered Types and Extensions (**read-only**).
 - **Specialization package:** Contains Specialization Types, Specialization Extensions, Customer Extensions and Unique Keys
 - **Deployment package:** Contains Deployment Extension and Unique Keys

- **Type:** Refers to the category of objects
 - **DS Type:** Type of objects, as delivered by **3DEXPERIENCE** applications (Example: VPMReference)
 - **Specialization Type:** Sub type that inherits directly or indirectly from a DS Type

- **Extension:** Refers to types of Extensions that can be added on an object
 - **DS Extension:** Type of Extensions, as delivered by **3DEXPERIENCE** applications
 - **Specialization Extension:** Sub Extension that inherits directly or indirectly from a DS Extension Type
 - **Customer Extension:** Extension that does not inherit from a DS Extension Type
 - **Deployment Extension:** Extension that does not inherit from a DS Extension Type

- **Unique key:** Set of Attributes that are used as a key for uniquely identifying objects

Type, Extension, Unique Key and Packages



Overview on Package



□ Introduction of Package

BearingPackage

Specialization Package

Types Extensions Unique Keys

Types List

| Name | Ownership | # of Extensions In Other Packages | Abstract - Deprecated | Deployed | Attributes |
|---------------|-----------|-----------------------------------|-----------------------|----------|------------|
| VPMReference | DS | 0 | - | Yes | 40 |
| RollerBearing | DS | 2 | No - No | Yes | 4 |
| VPMD | | | | | 22 |

Specialization Type

Attributes Scopes

Attributes List | of the Selected Object

| Name | Deployed | Type | Default Value | Keys (Packag | |
|---------------|----------|------|---------------|--------------|---|
| InnerDiameter | Simple | No | Yes | Integer | 0 |

Specialization Type Attribute

InnerDiameter

Defining Specialization Package



- Before creating Specialization Type, Specialization Extension or Customer Extension, a Specialization Package must be created:
 - From the Specialize Data Model menu, click on New option,
 - Parent Package: Select the DS Package to be specialized
 - Name: Define the name of the Specialized Package
 - Prefix: Define a prefix to avoid name conflict

The screenshot shows the SAP Fiori interface with a sidebar on the left containing the following items:

- ▼ Data & Behavior Customization
- Manage Packages
- Specialize Data Model** (highlighted with a blue box)
- Administrate Data Model
- Manage NLS

Below the sidebar is a main area titled "List of Specialization Packages". It features a toolbar with icons for creating, deleting, and managing packages. A table lists two packages:

| Name | Parent Pack |
|---------------------|-------------|
| CustomerRequirement | Requirement |

The screenshot shows the "Create New Package" dialog box with the following fields:

- Parent Package: PRODUCTCFG (with a browse icon)
- Name: BearingPackage (highlighted with a blue box)
- Prefix: Enter New Prefix: CDF (highlighted with a blue box)
- Comment: (empty text area)

At the bottom right are "Done" and "Cancel" buttons.

Defining Specialization Types



- Once the Specialization Package is created, it is possible to add a new Specialization Type:
 - Select the DS Type, and click on New
 - Name: Define the name of the Specialized Type

The screenshot shows the 'Types' tab in a software application. The 'Types List' contains two entries: 'VPMReference' and 'VPMPort'. The 'VPMReference' entry is highlighted with a light blue background and has a checked checkbox next to its name. The 'Name' column has a header 'Name' and a sub-section for 'VPMReference'. The 'Ownership' column shows two icons, each with a 'DS' symbol. The interface includes a toolbar with icons for creating, deleting, and managing types, as well as filters and search functions.

The screenshot shows the 'Create New Sub Type' dialog box. It includes fields for 'Parent Type' (set to 'VPMReference'), 'Name' (set to 'RollerBearing'), 'Abstract' (set to 'false'), 'Instance' (empty), 'Icon Name' (empty), and 'Comment' (empty). The 'Done' and 'Cancel' buttons are located at the bottom right of the dialog.

Defining Specialization Extension



- Specialization Extension Type is created from the Extension tab of the Package:
 - Select the DS Extension to be specialized, and click on New option

| Name | Ownership | Scopes | Abstract - Deprecated |
|-------------------|-----------|-------------|-----------------------|
| DELRmiPathway | DS | (Inherited) | - |
| ControlDevice | DS | (Inherited) | - |
| Conveyor | DS | (Inherited) | - |
| IndustrialMachine | DS | (Inherited) | - |
| Inspect | DS | (Inherited) | - |
| LogicController | DS | (Inherited) | - |

Create New Customization...

Fields in red italics are required

Parent Extension
IndustrialMachine

Name

Scopes
VPMReference

Icon Name

Comment

Done **Cancel**

Defining Customer Extension



- Customer Extension is created from the **Extension** tab of the Package. It is not required to select a DS Extension:
 - Click on New

Bearing Package

TypesExtensionsUnique

List of Extensions

Create New Customization Ex...

Fields in red italics are required

Parent Extension

Name
ApplicationCategory ApplicationCategory

Abstract
false ▼

Scopes
RollerBearing ... Clear

Comment

Done Cancel

Defining Attributes



- Once the **Specialization Type**, **Specialization Extension** and **Customer Extension** are created, it is possible to create new attributes for them:
 - Select the customized object (Type or Extensions)
 - From the **Attributes** tab (bottom of the window), click on **New Simple**

BearingPackage

Types **Extensions** **Unique Keys**

List of Extensions

| Name | Ownership | Scopes | Abstract - Deprecated | Deployed |
|---------------------|-----------------|---------------|-----------------------|----------|
| ApplicationCategory | Customer Ext... | RollerBearing | No - No | Yes |
| ProductCategory | Customer Ext... | | No - No | No |

Attributes **Scopes**

Attributes List | of the Selected Object

| Name | Nature | Deprecated | Deployed | Type |
|------|--------|------------|----------|------------------|
| | | | | No Objects Found |

Create New Attribute

Fields in red *italics* are required

Name

6W Predicate [Clear](#)

Type

String

Integer

Boolean

Date

Real

Choose magnitude: [...](#)

Preferred Unit

Multiline false

Multi Valuated false

Authorized values [Clear](#)

Done **Cancel**

Defining Deployment Package



- ❑ Before creating Deployment Extension, a Deployment Package must be created:

- From the Administrate Data Model menu, click on New option
 - Parent Package: Select the DS Package to be specialized
 - Name: Define the name of the Deployment Package
 - Prefix: Define a prefix to avoid name conflict

The screenshot shows a software interface for managing deployment packages. On the left, there's a sidebar with options like 'Data & Behavior Customization', 'Manage Packages', 'Specialize Data Model', 'Administrate Data Model' (which is highlighted), 'Manage NLS', 'Download CATNIs', and 'Manage Business Rules'. The main area is titled 'List of Deployment Packages' and contains a table with columns for Name, OtbERConfiguration, OtbIRPCConfiguration, and RequirementDeployment. There are icons for creating, deleting, and managing packages. The 'Specialize Data Model' option is also highlighted in the sidebar.

The screenshot shows a 'Create New Package' dialog box. It includes fields for 'Name' (set to 'BearingMaterial'), 'Prefix' (set to 'CDF'), and 'Comment'. At the bottom are 'Done' and 'Cancel' buttons. A note at the top says 'Fields in red italics are required'.

Defining Deployment Extension



- Deployment Extension includes a set of Attributes that will be systematically added to objects of a given Type (depending on the project).
 - One *Deployment Extension* could be associated with multiple Types.

- Once the Deployment Package is created, it is possible to add a new Deployment Extension:
 - Click on New option

The screenshot shows the 'Data & Behavior Customization' sidebar with options like 'Manage Packages', 'Specialize Data Model', 'Administrate Data Model', 'Manage NLS', 'Download CATNIs', and 'Manage Business Rules'. The main area is titled 'BearingMaterial' with 'Categories' and 'Actions' dropdowns. Below is a 'Deployment Extensions List' section with icons for creating, deleting, and managing extensions. A blue box highlights the 'Create' icon.

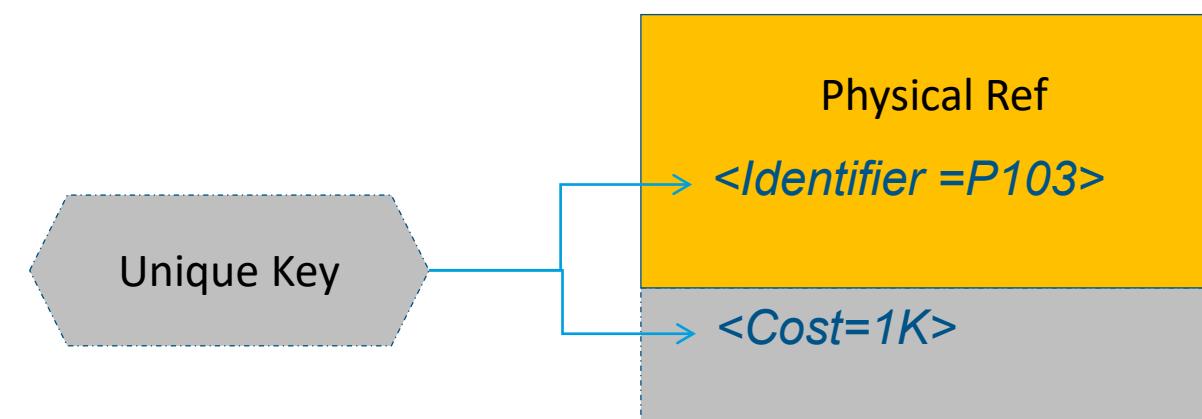
The dialog box is titled 'Create New Admin Extension' and contains the following fields:

- Name**: MaterialCatalogue
- Types**: RollerBearing (with a 'Clear' button)
- Automatic addition**: false
- Comment**: (empty text area)

At the bottom are 'Done' and 'Cancel' buttons.



- ❑ Unique Keys can be defined to ensure that two objects won't be saved with the same values for a set of chosen Attributes
 - For example: Type name + PLM External ID + Version number
 - This is used for « identifying » objects with a name that can be used in other applications (excel, SRM, etc.)
- ❑ Attributes can be taken from the Type and/or the Extension, if any. A Unique Key can also constrain some basic Attributes such as revision, owner, originated.
- ❑ By default, no Unique Key is necessary. Unique Keys should be defined only if necessary, since they cause a small performance impact for every exchange, in order to maintain uniqueness of metadata.
 - If a custom Type is constrained by Unique Key(s), then the Type cannot be deleted unless the Unique Key is disabled.
 - Before performing Unified Typing Migration, custom Unique Key created by users should be deactivated.
 - A new Unique Key cannot be activated if instances of the constrained Types and Attributes already exist with same attributes values.



Enterprise IP Schema Specialization App



Enterprise IP Specialization Manager



Types

Types specializing out of the box types



Attributes Groups

Attributes Groups applicable to types



Extensions

Extensions applicable to types



Unique keys

Create your own unique key



Tools

More tools to operate specialization

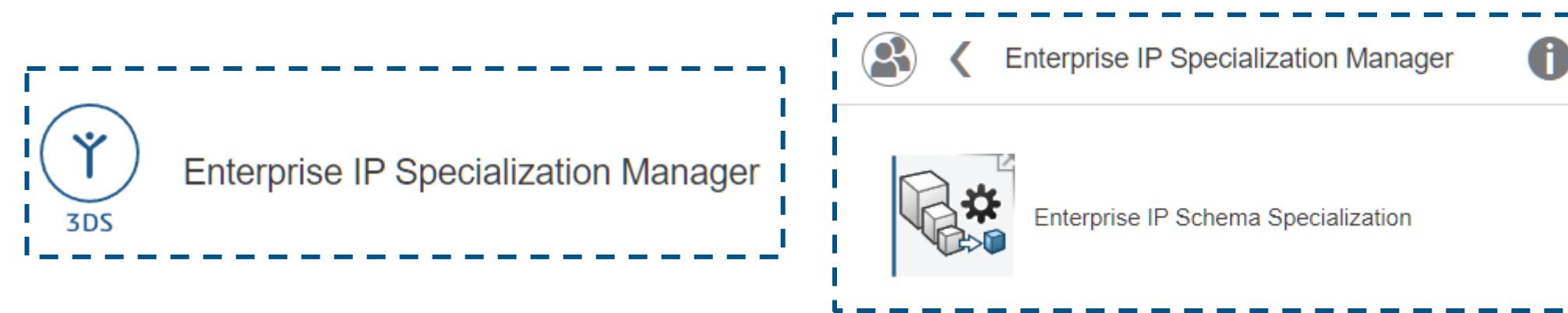
Introduction

In this chapter, we will discuss about Enterprise IP Schema Specialization App used for configuration of Data Model On Cloud.

Enterprise IP Schema Specialization App – Introduction



- ❑ To customize Data Model for **3DEXPERIENCE** platform that is deployed on **Private or Dedicated Cloud**, the **Enterprise IP Schema Specialization (DBS)** Dashboard app is used.
- ❑ It enables *Platform Manager* to customize the data model according to their business needs.
- ❑ User must be assigned the **Enterprise IP Specialization Manager** role in order to access **Enterprise IP Schema Specialization** app
 - *Enterprise IP Specialization Manager* role is assigned by associating DBS license with the user.



- ❑ For more details you can refer to User Assistance [link](#)



- ❑ *Enterprise IP Schema Specialization* is not available *on premises nor on public clouds*.
- ❑ Data Model Configuration **can** be performed in **Development** environment only.
- ❑ Data Model Configuration **cannot** be performed in **Production** environment.
 - Only import of custom data model is allowed in **Production** environment.



- ❑ Type
 - Type of objects.
- ❑ OOTB DS Type
 - Type delivered by DS.
- ❑ Specialization Type
 - Type created by a customer by deriving a specializable OOTB DS type.
- ❑ Attributes Group
 - Attributes group enable the administrator to add attributes to existing DS types.
- ❑ Extension
 - An extension defines an additional attribute for a particular object.
- ❑ Unique Keys
 - Set of Attributes that are used as a key for uniquely identifying objects
- ❑ Internal Name
 - Name given to a specialization element at creation time.
- ❑ External Name
 - Name of a specialization element that appears on the end-user apps according to the end-user app's language.

Enterprise IP Schema Specialization App – Capabilities



The Enterprise IP Schema Specialization tool provides the following key capabilities:

- Customize the data model by performing the following:
 - Create *Specialization Types* by specializing OOTB DS types.
 - Create *Attributes Groups*.
 - Create *Extensions*.
 - Create Unique Keys
- Import Business Rules created and exported with Data Setup.
- Work in a development environment to safely create specializations before importing them into a production environment.



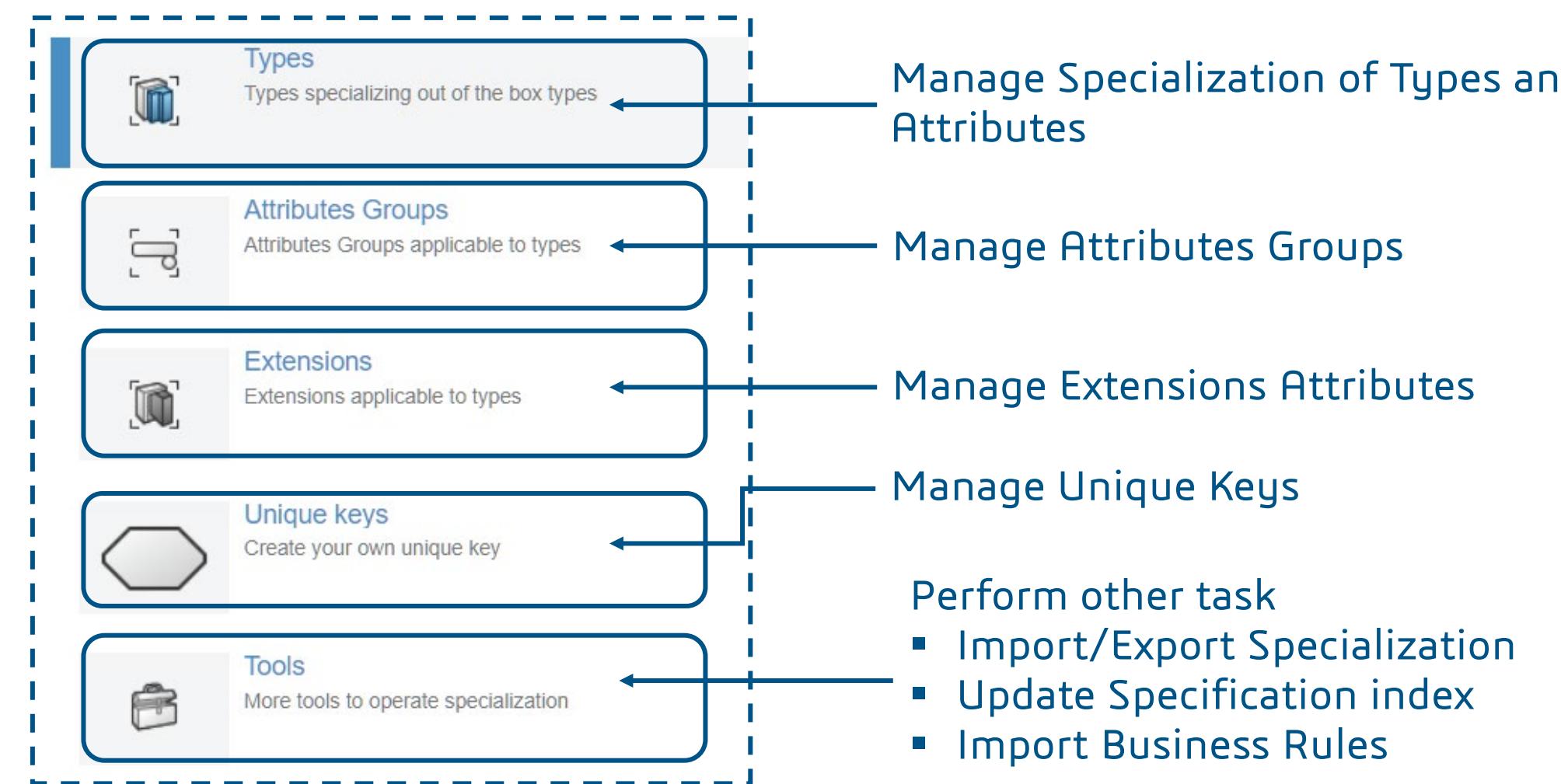
- Platform specialization capabilities are limited by set of specialization features ("Specialization Set")
- One Specialization Set allows
 - 10 Specialization types
 - 20 Extensions
 - 10 Attributes Groups
 - 40 Specialized attributes per object
- In order to manage additional Specialization Sets, Customer must order the required number of *Enterprise IP Specialization Capacity*.
- At any given time, Customer cannot have, for a given Platform, more than four (4) *Enterprise IP Specialization Capacity*

Enterprise IP Schema Specialization App – Usage



Typical usage of Enterprise IP Schema Specialization App is the following:

1. Using **Enterprise IP Schema Specialization** widget, *Enterprise IP Specialization Manager* develops and tests the specializations in a **development** environment.
2. *Enterprise IP Specialization Manager* will export the Specializations to a file
3. *Enterprise IP Specialization Manager* then imports the file in the Production environments using **Enterprise IP Schema Specialization** widget and the Specialization is made available.



Data Layer Configuration Overview



Here are the key takeaways from this lesson:

- ❑ Data Model defines how objects are related to each other.
 - ER Data Model governs **3DEXPERIENCE** platform Web Apps
 - IRPC Data Model governs **3DEXPERIENCE** platform Native Apps
- ❑ *Platform Manager* can use **Collaborative Spaces Configuration Center** app to configure OOTB functionalities.
- ❑ **Data Model Customization** app (TXO) is used to customize the Data Model on premises.
- ❑ **Enterprise IP Schema Specialization** app (DBS) is used to customize the Data Model on private or dedicated Cloud.

Presentation Layer Configuration Overview



—

After completing this lesson you will:

- Know about Widgets and its configuration
- Know about Internationalization



Widget Configuration



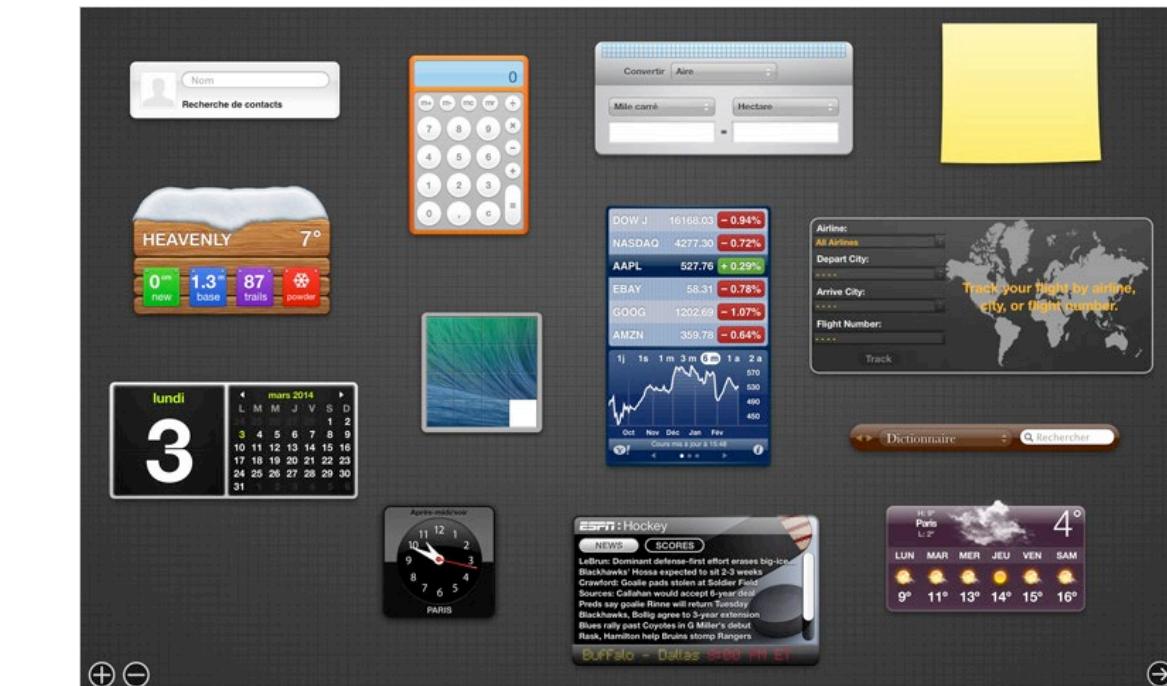
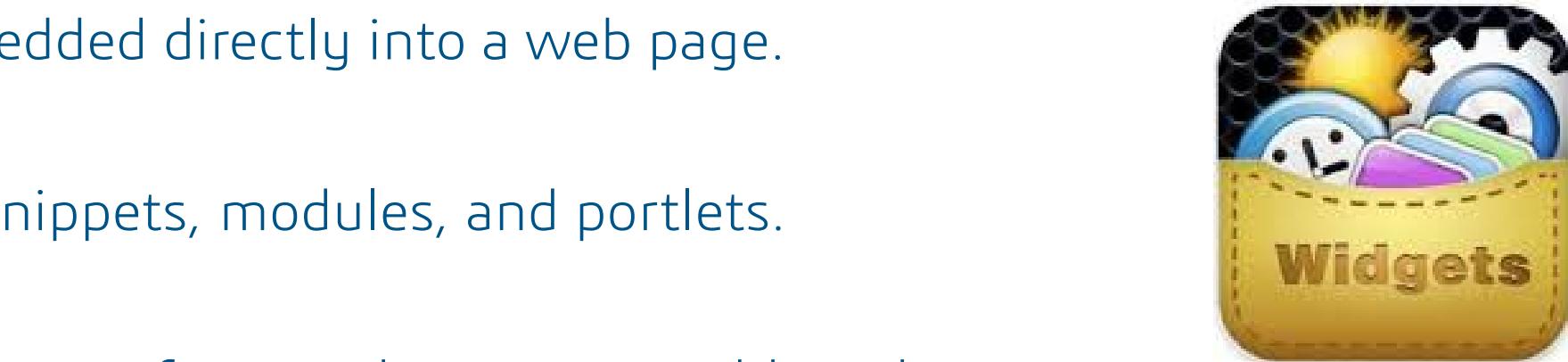
Introduction

In this chapter, we will get introduced to Widgets, and learn about the fundamental concepts of a Widget.

Widget – Definition



- A **Widget** is a small helpful software program embedded directly into a web page.
- The widgets are also referred to as web-gadgets, snippets, modules, and portlets.
- A single Widget or multiple widgets in combination are often used to create Dashboard pages.
- Widgets are lightweight HTML and script-based applications that provide the ability to derive and present information or functionality from a variety of sources, such as local applications and controls, or websites and services.
- The Widgets are designed to provide control to the user. In many cases, the page users have the authority to choose and place web-widgets where they want.
- Examples of widgets:
 - Daily weather reports
 - Clocks and countdown timers
 - Flight arrivals at the airport
 - Latest local news headlines
 - Latest international news, etc.





- A Widget is a specialized application based on standard web-technologies (HTML5, JavaScript, CSS)
- Content that are displayed in a Widget within 3DEXPERIENCE Dashboard may be drawn from the following sources:
 - Internal Content: Drawn from sources like 3DSwym, 3DSpace, etc.
 - External Content: Drawn from web-sources like YouTube, Weather forecast, RSS Feeds, Stocks, etc., or from external third party applications
- Widgets have the following features:
 - They are movable, resizable, configuration, sharable and are also removable from the Dashboard.
 - Widgets allow end-users to get and share a personalized view of their information
 - Widgets load UI elements only once, and in subsequent requests only new data are loaded. This leads to faster performance.
- Widgets are not stand-alone, i.e., they will not run in a browser page, just by calling their URL
 - For a 3DDashboard Widget to run, it has to be first hosted on a Dashboard

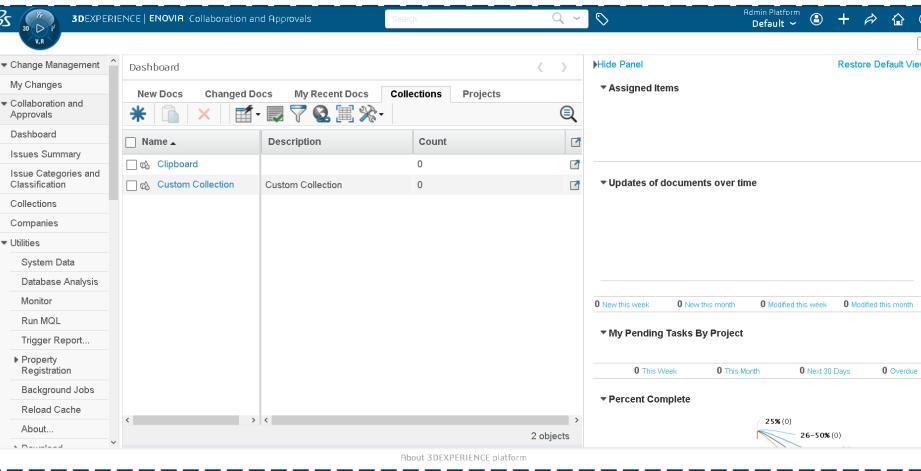
DS do not provide any framework for designing the Widget UI. Customers may choose to use other standard frameworks, similar to the ones mentioned below, at their own risk:

- Backbone JS
- jQuery
- Angular JS

3DEXPERIENCE platform: Classic Web-App versus Widget



Legacy Webtop



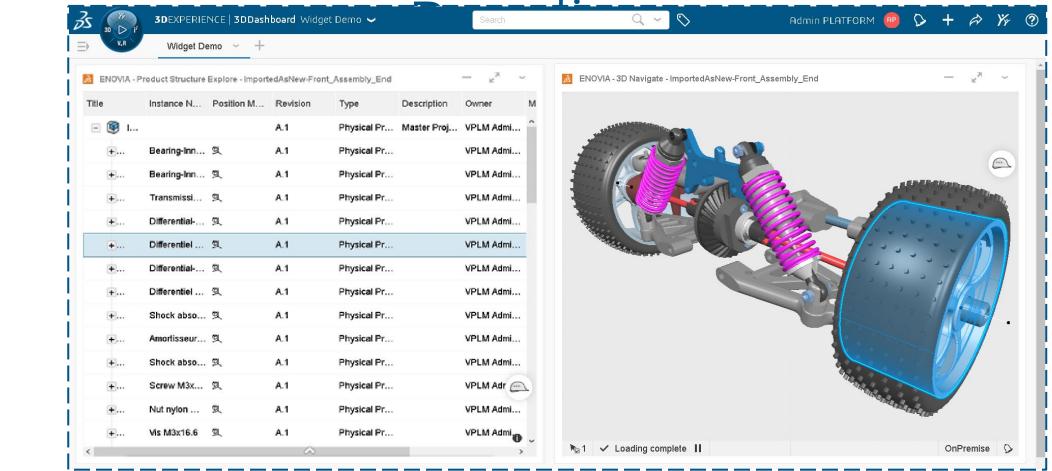
Pages in HTML & JavaScript

Load UI Elements AND
Data for each request



JSPs, Servlets, BPS
Business Logic in JPOs
3DEXPERIENCE Kernel
DB

Dashboard



Widgets in JavaScript (UWA API) & HTML5

Load UI Elements Once
Then Data only at each exchange



RESTful WEB Services
Business Logic in JPOs
3DEXPERIENCE Kernel
DB

Widget – Kinds of Widgets



- ❑ A Dashboard app is a **3DEXPERIENCE** platform app instantiated within a tab page of a 3DDashboard
- ❑ There are the three different ways to integrate a web application in a dashboard of the **3DEXPERIENCE** platform

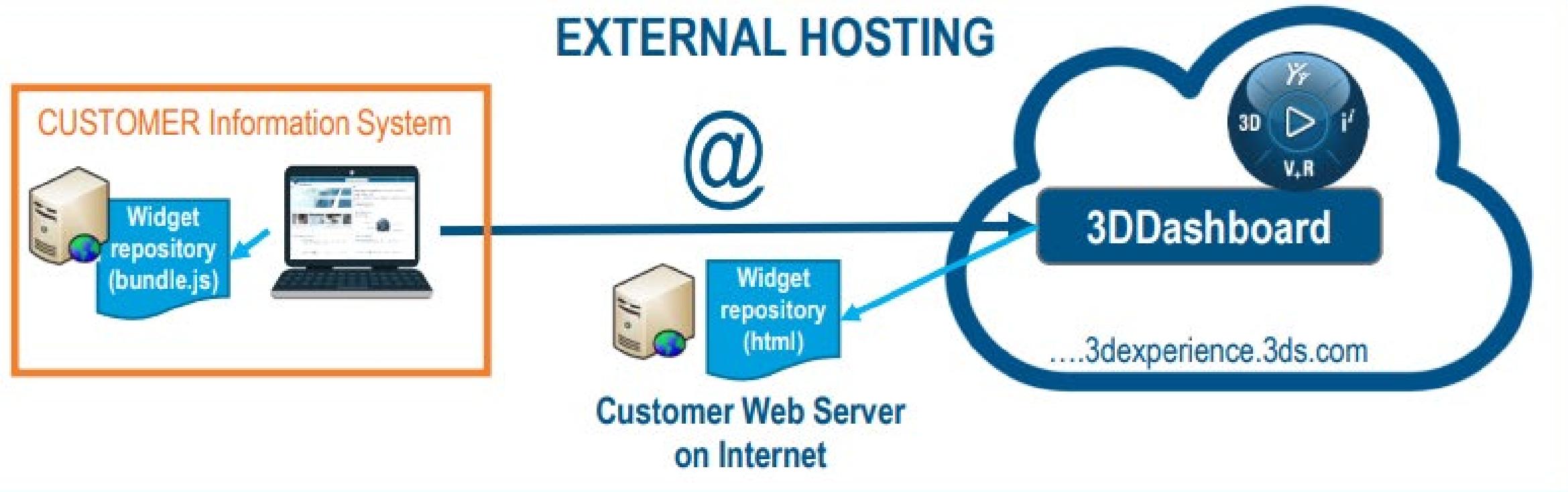
| | Re-use a Web App (*) | Create a Widget App (**) | |
|---|---------------------------|-------------------------------|--------------------------------|
| Functionalities | Using Web Page Reader | Using Run your App Widget | Defining an Additional App |
| Integrate into a dashboard, organize, move, resize, share | Yes | Yes | Yes |
| Takes advantage of 3DEXPERIENCE User experience features: Interacts with other widgets | No | Yes | Yes |
| Access 3DEXPERIENCE Services through REST Web Services | No | Yes | Yes |
| Trusted | No | No | Yes |
| Sharing Mode | As Widget | Yes | Yes |
| | As Dashboard | Yes | Yes |
| | As Icon on Compass | No | No |

* Refers to a web-application which is capable of running stand-alone in any browser
** Refers to a web-application based on UWA protocol - i.e. as a widget

Widget HTML Skeleton Hosting

Prior to R2024x

Customer had to expose a 3DDashboard accessible widget repository on the Internet



New in R2024x

A Widget skeleton can be imported within 3DDashboard
Both hosting options remain available





- Widget loaded by 3DDashboard as Additional App, must be an **UWA widget**, i.e., it must run in the UWA protocol.

- The following standard events are supported for a Widget by the UWA protocol, such as:
 - **onLoad**: Triggered when the widget is launched
 - **onRefresh**: Triggered when the widget is refreshed (manually or programmatically)
 - **onEdit**: Triggered when edition of preferences begins
 - **endEdit**: Triggered when Edit Preference dialog box is closed (save or cancel buttons are pushed)
 - **onResize**: Triggered when the widget size (height and width values) changes. It can happen if the end-user resizes the widget's window, or if the widget's contents increases/decreases the window's height
 - **onViewChange**: Triggered when the widget is windowed, maximized or minimized
 - **onKeyboardAction**: Triggered when a key is pressed within the widget's area

- Custom Widgets may be developed with the following features:
 - Interaction with platform components – 3DSearch and 6WTags
 - Perform Drag and Drop of data
 - Respond to Publish and Subscribe events
 - Interact with other platform services like 3DSpace, 3DDrive, 3DSwym, etc., using web-services

- Custom Widgets must be deployed in a separate dedicated web server, outside the platform servers.

- More information about Widget Development can be found in [Developer's Guide](#).

Please refer to *3DEXPERIENCE Widget Development Fundamentals* course for more details regarding the process.

OOTB Widget Configuration

```
{  
    "name": "CDFBaseOil",  
    "group": "attributeGroup1"  
},  
{  
    "name": "CDFInnerDiameter",  
    "group": "attributeGroup1"  
},  
{  
    "name": "CDFOuterDiameter",  
    "group": "attributeGroup1"  
},  
{  
    "name": "CDFLengthOfRoller",  
    "group": "attributeGroup1"  
}  
]  
,  
"groups":{  
    "attributeGroup1":{  
        "NLS": "Bearing Properties",  
        "weight": 0  
    }  
},  
ENOVIA - Information - Physical Product00000009
```

Physical Product00000009 A

Properties

Bearing Properties

| CDFBaseOil | Ester |
|-------------------|-------|
| CDFInnerDiameter | 0 |
| CDFOuterDiameter | 0 |
| CDFLengthOfRoller | 0 |

Attributes

| Type | CDFRollerBearing |
|-------|--------------------------|
| Title | Physical Product00000009 |

Introduction

In this chapter, we will get introduced to concept of OOTB Widget Configuration.

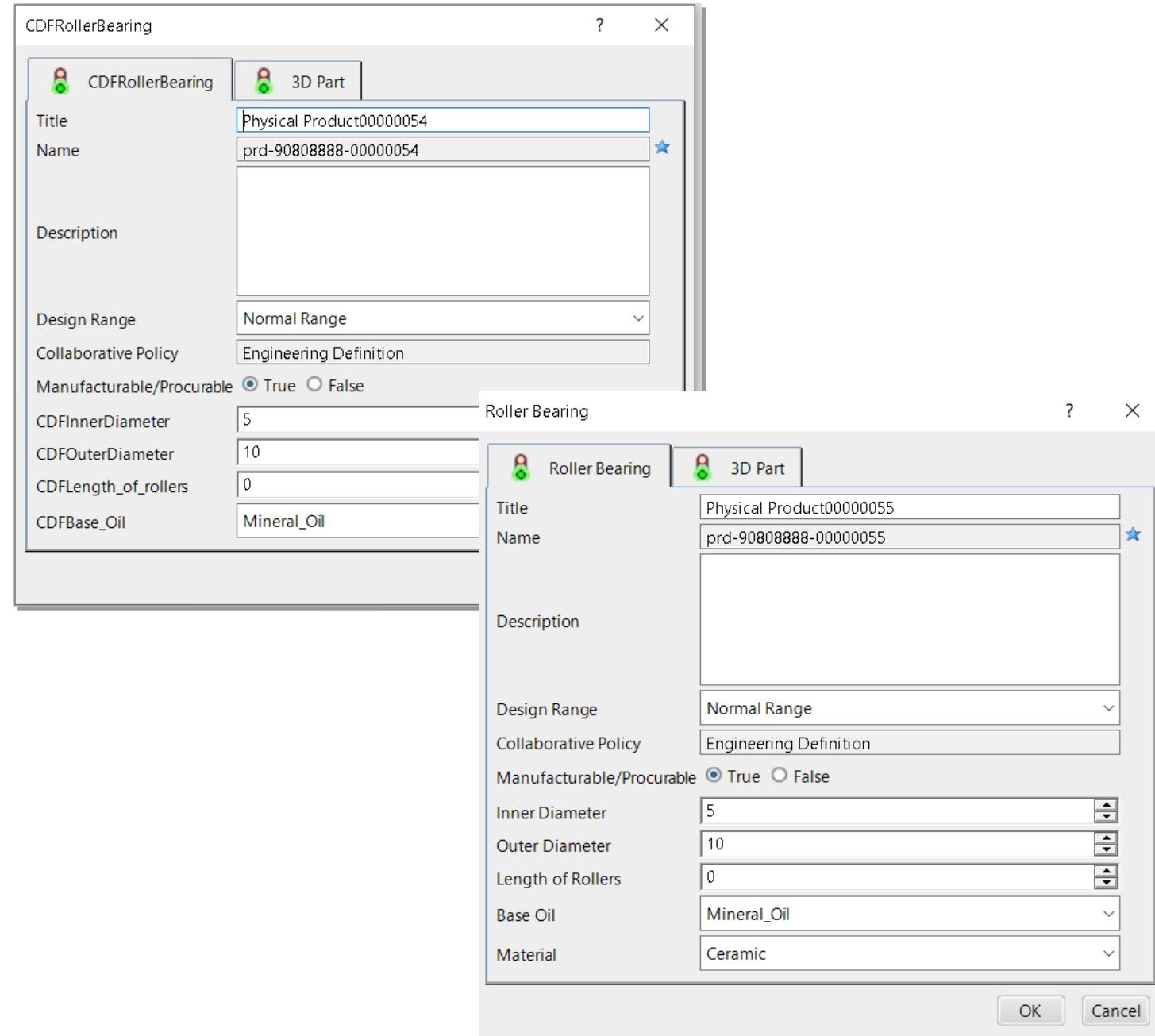


- ❑ ENOVIA – Information widget allows user to view and modify the properties of an object.
- ❑ It is not available in Compass but can be launched only using *Information* option available on other widget application.
- ❑ ENOVIA – Information widget is organized in tabs.
- ❑ These tabs are different depending on the app they are launched from, and the type of object that is selected.
- ❑ Below are frequently used tabs
 - **Information/Properties**
 - The **Information** or **Properties** tab displays details about the object.
 - The list of properties available in this tab depends on your app and the type of object you select.
 - **Comments**
 - The **Comments** tab displays the comments added to an object for collaboration
 - **Relations**
 - The **Relations** tab displays the related objects of the selected object.
 - **Revisions**
 - The **Revisions** tab displays the object's revision graph.
 - **Sharing**
 - The **Sharing** tab lists the permissions granted for an object to other users, and let you manage them.



- ❑ The content displayed in the Properties tab of the ENOVIA Information widget can be configured.
- ❑ **Administrator** role is essential to configure the layout.
- ❑ Administrator can configure below options:
 - The properties you want to display or hide
 - The order in which the properties are displayed
 - Whether the properties are read only or editable
 - Whether the properties are mandatory or not
 - Forbidden characters for a specific property's value
 - Authorized values for a specific property, their order, and the default value
 - The sorting of authorized values
 - Groups of attributes and their order
- ❑ We can configure both the Properties panel and Creation panel (*Product Structure Editor*).
- ❑ We can display and customize:
 - Attributes associated with a type
 - Customer-defined attributes added to a default type using Collaborative Spaces Control Center
 - Customer-defined attributes specified in deployment extensions or specialized types
- ❑ For more information, follow the [link](#)

Internationalization



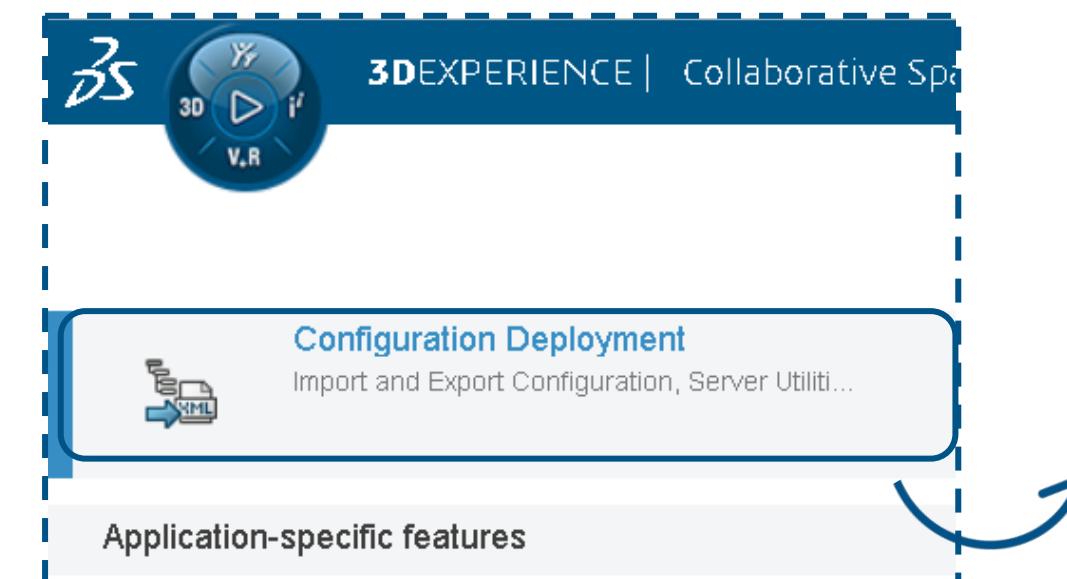
Introduction

In this chapter, we will get introduced to concept of Internationalization.

National Language Support (NLS) Principles (1/2)



- To translate the displayed names of specialized Administrative Objects (Types, Extensions and Attributes) based on the local language, Administrator needs to perform changes on NLS files generated by Data Model Customization App.

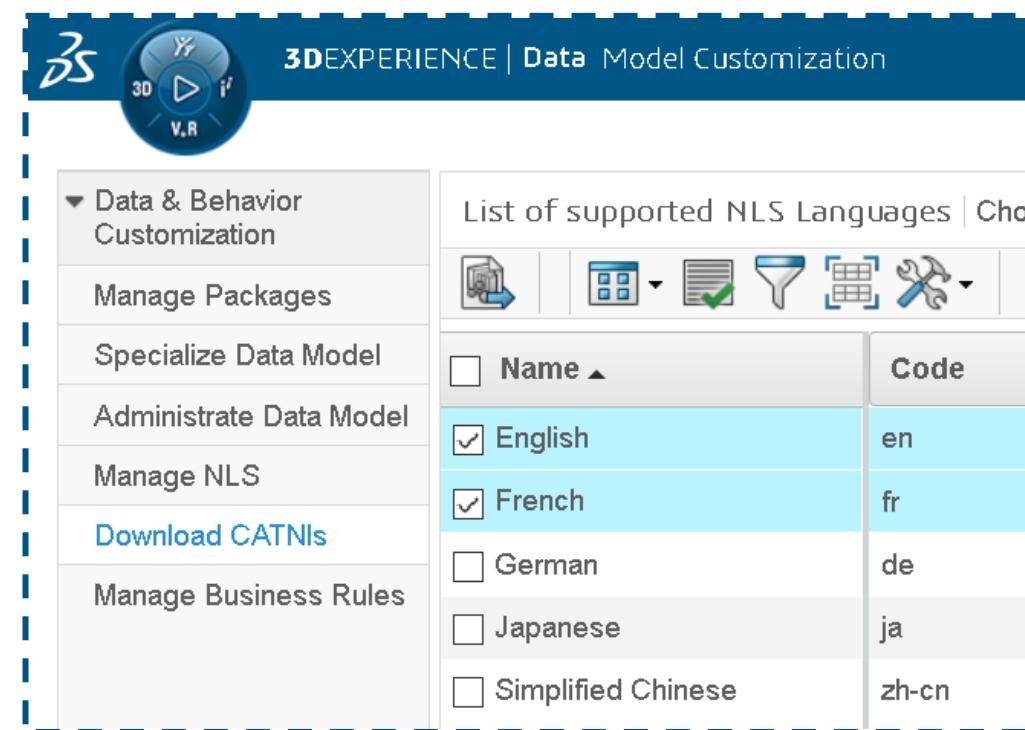


Configuration Deployment

Import and Export Configuration, Server Utilities, Download CATNLS ...

- ▶ Import and Export Configuration
- ▼ Configuration and Server Utilities
 - Update Index Model with Added or Removed Attributes Update Index Model
 - Reload Server Cache Reload Cache
 - Download CATNls Files (ZIP File) Download

OR



| This PC > Downloads > PackagesCATNls.zip | | |
|--|-------------|--------|
| Name | Type | Compre |
| French | File folder | |
| CDF_RequirementDeployment.CAT... CDFBearingMaterial.CATNls CDFBearingPackage.CATNls ClassificationDefaultPack.CATNls OtberConfiguration.CATNls OtbIRPCConfiguration.CATNls TRGPartPackage.CATNls | CATNLS File | |



- ❑ Using **Collaborative Space Configuration Center**, Administrator can download NLS files only for those configurations which have been implemented using the same app.
- ❑ For Data Model Customization done using the **Data Model Customization App**, NLS files can be downloaded from the **Download CATNls** command.
- ❑ The following steps are to be performed by Administrator to download and deploy NLS files for custom Data Model.
 - Download NLS file package by using either *Collaborative Spaces Configuration Center* dashboard widget, or by using the *Data Model Customization* web-app, as shown in the previous page. In both of these cases, a Zip file containing the NLS files will be downloaded.
 - Extract the downloaded Zip file to the directory as referenced by the **CATMsgCatalogPath** variable in user's application setup. The default location is:

`<Native_Apps_Install_Directory>\win_b64\resources\msgcatalog`

➤ After extracting the above contents to the mentioned location, restart Native Apps to reflect the translated names of the custom Types, Extensions and Attributes.

Presentation Layer Configuration Overview



Here are the key takeaways from this lesson:

- Widget app is based on standard web-technologies (HTML5, JavaScript, CSS)
- Widget App can be integrated with dashboard using 3 different ways
 - Web Page Reader Widget
 - Run Your App Widget
 - Additional App Widget
- Administrator can configure OOTB widgets.
- Internationalization can be configured using NLS files.

Integration Layer Configuration Overview



—

After completing this lesson you will:

- Have basic understanding of Enterprise Knowledge Language (EKL)
- Have basic understanding of Automated Business Logic – Precommit Openings
- Have basic understanding of Component Application Architecture (CAA)
- Have basic understanding of Web-Services
- Have basic understanding of Event Publishing



```
Let ID(String)
Let idName(String)
Let counter(String)
Let ObjectCounter(String)

ObjectCounter="VPMReference"
counter=GetUniqueKeyFromString(ObjectCounter)

ID="DS_Training_"+counter
idName="Training-"+counter
```

Introduction

In this chapter, we will be introduced to Enterprise Knowledge Language (EKL) that is used to configure Native Apps



- ❑ Enterprise Knowledge Language (EKL) is the language used to define programs and procedures specific to apps. This language includes **knowledge packages, types, methods and functions** where:
 - **Knowledge Packages** : Provides essential information on the types, methods, and functions that you can access when working with the browser.
 - **Types** : The different objects you can handle.
 - **Methods and Function** : Manipulation of object and its attribute
- ❑ EKL is oriented to provide a quick development environment to enable easy customization to automate task execution on Client or Server.
- ❑ EKL is a DS, portable, object oriented, interpreted, Automation language that directly manipulates the V6 objects:
 - DS: designed, engineered, and maintained by Dassault Systèmes.
 - Portable: independent from OS
 - Object Oriented: manipulates objects with methods
 - Interpreted: does not require compilation
 - Automation: enables to create & manipulate existing classes but does not enable to declare new classes.
 - There is an extension called **KML(Knowledge Modeling Language)** that adds some object modeling capability. This is part of **Company App Builder Role (KAC)**.
 - V6 objects: provides native access to V6 constructs



- ❑ *Enterprise Knowledge Language (EKL)* is used by a series of toolkits & products on top of V6 architecture. They are used for the below purpose:
 - Business Rules
 - Validation Rules
 - Generative Rules
 - Associative Rules
 - User Interface Logic Rules
- ❑ DS Developers will also use it to describe & deliver Business Rules default implementation for their modelers.
- ❑ Administrators (at customer site) will use it to describe Business Rules that will adapt our products & foundations to their business processes.
- ❑ Customers will use *EKL* to implement & execute rules from a Knowledge Automation or a Quality assurance perspective.
- ❑ Apps like *Data Setup*, *Data Model Customization App*, etc. may be used to work with Business Rules.

Automatic Business Logic – Precommit Openings

The screenshot shows a software interface for managing business logic. At the top, there are two tabs: 'Resource Tables' (selected) and 'Know-how Apps Packages'. Below the tabs, there is a search bar labeled 'Resource Set ID / Table Name' with a lock icon and a dropdown menu. The main area displays a list of resources under the heading 'ENOLibraryBuisnessLogicServer'. One item, 'ER generic server openings', is highlighted with a blue border. A tooltip for this item states: 'Description : Business Rules executed on server at initialization and precommit operations'. Below this, a table titled 'Resource Table' lists various precommit operations, each associated with a 'Business Rule'. The table has columns for 'Logical Name' and 'Resource Type'. The first row, 'precommit at promote', is highlighted with a teal background.

| Logical Name | Resource Type |
|---------------------------------|---------------|
| precommit at promote | Business Rule |
| precommit at demote | Business Rule |
| precommit at create | Business Rule |
| precommit at connect | Business Rule |
| precommit at delete | Business Rule |
| precommit at disconnect | Business Rule |
| precommit at modify attribute | Business Rule |
| precommit at modify description | Business Rule |
| precommit at change name | Business Rule |
| precommit at change owner | Business Rule |
| precommit at copy | Business Rule |
| precommit at major revision | Business Rule |
| precommit at batch clone | Business Rule |
| precommit at batch revise | Business Rule |

Introduction

In this chapter, we will be introduced to Automatic Business Logic using *Precommit Openings*

Precommit Openings (1/2)



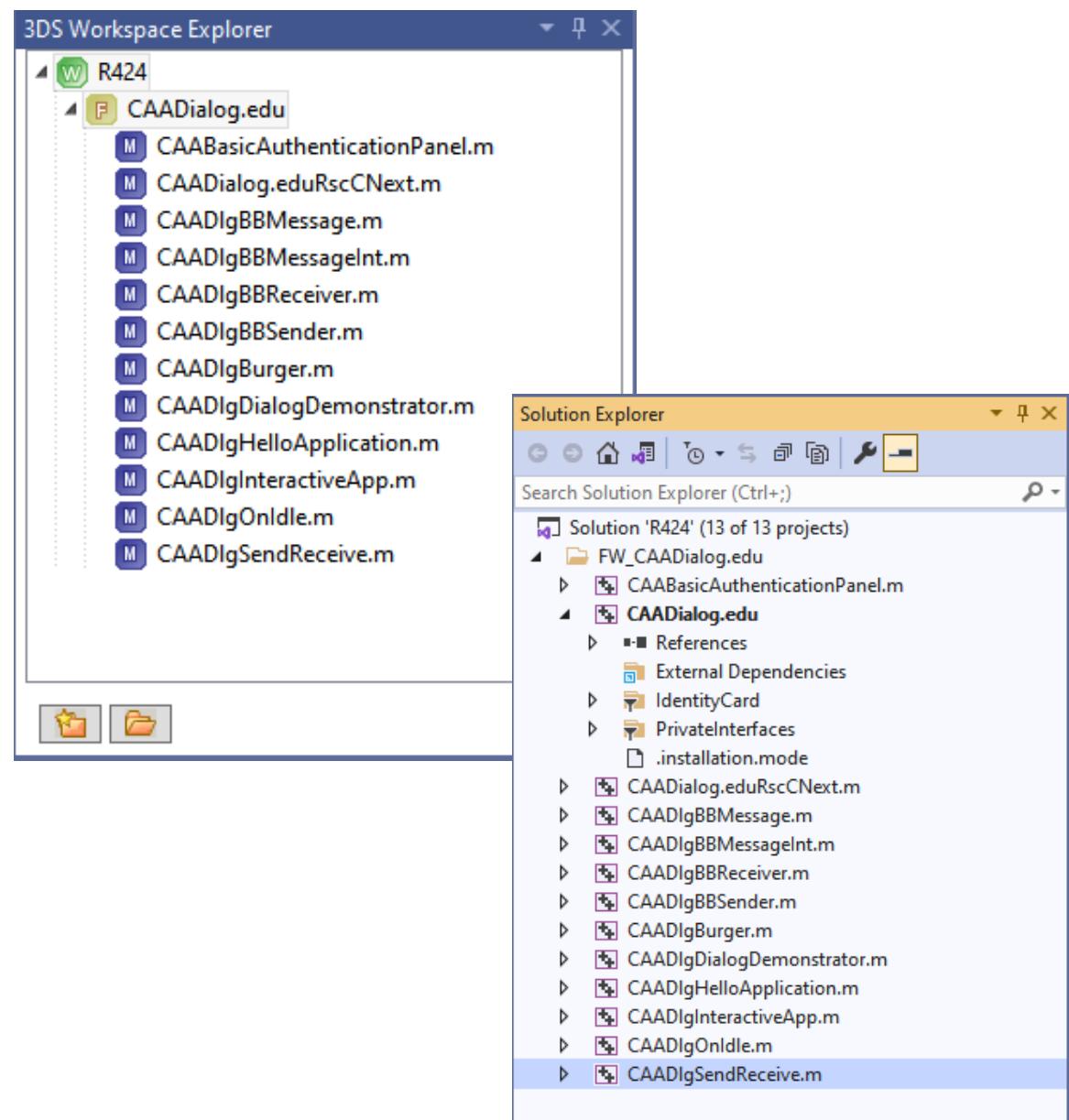
- ❑ Precommit openings allow you to customize server operations like *create, delete, modify, promote*, etc. on Private cloud, Dedicated cloud and On Premises.
- ❑ Precommit openings are *not available on public cloud*.
- ❑ Business Logic for Validation on Server resource set is now **deprecated** and should not be used for any new implementations.

The screenshot shows a software interface for managing business logic. At the top, there are tabs for "Resource Tables" and "Know-how Apps Packages". Below this, a search bar contains the text "Ve...on". A dropdown menu is open, showing a list of resource sets and packages, with "ER generic server openings" selected. To the right, a table titled "Resource Table" displays a list of precommit openings. The table has columns for Logical Name, Resource Type, Resource, Criteria, and Description. The first row, "precommit at promote", is highlighted in green. The table includes a note: "Description : Business Rules executed on server at initialization and precommit time". There are checkboxes for "Group by type" and "Show resource preview".

| Logical Name | Resource Type | Resource | Criteria | Description |
|---------------------------------|---------------|----------|----------|---|
| precommit at promote | Business Rule | | | this opening is fir...he object promotion |
| precommit at demote | Business Rule | | | this opening is fir...he object demotion |
| precommit at create | Business Rule | | | this opening is fir...e the object creation |
| precommit at connect | Business Rule | | | this opening is fir...e object connection |
| precommit at delete | Business Rule | | | this opening is fir... the object deletion |
| precommit at disconnect | Business Rule | | | this opening is fir...bject disconnection |
| precommit at modify attribute | Business Rule | | | this opening is fir...tribute modification |
| precommit at modify description | Business Rule | | | this opening is fir...ption' modificatio |
| precommit at change name | Business Rule | | | this opening is fir...'name' modification |
| precommit at change owner | Business Rule | | | this opening is fir...ject changes owner |
| precommit at copy | Business Rule | | | this opening is fir...the object is copied |
| precommit at major revision | Business Rule | | | this opening is fir...ect (major) revision |
| precommit at batch clone | Business Rule | | | this opening is fir...atchclone operation |
| precommit at batch revise | Business Rule | | | this opening is fir...tchrevise operation |



- ❑ Business rules using Precommit openings has two options
 - Create a precommit opening business rule and deploy the business rule
 - We can only create business rules and resource sets for Precommit openings using **Data Setup** app in the development environment.
 - **ER generic server openings** resource table must be used.
 - **ER generic server openings** resource table contains the Precommit openings openings.
 - We do not need to bind precommit opening business rules to a collaborative space in Data Setup
 - Import the server business rules into another platform
 - Use **Enterprise IP Schema Specialization** app to import business rules.
 - This is applicable only for **Private and Dedicated cloud**
- ❑ Execution of business rule with a precommit opening is different from other openings. The transaction follow below sequence of operations:
 - The kernel registers the transaction
 - The kernel executes the action (e.g. create or promote)
 - The kernel fires the precommit opening
 - If no error occurs, the kernel commits the transaction.
 - Else the transaction is aborted.
- ❑ For more information documentation is available at
 - <https://help.3ds.com>
Installation and Setup > 3DEXPERIENCE Platform > 3DEXPERIENCE Configuration and Customization > Behavior > Data Setup > List of Resource Set IDs > Infrastructure Business Logics > ER Generic Server Openings



Introduction

In this chapter, we will be introduced to concept of CAA.

- Component Application Architecture (CAA) is the Unified architecture to develop Applications

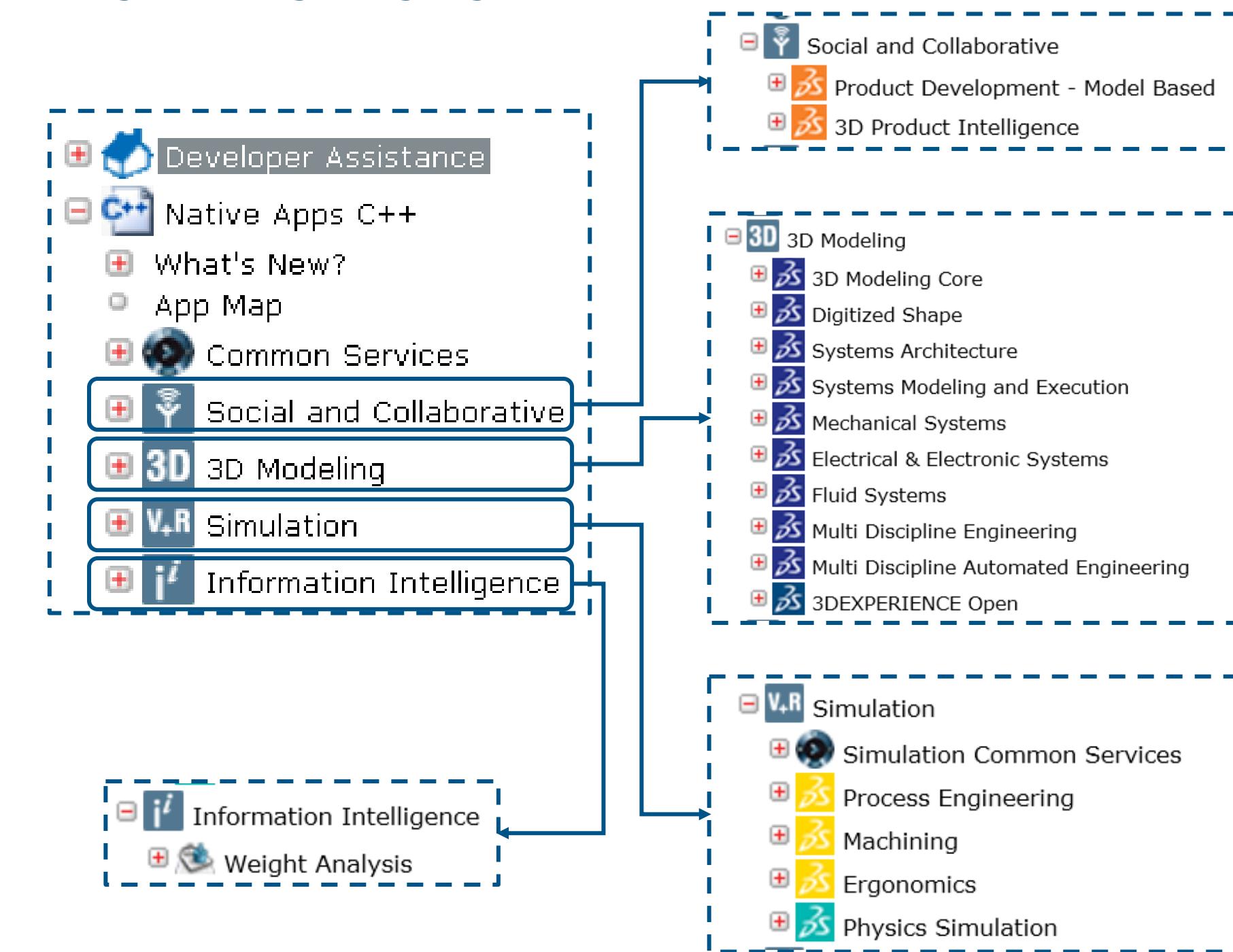
- It involves the usage of **Compiled, Object Oriented Programming Language**

- It makes use of C++ and COM technologies
- Allows to build dynamic link libraries (dll)
- Enhances implementation encapsulation
- Enables objects behavior federation

- Large scope of customization

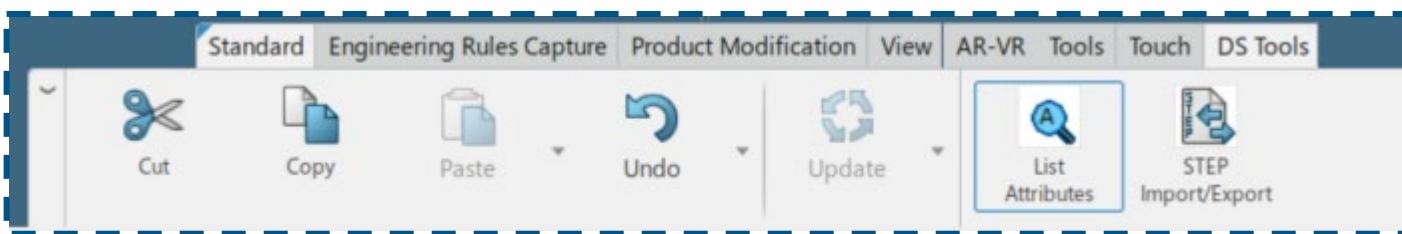
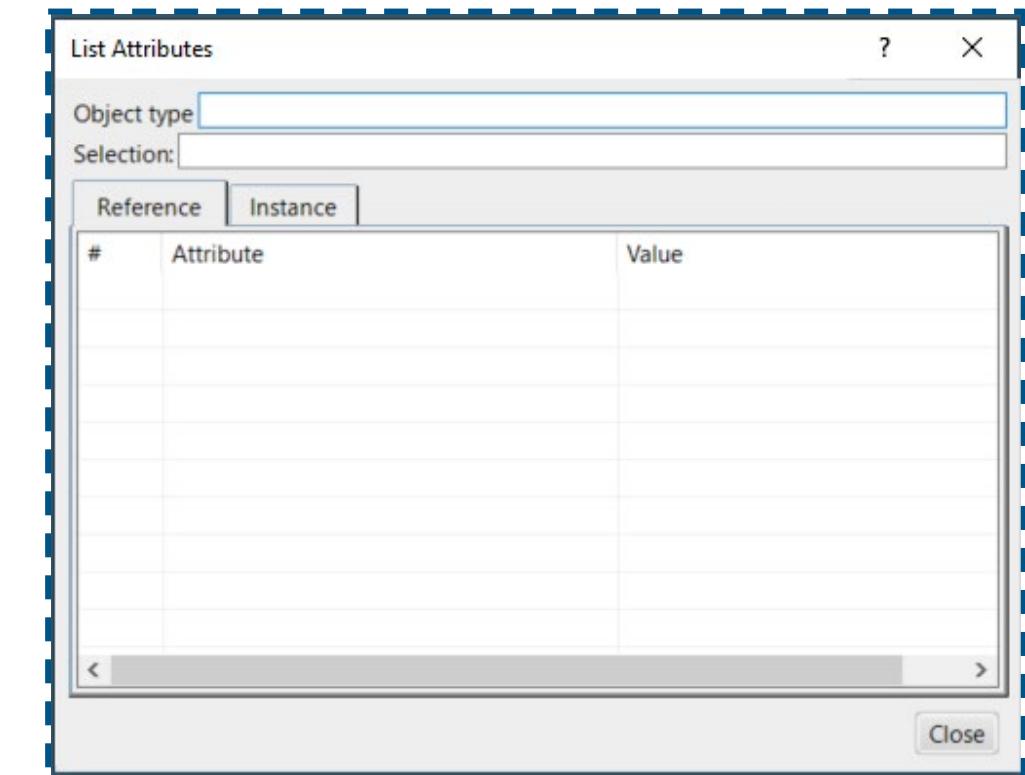
- IP Integration
- 3DEXPERIENCE Model Customization
- Batch Creation
- User Interface Modeling for Native Apps

- CAA provides capability to integrate customer's processes and industry knowledge into DS products





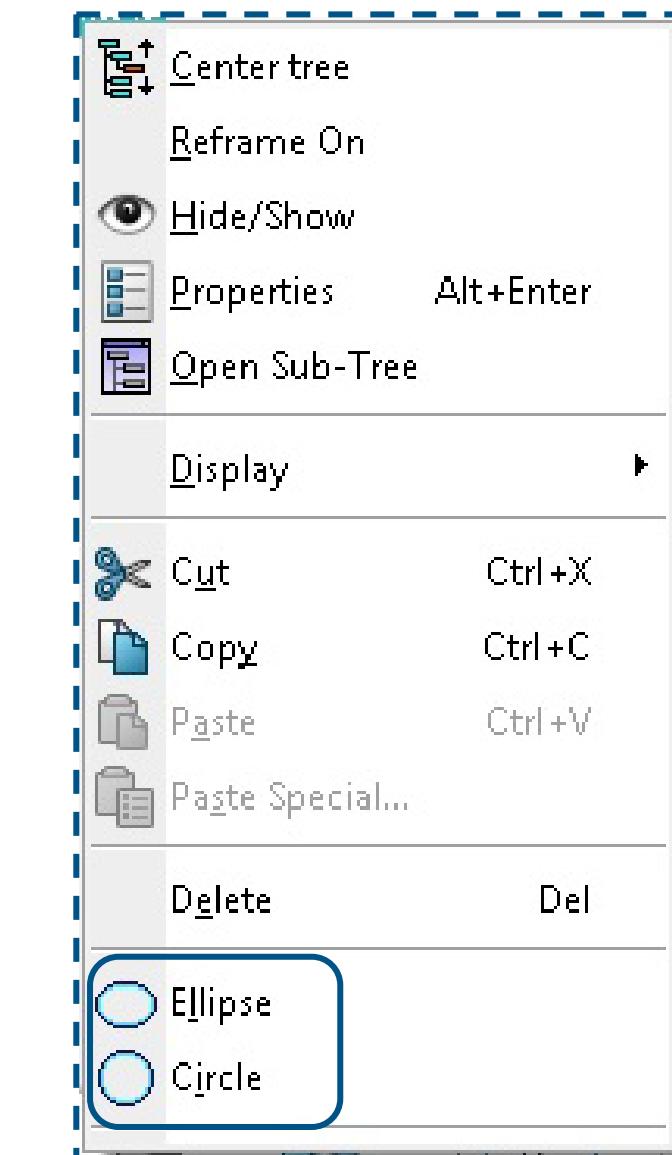
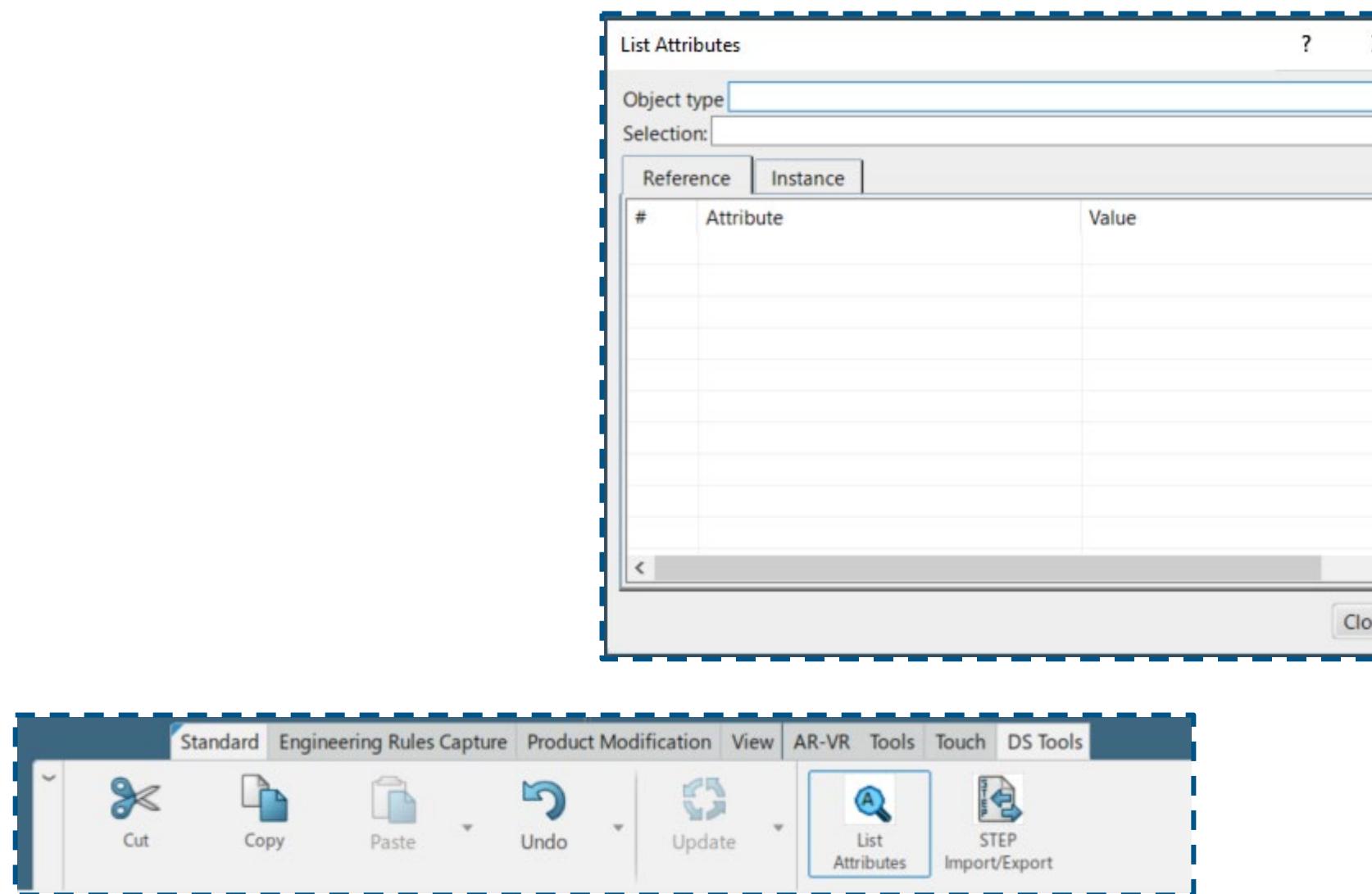
- ❑ IP Integration
 - Integrate customer's Intellectual Property in CATIA, SOLIDWORKS, DELMIA and SIMULIA applications
 - Read and modify attributes
- ❑ User Interface
 - Create custom commands and dialog boxes
 - Create custom commands for use in different authoring sessions
- ❑ Batch Creation
 - Creating executable files
- ❑ CAA can be exposed to VB
- ❑ CAA can be exposed to EKL
- ❑ CAA can execute a web service



User Interface Modeling – Capabilities



- Create your own **workbench** to address specific type of document
- Create customized **toolbars** for existing workbench
- Create **Dialog Boxes** to interact with end users

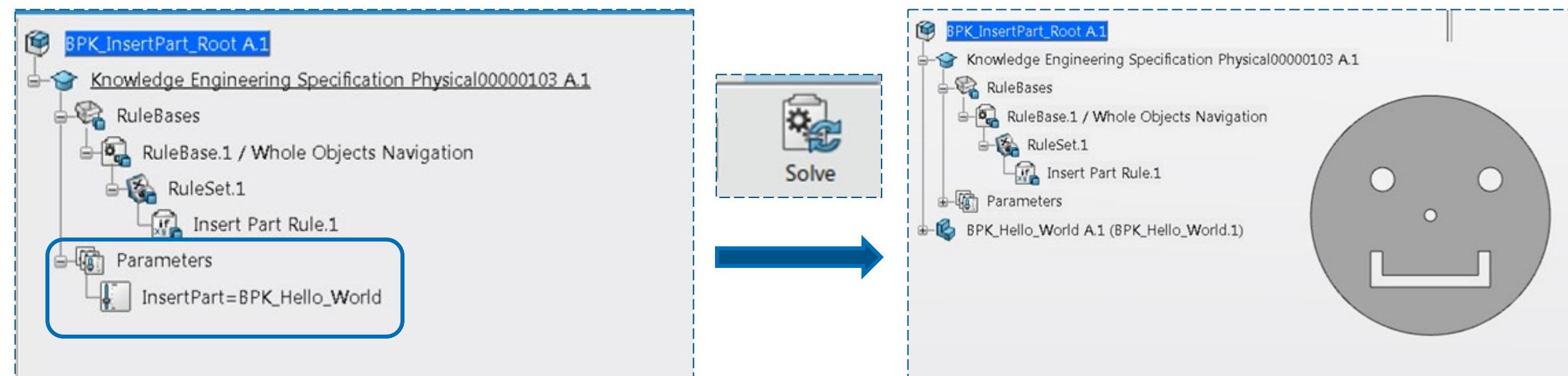


Contextual Command



Expose CAA to EKL – Sample Scenario:

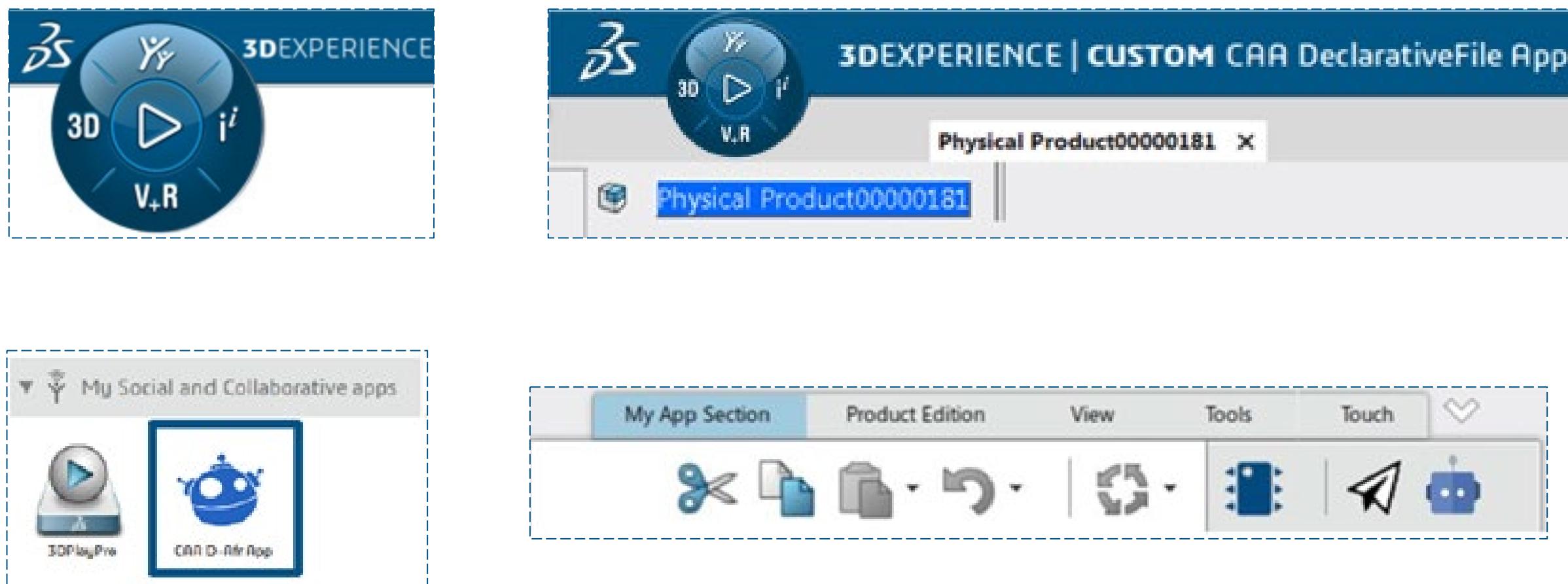
- ❑ In the below sample scenario, CAA was leveraged to read the **InsertPart** parameter value, search the database for a part and insert the same into the model.
 - This CAA interface was exposed to EKL.
- ❑ When the user clicks on the **Solve** button, the EKL script executes the exposed CAA interface to perform task.



Native Apps User Interface Customization – Example 2

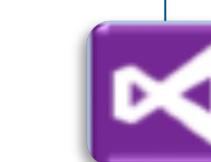
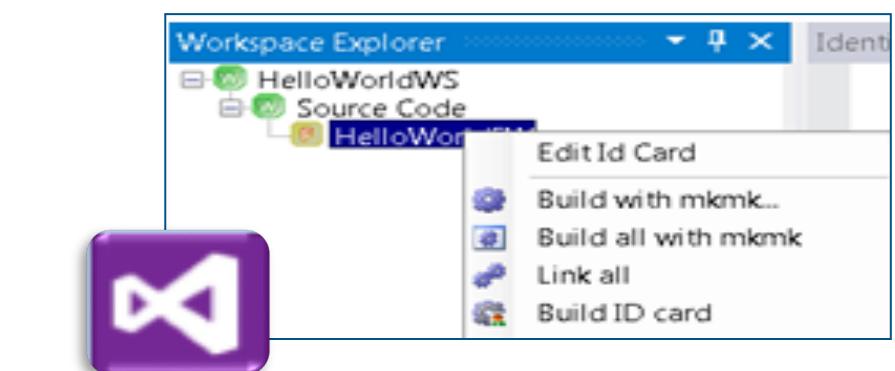
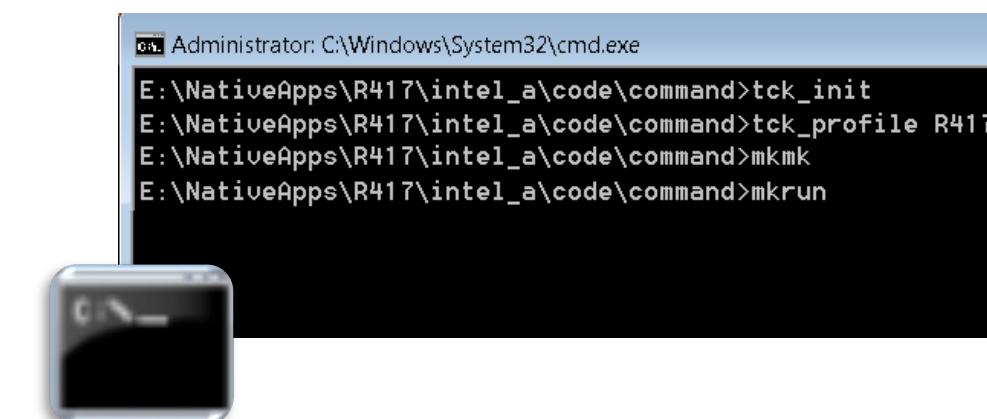
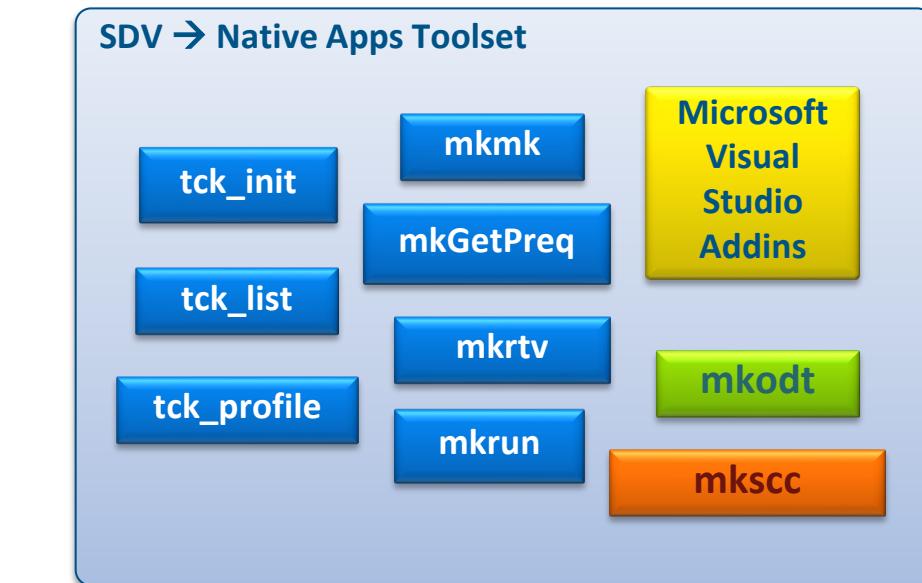


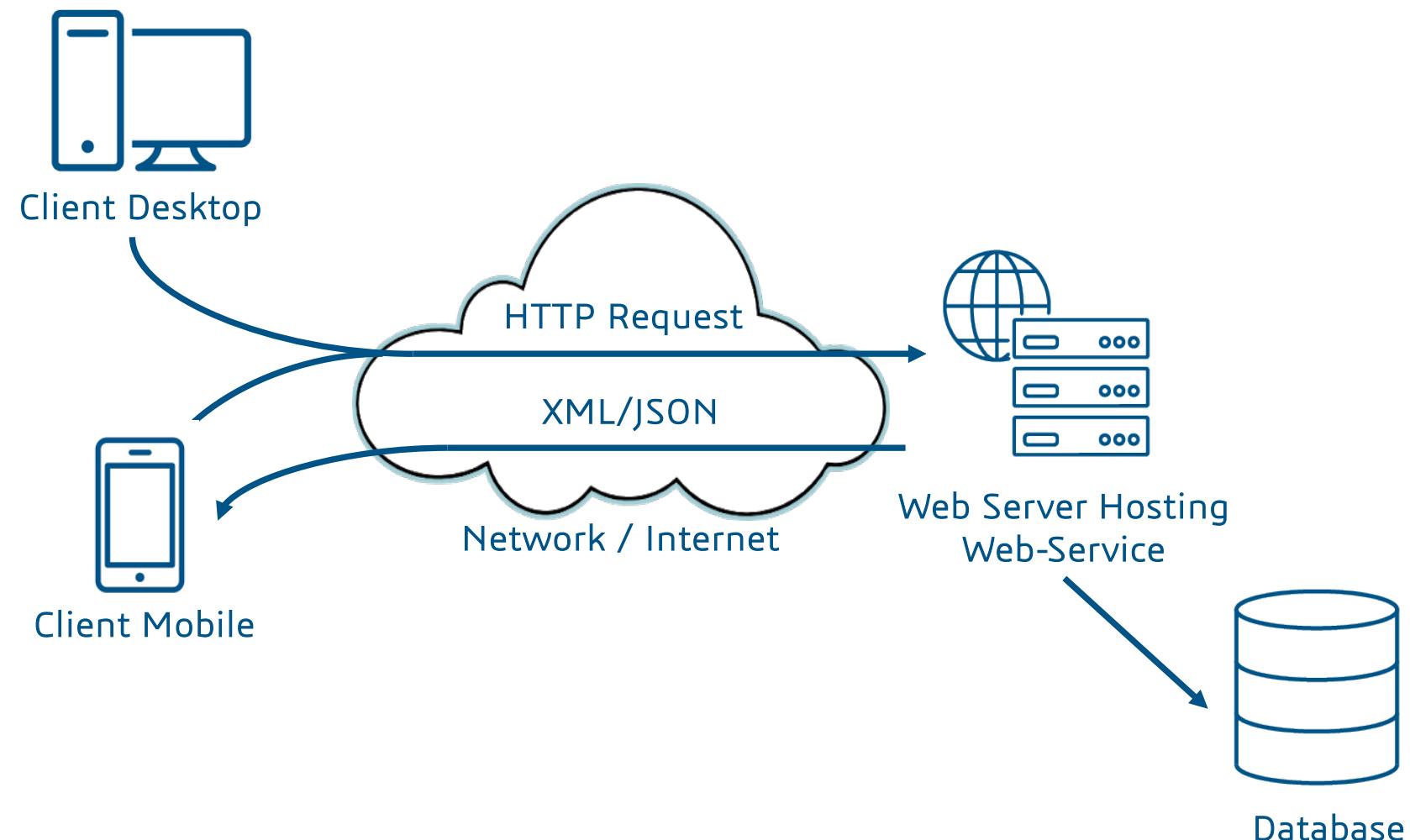
- You can create your own App and Sections and tools or add to an existing one
- In the following example, a Custom App has been created in the Compass, with which custom tools have been made available.





- Native Apps Development Toolset is a platform of developer tools
- It is integrated in the *Application Development Suite (SDV)* product
- It proposes the following tools for Native client development
 - App Builder, Compilation and run tools
 - Microsoft Visual Studio® integration
 - Unit Test Manager Product
 - Source Checker
 - Dialog Designer
- Each configuration provides a set of tool commands
 - Integrated in Microsoft Visual Studio® user interface
 - Used as it is through command prompt





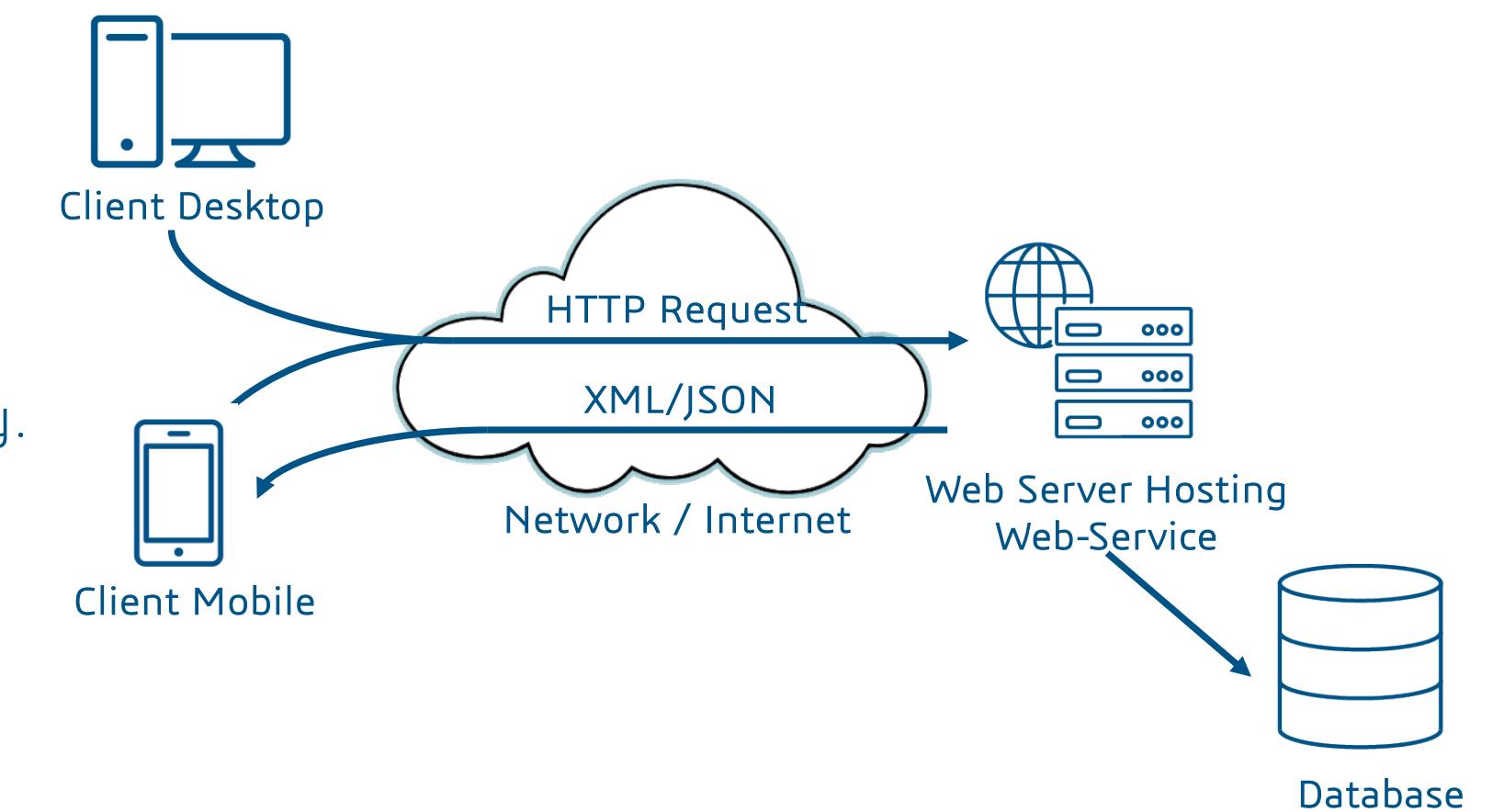
Introduction

In this chapter, we will be introduced to Web Services and its utility in **3DEXPERIENCE** platform

Introduction



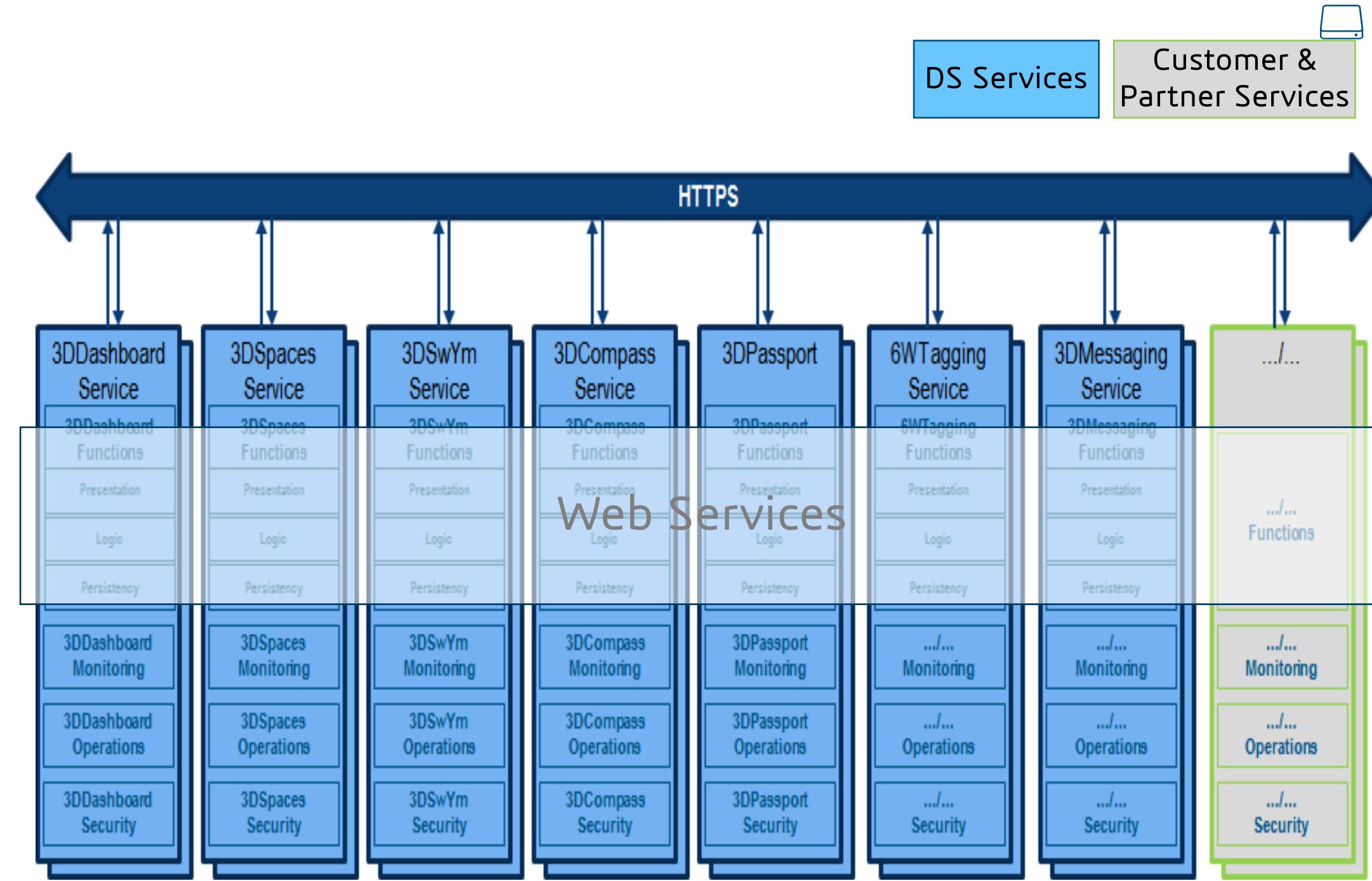
- A **Web-Service** is any standard Application that is available over the network (Internet) which is capable of interacting with other Applications over the same network.
- In simple terms, a **Web-Service** is an application that responds to a call made by another application over the network.
- As a response, the **Web-Service** can:
 - Perform a task. For example, create some new data in the Database
 - Send back a response to the calling application, in a standard exchange format. The response format can be in XML, JSON, etc.
- Mainly, there are two types of Web-Services:
 - SOAP Web-Services
 - RESTful Web-Services
- Both SOAP and REST are to be used over HTTP as the transport methods.
- Web-Services are typically designed to allow interoperability.
This means that:
 - Web-Services are operating system independent.
 - Web-Services are programming language independent.





- REST stands for REpresentational State Transfer.
- REST is an architectural style and uses HTTP protocol for communication
- REST defines a set of constraints to be used for creating web services
- In REST Architecture everything is a resource, identified by URI
- The resources can be Text Files, Html Pages, Images, Videos or Dynamic Business Data
- REST is stateless web service i.e. server does not maintain the application state for any client
- REST defines a uniform way of interacting with a given server irrespective of device or type of application
- Resources are manipulated using HTTP Method (GET, POST, PUT, PATCH, DELETE).
 - GET - Retrieve resource from the server
 - POST - Create a new resource
 - PUT - Replace a resource with another one, or fully modify a resource
 - PATCH - Update a resource (diff only)
 - DELETE - Delete the resource

- 3DEXPERIENCE platform app servers communicate with each other, as well as with external services using REST web-services.



- The list of web-services that are exposed by 3DEXPERIENCE platform OOTB can be found in **Developer Assistance** at this [link](#).



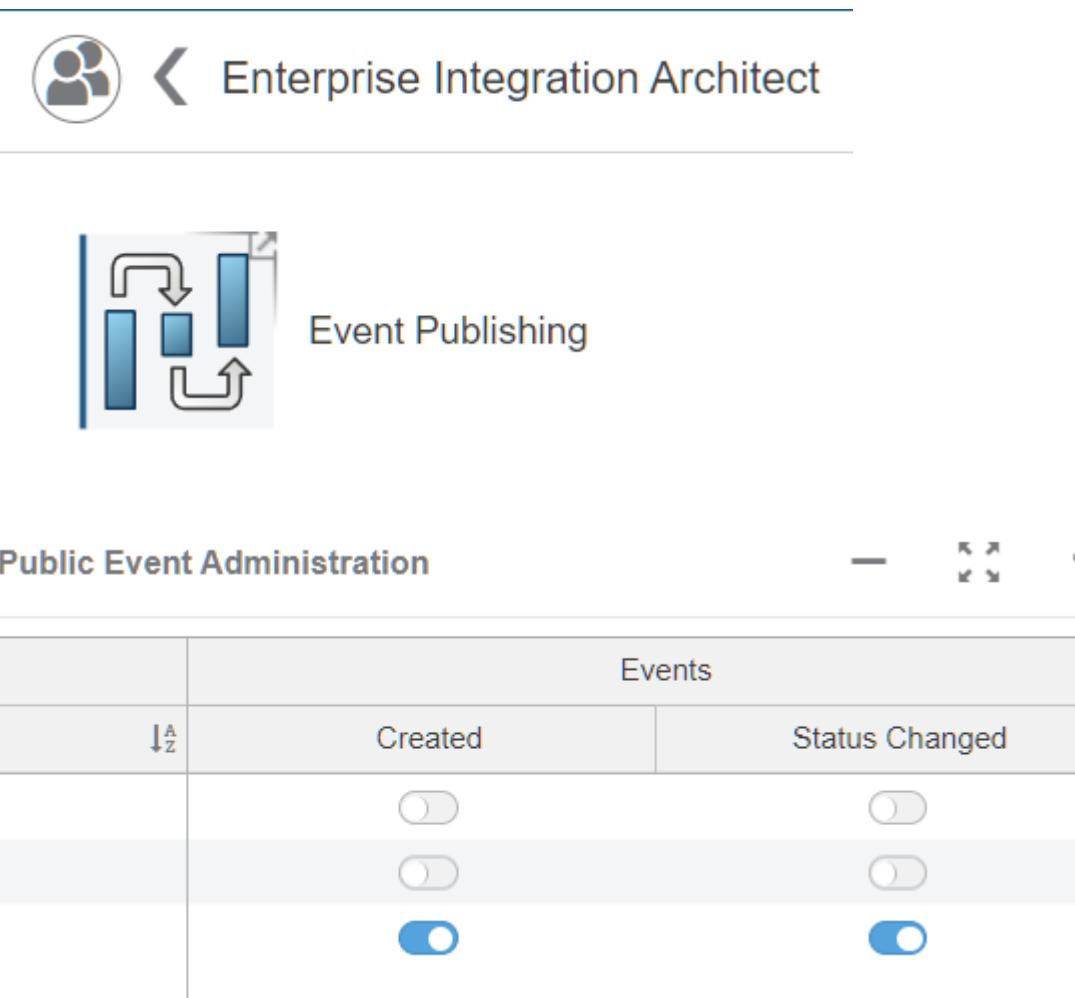
□ There are four contexts to consume 3DEXPERIENCE platform web-services:

- From a dashboard application (Additional App widget), one can reach 3DEXPERIENCE services. The 3DEXPERIENCE authentication (username/password) has to be performed manually by the concerned person.
 - For calling 3DEXPERIENCE platform web-services, 3DDashboard Widgets make use of the [DS/WAFData/WAFData](#) JavaScript API.
- From any non-DS applications (web or batch) you can reach 3DEXPERIENCE services using your personal credential. Before consuming any DS web-services, the 3DEXPERIENCE authentication must be executed.
- From any non DS applications (web or batch) you can reach 3DEXPERIENCE services on behalf of someone. The capability is possible through *service agent*, and is available for on-premises or private cloud only
 - Administrators can declare a service agent using 3DPassport Control Center widget. Service agent is a combination of username and passkey.
- On the cloud, you can consume 3DEXPERIENCE web-services on behalf of someone using an *openness agent*.
 - An agent is valid for one platform instance and for one user.
 - Using an agent, you can call only public 3DEXPERIENCE web services enabling the "on-behalf" connection.



Only platform member people may be "on-behalf" person.

Event Publishing



The screenshot shows the Enterprise Integration Architect software interface. At the top left is a user icon and the text "Enterprise Integration Architect". Below this is a toolbar with a magnifying glass icon and the text "Event Publishing". The main area displays a table titled "Events" with columns "Types", "Created", and "Status Changed". There are three rows: "Change Action" (both switches off), "Document" (both switches off), and "Engineering Item" (both switches on). The table has a header row and three data rows.

| Events | | | |
|--------|------------------|-------------------------------------|-------------------------------------|
| | Types | Created | Status Changed |
| 1 | Change Action | <input type="checkbox"/> | <input type="checkbox"/> |
| 2 | Document | <input type="checkbox"/> | <input type="checkbox"/> |
| 3 | Engineering Item | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

Introduction

In this chapter, we will be introduced to concept of Event Publishing.



- Exchange Integration Framework (EIF) allows external systems to integrate with 3DEXPERIENCE platform using a set of services.
- Event Publishing Service is a part of *Exchange Integration Framework*.
- Event Publishing is a service which receives events from 3DEXPERIENCE services and publish messages for internal or external consumption after the associated data is indexed.
- Event Publishing service serves two main functions
 - Enable and manage the public events that the Customer wants to use within his Enterprise integration process
 - Ensure events are published only after indexing of the newly created object.
- Event Publishing app is used to control the events which will be published using the service.
- Enterprise Integration Architect (PFI) and Platform Manager role is needed to access *Event Publishing* app.

Event Publishing – Introduction (2/2)

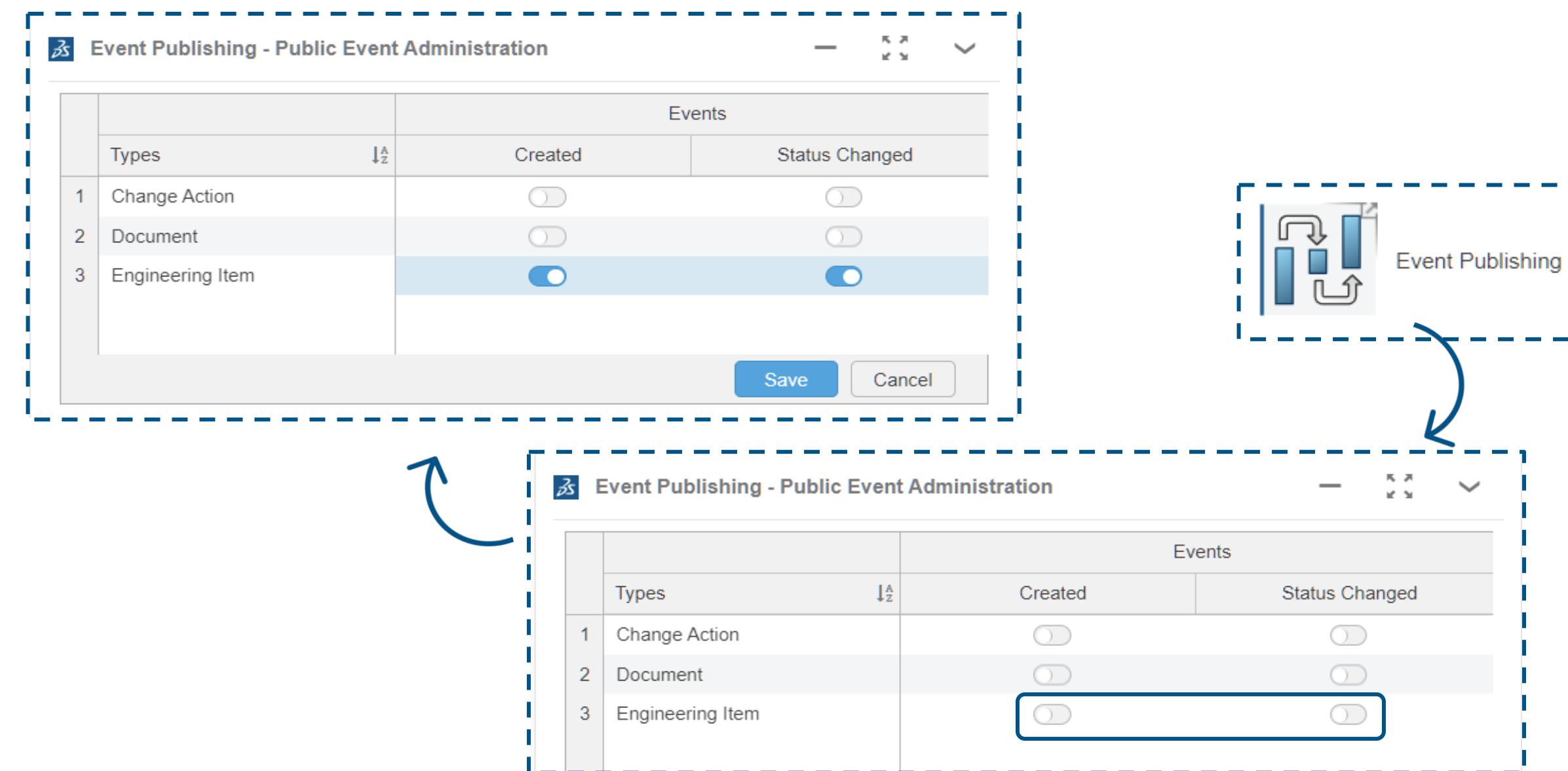


- Administrators can authorize which messages about events can be published for supported object types.
- For On Cloud environment, Event Publishing uses SaaS Secure Message Service to publish the messages.
- For On Premises environment, message broker (*ActiveMQ*, *Kafka*, etc.) need to be configured. Default *ActiveMQ* implementation provided.
- When an event occurs,
 - Event Publishing app adds the message to a queue.
 - Cloud View Indexer indexes the data.
 - The messages are published for indexed data.
- To access *Event Publishing* app, **ENOVIAEventPublishingFoundation** must be installed (GA and FD) on top of 3DSpace.

Configure Events



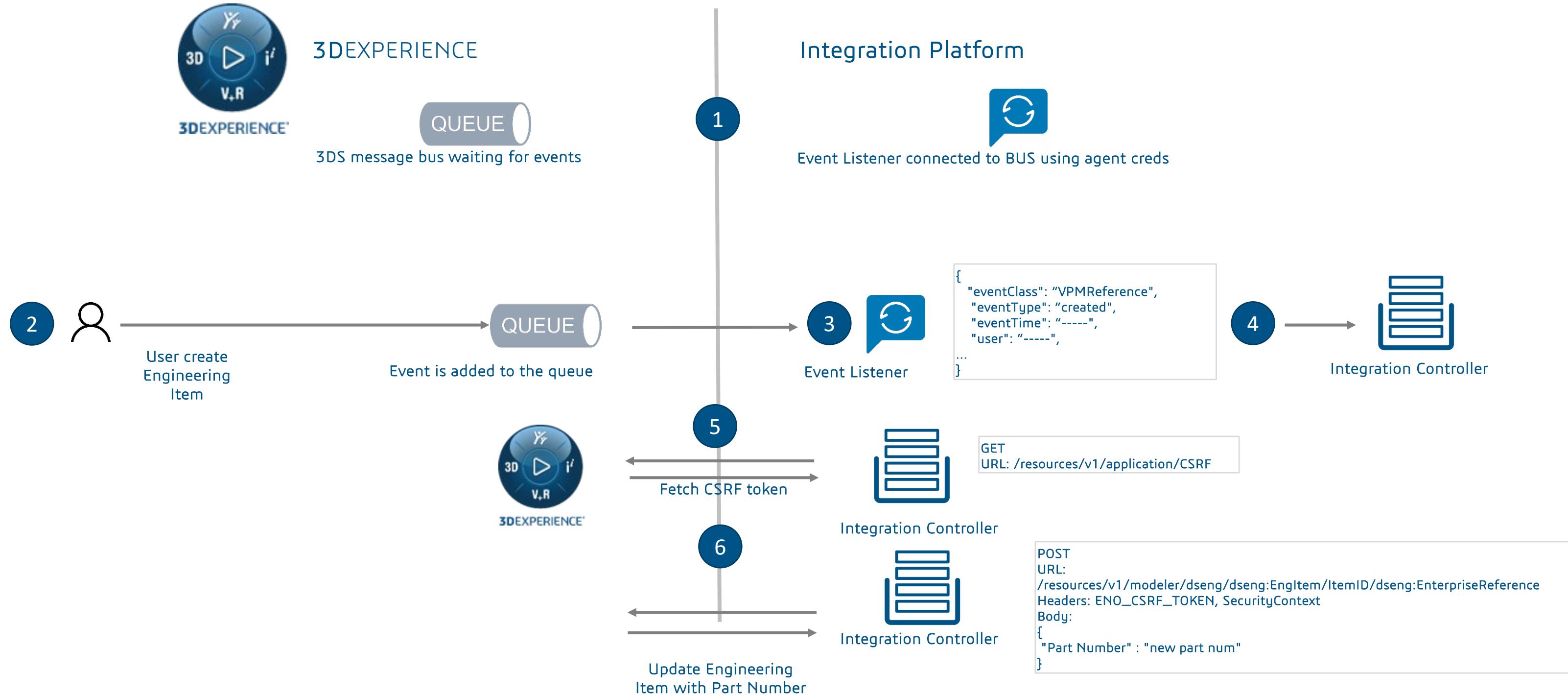
- After configuring event mapping, Platform Manager can use PFI role to launch Event Publishing app to configure events.
- All events are initially disabled.
- Administrator choose events for the supported types and enables them



Event Publishing – Use Case



- Example – Adding Part Number to newly created Engineering Item using Engineering Web Service



Integration Layer Configuration Overview



Here are the key takeaways from this lesson:

- ❑ **EKL** (Enterprise Knowledge Language) is the language of choice for working with Business Rules
- ❑ **CAA** (Component Application Architecture) is used to work with Data Model Customization for Native Apps, develop customized Native Apps, etc.
- ❑ **Web-Services** are used to exchange data between different application servers which are on trusted domain.
- ❑ **Event Publishing** is a service which receives events from **3DEXPERIENCE** services and publish messages for internal or external consumption after the associated data is indexed.

Legacy Utilities



—
After completing this lesson you will:

Have basic understanding of Legacy Utilities

30 Minutes



Introduction

In this chapter, we will be get an overview about the different Administrative Objects used to configure the User Interface for Web apps.

Presentation of the Web Interface



The Home page is divided into three sections, the navigation pane and two panels.

- The left-hand panel provides a list of objects with detailed information about recent document changes, collections, and apps. You can refine the list of objects by filtering types of information and by selecting data that appears in the right-hand panel in the charts.
- The charts provide visual data about pending issues and tasks that the user is responsible for. The default Home page opens when you login.

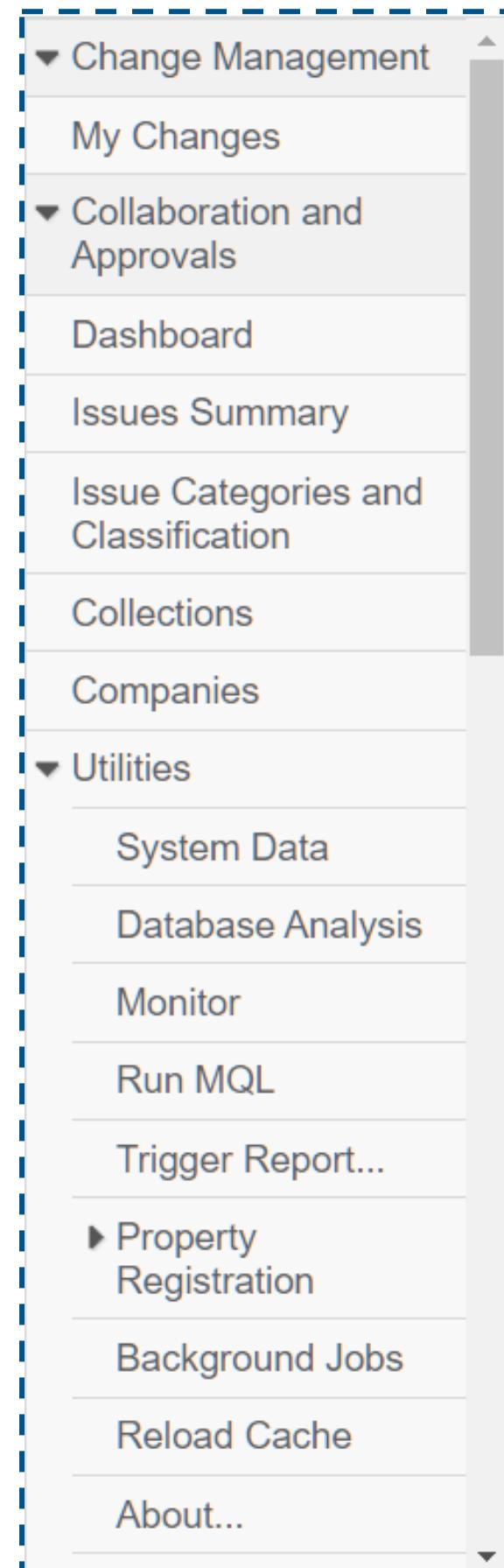
The screenshot shows the 3DEXPERIENCE ENOVIA Collaboration and Approvals web interface. The top navigation bar includes the 3DEXPERIENCE logo, the platform name, a search bar, and user account options. The left sidebar contains a navigation menu with categories like Change Management, Collaboration and Approvals, and Utilities. The main dashboard area displays a grid of documents with columns for Name, Title, Actions, and document status icons. To the right of the grid are several data visualization panels: 'Assigned Items' (empty), 'Updates of documents over time' (empty), 'My Pending Tasks By Project' (empty), and 'Percent Complete' (a chart showing 25% (0) and 26-50% (0)). A footer note at the bottom of the dashboard states 'Documents that have been created within the last 10 days'.

Navigation Pane



A new **Navigation Pane** has been present since the R2015x release, instead of the **My Desk** or **Apps** menus in the past.

- Depending on the current application you are working on (chosen in the compass) you see different items in the **Navigation Pane**.
- Usually each item represents a link to a Portal page or Table with objects of the appropriate Type.
- You can hide the **Navigation Pane** by clicking the triangle arrow on the left border of the page.
- When you are working in a context of an object - **Navigation Pane** automatically displays the categories for the object Type you are working on.



Administrative Objects used for the Graphic User Interface (1/3)



- The *User Interface* is created using standard components called **Configurable Components**

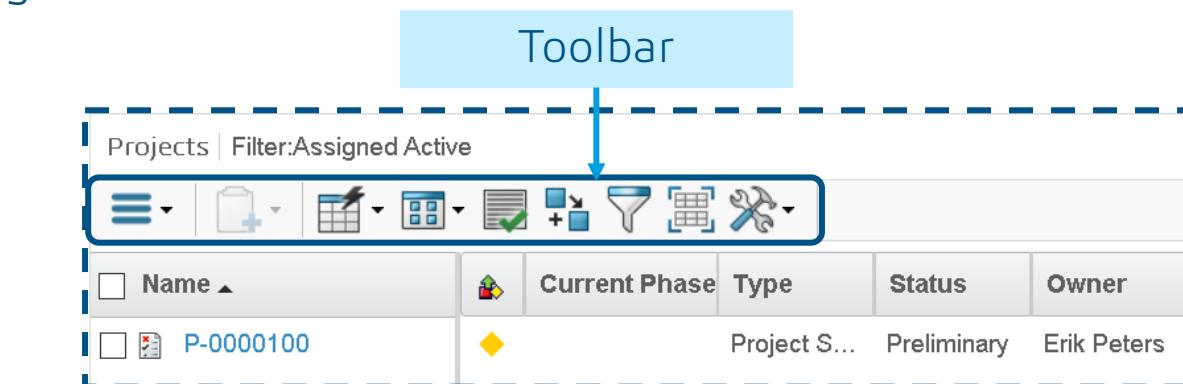
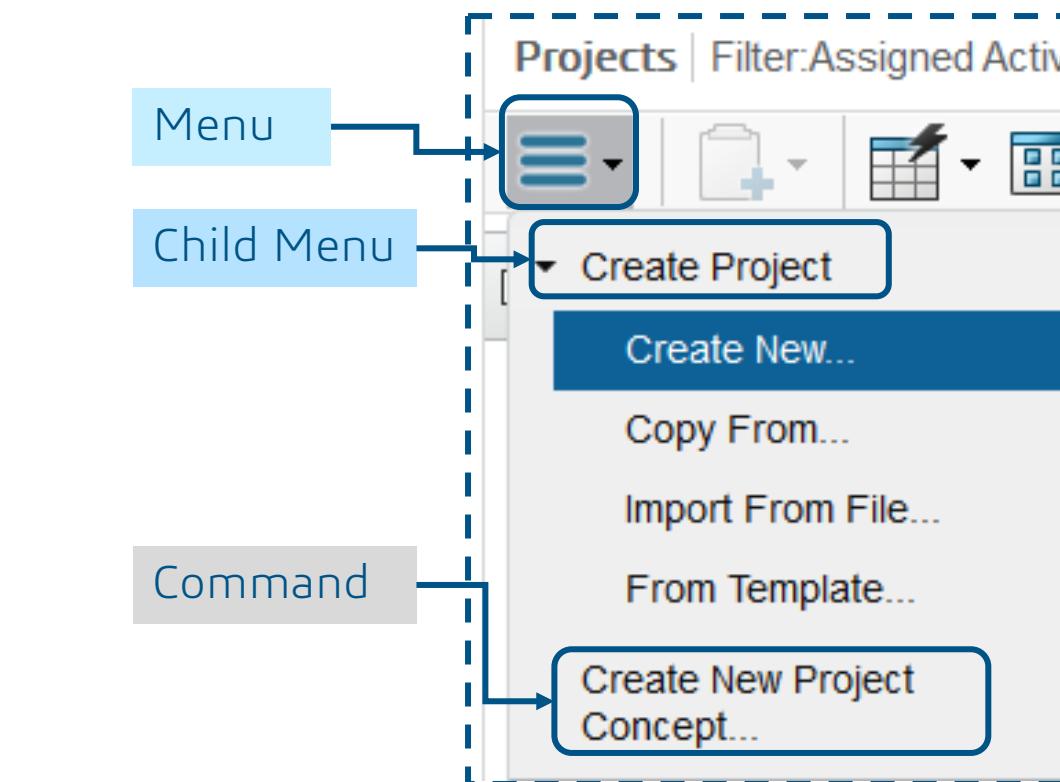
- These Configurable Components are *Administrative Objects*:

- **Menus**

- They are the list of options, from which the user chooses the required option. A menu may consist of child Menus or Commands.
 - They are used also to define the different Toolbars

- **Commands**

- They appear at the last level in the Menus, or as buttons in Toolbars
 - They are used to start an action : display information, create/delete objects, etc.
 - They call specific JSP pages which will display the table, forms, portals etc.



Administrative Objects used for the Graphic User Interface (2/3)



➤ Tables

- Tables are used to display list of elements
- The Table object defines the different columns and controls how data are displayed
- They can be structured or flat with cell update and mass update capability

| Name | Revision | Type | Description | Status |
|--------------|----------|------|-------------------|---------|
| A-0000101-01 | A | Part | Part A-0000101-01 | In Work |
| B-0000100-01 | A | Part | Part B-0000100-01 | In Work |

➤ Webforms

- Webforms are used to display the details of a specific object
- They are used to create/edit/view a specific objects

Part-0000100-01 Maturity : In Work → Frozen Owner : Admin Platform Modified : Mar 21, 2022, 3:05:36 PM

Properties Images Revisions Customer Extension... >

Part-0000100-01 A

Engineering BOM
Equivalents
Change Management
Classification PowerView
Part Specs & Documents
Related Parts
Lifecycle and Collaboration

Part-0000100-01 A

Type Part
Name Part-0000100-01
Title Part-0000100-01 Structure Content Root
Description
Revision A Phase Developer
State In Work Collaborative Policy EC Part
Classification Path Part Family
Derived From
Creation Date 11 Mar 2022 Modification Date 21 Mar 2022
Organization Company Name Collaborative Space CDFTraining
Owner Admin Platform

View Form

Create Part

Fields in red italics are required

Type Part
Name AutoName
AutoName Series General Series
Title
Description
Phase Development
Options Enable Change Required Design Collaboration
Work Under Change
Part Family
Number of Parts 1

Create Form

Administrative Objects used for the Graphic User Interface (3/3)



➤ Channels and Portals

- A Channel will display a specific view (Table or Webform)
- A Portal can display one or many Channels
- These objects are used to create Powerviews

The screenshot displays a Microsoft Dynamics 365 application window with a dashed blue border. Inside, there are two distinct sections separated by a horizontal line.

Channel 1 with 6 Tabs: This section contains the following elements:

- Header:** Includes fields for "Part-0000100-01" (with a "Drop images here" placeholder), "Maturity : In Work → Frozen", "Owner : Admin Platform", and "Modified : Mar 21, 2022, 3:05:36 PM".
- Left Sidebar:** A vertical list of links: Part-0000100-01 A, Engineering BOM, Equivalents, Change Management, Classification PowerView, Part Specs & Documents, Related Parts, and Lifecycle and Collaboration.
- Top Bar:** Navigation tabs: Review, Route, Discussions, Issues, Subscribe, and Sharing.
- Lifecycle:** A section titled "Part-0000100-01 rev A: Lifecycle" with a status bar showing "In Work → Frozen → Approved → Released → Obsolete".
- Bottom Bar:** Icons for Tasks/Signatures, Approvals, and other filters.

Channel 2 with 2 Tabs: This section contains the following elements:

- Approvals Tab:** Shows a table with columns: Name (sorted by Name), Approver, Title, Approval Status, Approval/Due Date, and Comments/Instructions. A message "No Objects Found" is displayed below the table.
- Bottom Bar:** Icons for Tasks/Signatures, Approvals, and other filters.

The Global Toolbar (1/2)

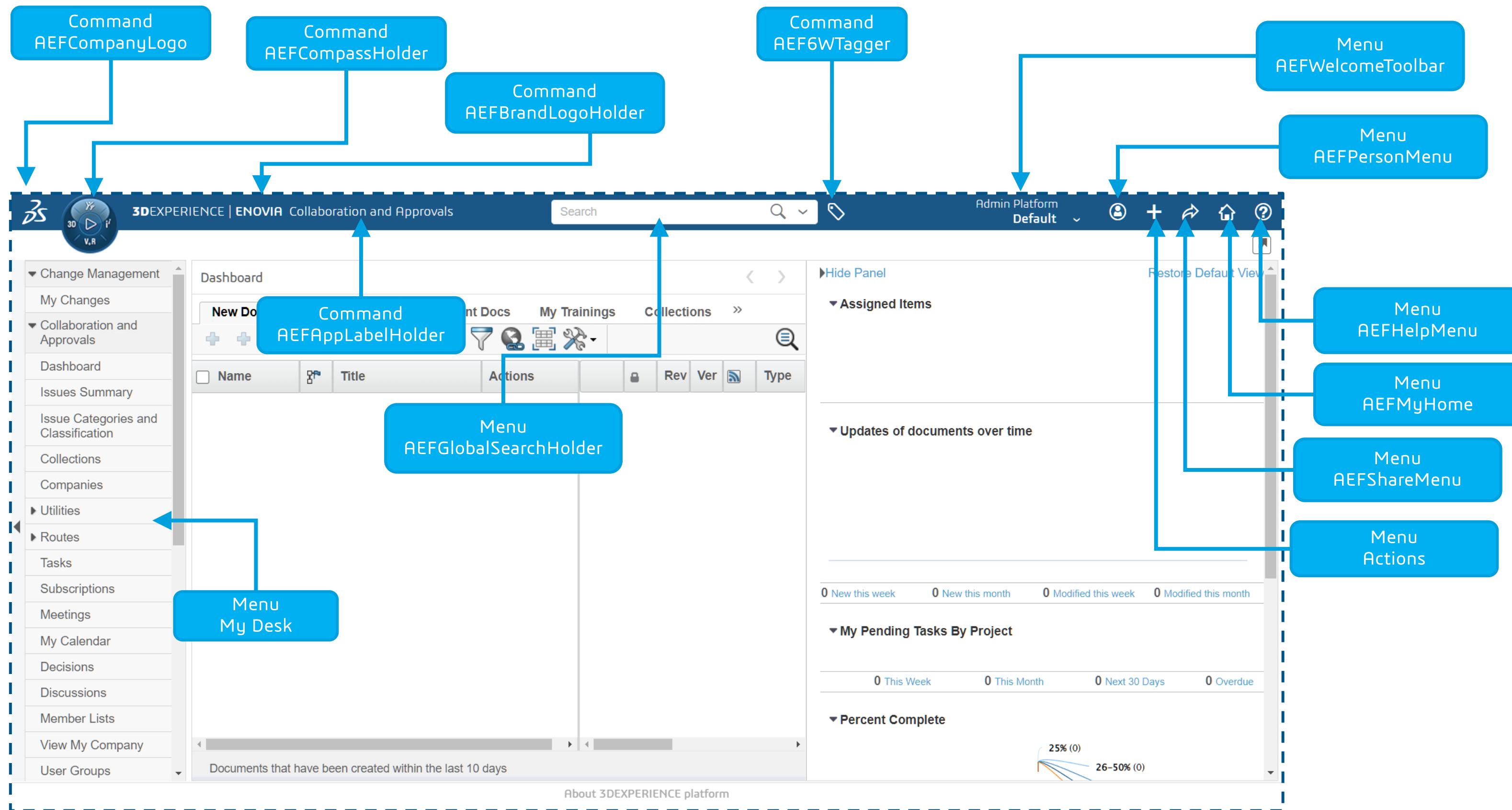


- The Global Toolbar is the main Toolbar that is seen on top of the web-application
- It cannot be modified, but it is possible to add sub-Menus and Commands to the existing Menus
- The Global Toolbar is represented by the AEFGlobalToolbar Menu Administrative Object.
- Use the MQL command as given below, to know the names of linked Commands and sub-Menus.

[print menu AEFGlobalToolbar !history;](#)

```
MQL<5>print menu AEFGlobalToolbar !history;
menu AEFGlobalToolbar
  description
    setting Licensed Product value ENO_BPS_TP,ENO_AEF_TP
    setting Registered Suite value Framework
  children
    command AEFCOMPANYLOGO
    command AEFCOMPASSHOLDER
    command AEFBRANDLOGOHOLDER
    command AEFAAPLABELHOLDER
    menu AEFGlobalSearchHolder
    command AEF6WTAGGER
    command AEFWelcomeToolbar
    menu AEFPersonMenu
    menu Actions
    menu AEFShareMenu
    menu AEFMyHome
    menu AEFHelpMenu
    menu My Desk
  nohidden
  property application value Framework
  property installed date value 01-12-2023
  property installer value ENOVIAEngineering
  property original name value AEFGlobalToolbar
  property version value R215.HF1
  created 1/12/2023 12:09:35 PM
  modified 1/12/2023 12:09:54 PM
```

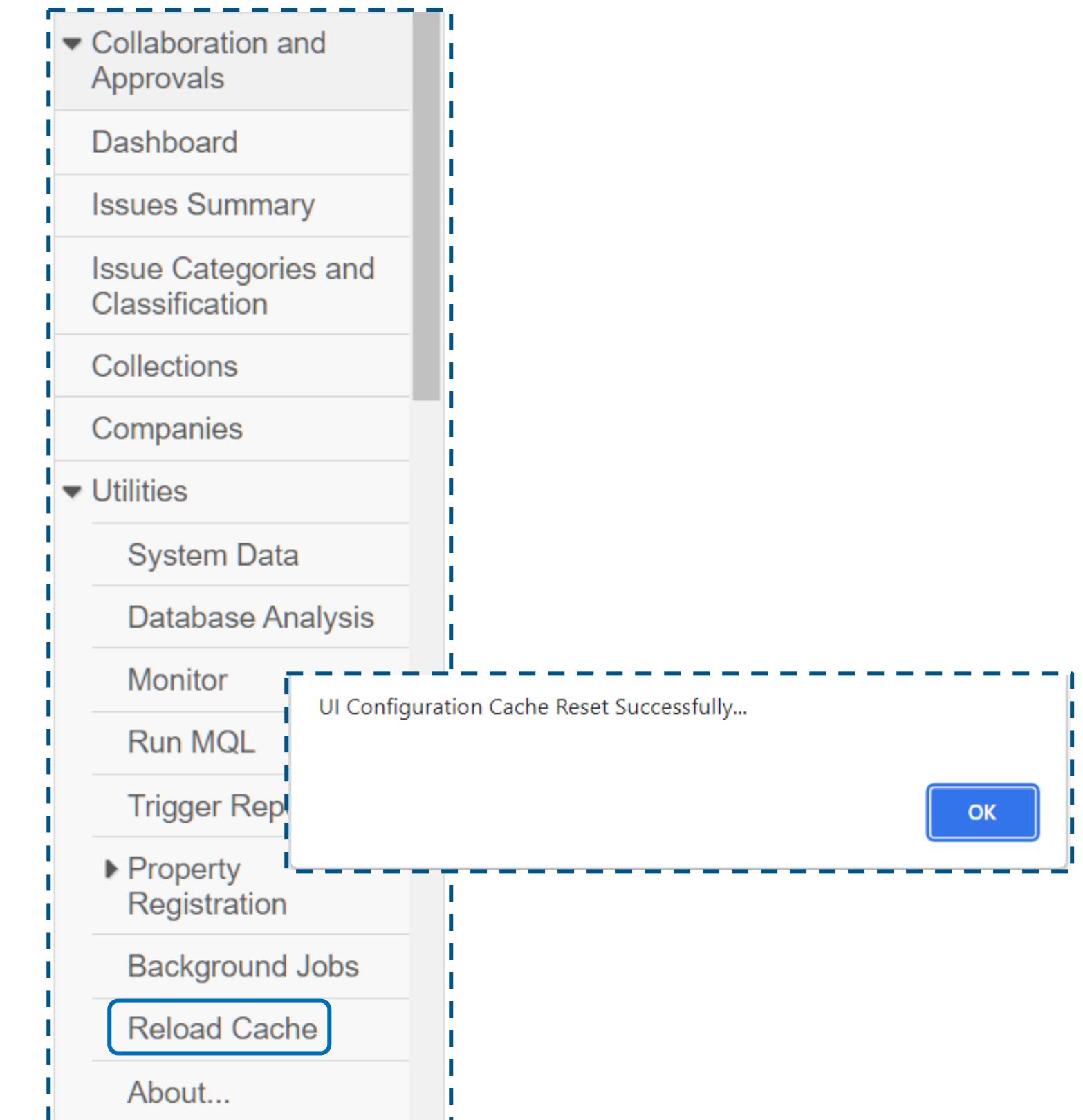
The Global Toolbar (2/2)



Reload UI Cache



- This action is needed to take into account all modification that have been performed on GUI Components
- When the **Reload Cache** tool is clicked, the entire cache data is reset and all previously cached data expires.
- The *Reload Cache* command is available to an user with any one of the below Roles:
 - VPLMAdmin
 - Administration Manager
- After Reloading the Cache, the Browser needs to be refreshed to reflect the changes.
- There are two items that are cached and never get cleared (until application-server is restarted):
 - Properties
 - String Resource Properties
- Program Administrative Objects are never cached





3DEXPERIENCE®

Introduction

In this chapter, we will be get an overview about different Administrative Objects used to configure the Business Logic for Web apps.

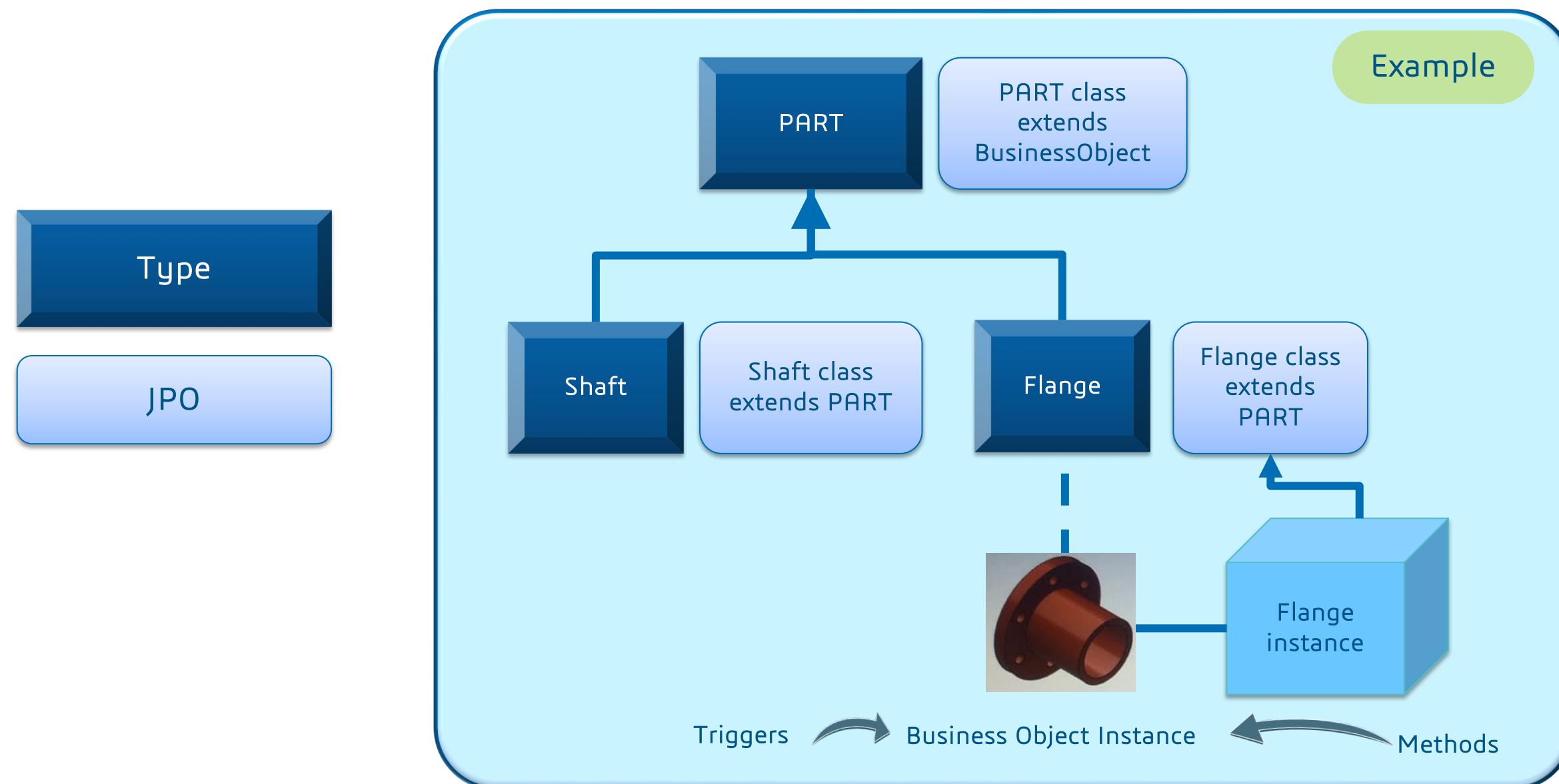


- ❑ A Java Program Object (JPO) is a Program Administrative Object that contains code written in the Java language. In general, we can say, anywhere in the platform, where a Program object is used, you could potentially use a JPO.
- ❑ JPOs provide the ability to run Java programs natively inside the kernel without creating a separate process and with a true object-oriented integration feel as opposed to working in MQL
- ❑ JPO contain Java classes and methods representing Business Logic
- ❑ JPOs are compiled. Because of this, they are much faster, as compared to a TCL program.
- ❑ JPOs run inside the same Java Virtual Machine (JVM) as **3DEXPERIENCE** platform
- ❑ JPOs offer a native object-oriented programming approach, unlike MQL/TCL Program objects
- ❑ JPOs may represent Business Object Types and can provide implementation of business logic that are specific to the Types. JPOs may be potentially used anywhere there is scope for a Program object. Common usages for JPOs are as follows:
 - Triggers
 - Methods
 - Attribute and relationship management

JPOs are stored in **ADMINISTRATION** vault



- JPO code should be considered server-side Java and should not be used to construct Java GUI components
- JPOs can be used as stand-alone Java classes or associated with any 3DEXPERIENCE platform Business Type.
- Naming convention of an extracted JPO is <NAME_IN_DATABASE>_mxJPO.java
 - Example: emxPart_mxJPO.java



Introduction to Triggers



- ❑ The web-apps automate business rules using Trigger programs that run when a particular trigger event occurs.
- ❑ The 3DEXPERIENCE platform Business Process Applications contain Trigger programs that automate business rules.
- ❑ The trigger programs can be developed using
 - TCL
 - Java – A *Program* administrative object with Java code is also referred to as a Java Program Object or JPO.

It is recommended to write *Trigger* programs using JPO, and not using TCL.

- ❑ There are three kinds of trigger programs:
 - Check Trigger
 - Override Trigger
 - Action Trigger
- ❑ Any event could be associated with multiple *Trigger* programs. This allows for defining multiple separate and independent business logic for any database transaction.
- ❑ You can also run multiple *Trigger* programs of the *same kind* for an event (for example, run 3 separate *Check Trigger* programs when an object is created). It also makes it easier to enter input for the trigger program arguments.



- ❑ Business Objects support a lot of events, which can be configured on it's Type, Policy States, Attributes and Relationships.
- ❑ Some frequently encountered **events** are:
 - Revision
 - Connect
 - Create
 - Delete
 - Disconnect
 - Lock
 - Modify
 - ModifyAttribute
 - Promote
 - Demote

MQL<18>`help type;`

```
where EVENT_TYPE is:  
| addinterface  
| changevault  
| changename  
| changeowner  
| changepolicy  
| changetype  
| checkin  
| checkout  
| connect  
| copy  
| create  
| grant  
| delete  
| disconnect  
| lock  
| majorrevision  
| minorrevision  
| implicitcopy  
| implicitdelete  
| implicitmajorrevision
```

Please refer to User Assistance documentation ([link](#)) to get the full list of events available for each Objects (Attribute, Type, etc.)



Please note that the list of events on Admin objects is a standard list. It implies that you cannot add your own events to this list, and hence it is not customizable.



□ Check Trigger

- The Check Trigger executes before any of the normal system code that carries out the recognized event has a chance to execute.
- A Check Trigger program can block the event from actually occurring, aborting the event transaction, which means that no other trigger programs are executed. This is a **hard block**.
- A system error message indicating the blocked event is displayed in case condition check fails.

□ Override Trigger

- The Override Trigger executes after the Check Trigger (if not blocked) and before the trigger event has a chance to execute.
- It can replace the normal event processing, with an alternate event, in effect blocking the event, which is also referred to as a “**soft block**” on the event.
- In this case, only the user-defined error message will be displayed.

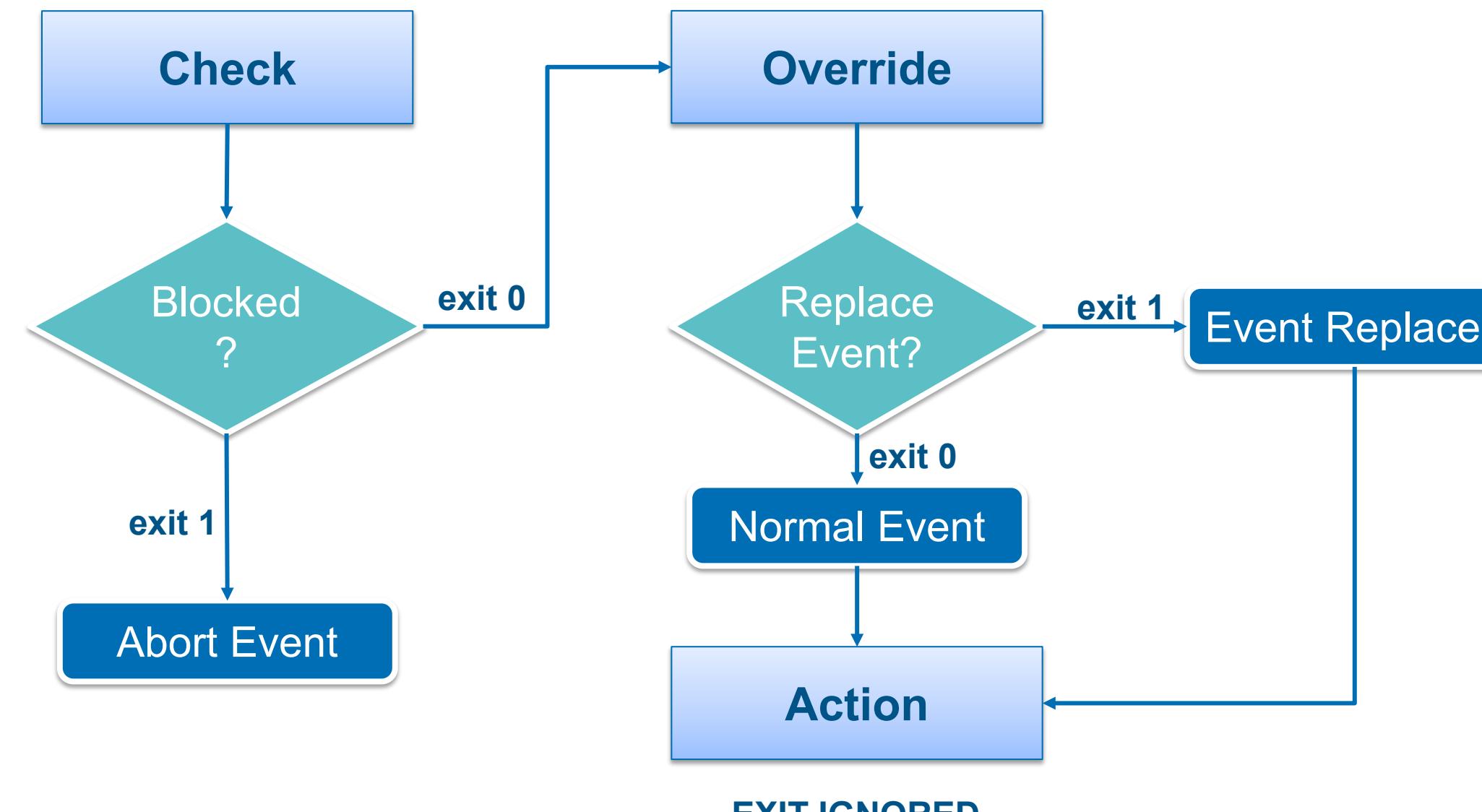
□ Action Trigger

- The Action Trigger executes after the event transaction is committed, irrespective of whether the normal event took place or the override program replaced the event.
- An Action Trigger is only executed if the previous event is successfully committed.

Trigger Timing Flow and Exit Codes



- Check Triggers **block** an event by returning a non-zero value to the 3DEXPERIENCE platform (**Hard-block**).
- Override Triggers **replace** the current event by returning a non-zero value to 3DEXPERIENCE platform (**Soft-block**).
- Action Triggers return value is **ignored**



TCL trigger programs return a value to 3DEXPERIENCE platform using the **exit** command



Here are the key takeaways from this lesson:

- The Web App *User Interface* is created using standard components called **Configurable Components**
 - Menus
 - Command
 - Web Forms
 - Tables
 - Portals
 - Channels
- Web App business logic can be customized using
 - JPOs
 - Triggers



Congratulations!
You have completed this training.

Dassault Systèmes. All rights reserved. 3DEXPERIENCE, the Compass icon, the 3DS logo, CATIA, BIOVIA, GEOVIA, SOLIDWORKS, 3DVIA, ENOVIA, NETVIBES, MEDIDATA, CENTRIC PLM, 3DEXCITE, SIMULIA, DELMIA and IFWE are commercial trademarks or registered trademarks of Dassault Systèmes, a French "société européenne" (Versailles Commercial Register # B 322 306 440), or its subsidiaries in the United States and/or other countries. All other trademarks are owned by their respective owners. Use of any Dassault Systèmes or its subsidiaries trademarks is subject to their express written approval.

