What’s New

Highlighting of changes and what this means for your project.

Knowledge session

Blueriq

Topics

During this What's **New** we'll be discussing the following topics.

Introduction

- History
- Categories

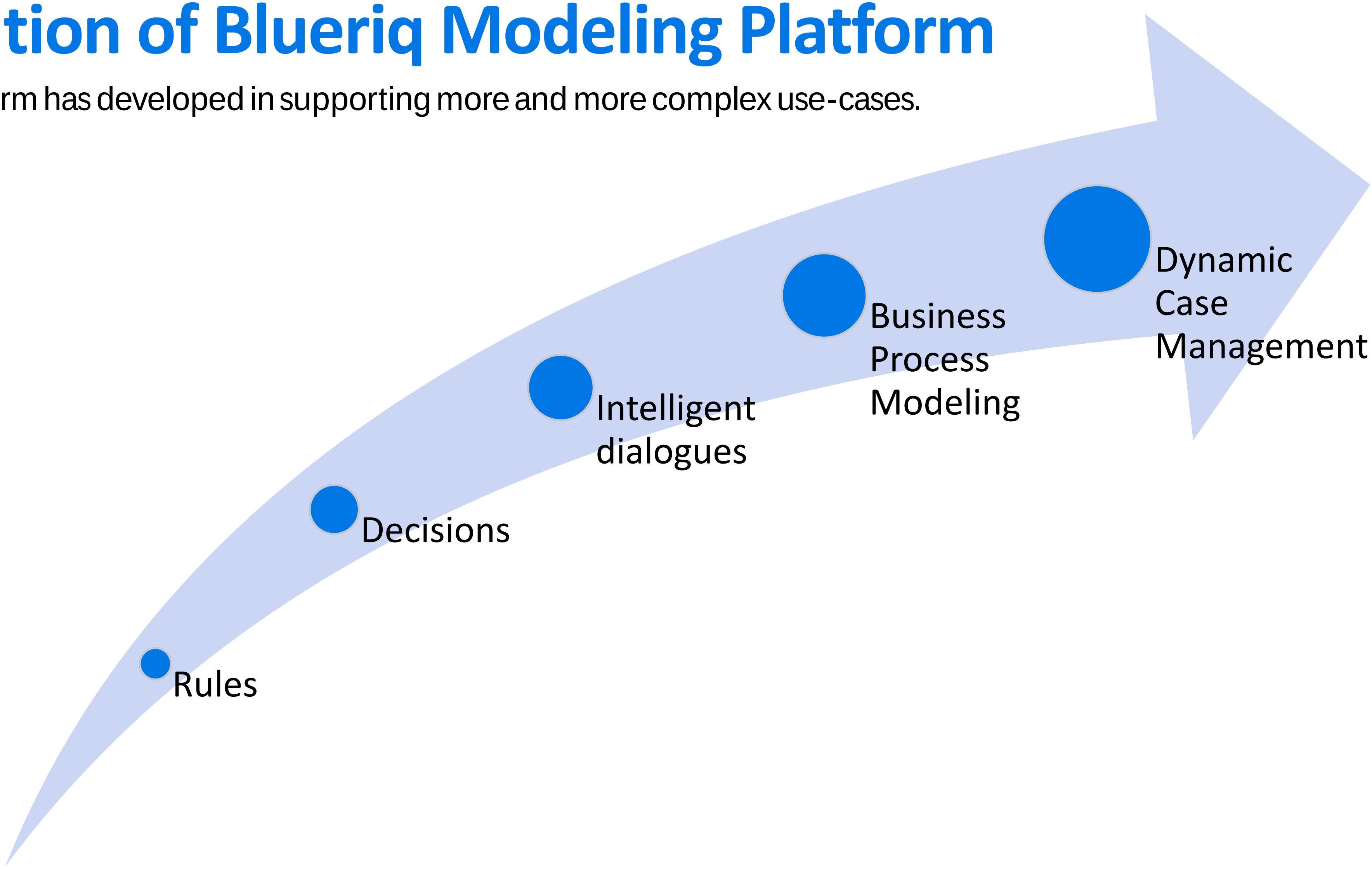
Releases

- Blueriq 10
- Blueriq 11
- Blueriq 12
- Blueriq 13

Improvement tips

Evolution of Blueriq Modeling Platform

Our platform has developed in supporting more and more complex use-cases.



Categories

Though not official sorted into categories by the product development team, we can distinguish certain traits of these changes.

Keeping up with technology

Ensuring we keep up to date with technological developments.

Time to market

Increasing the speed of development and delivery.

Quality

Having means to determine what the state of the model and resulting application is.

Efficiency

Needing less time to do the activities related to the Blueriq products.

Usability

Ensuring a user-friendly interaction with Blueriq products.

Business Engineering

Offering an ever-growing toolset to professionals in order to solve business problems by creating a knowledge model that can be executed.

Integrating Blueriq

Having means to integrate Blueriq into a landscape.

Separation of concerns

Ensuring that functionality, responsibility and implementation of components are separated.

Easy updates

Lowering the bar to adopt new versions and updates.

For each version we´ll be discussing the following topics.

Release version

Studio

- New features
- Enhancements
- Usability
- Unit testing

Runtime

New products

Front-end

Removed

Blueriq 12

Studio

- New features
- Enhancements
- Usability
- Unit testing

Runtime

New products

Front-end

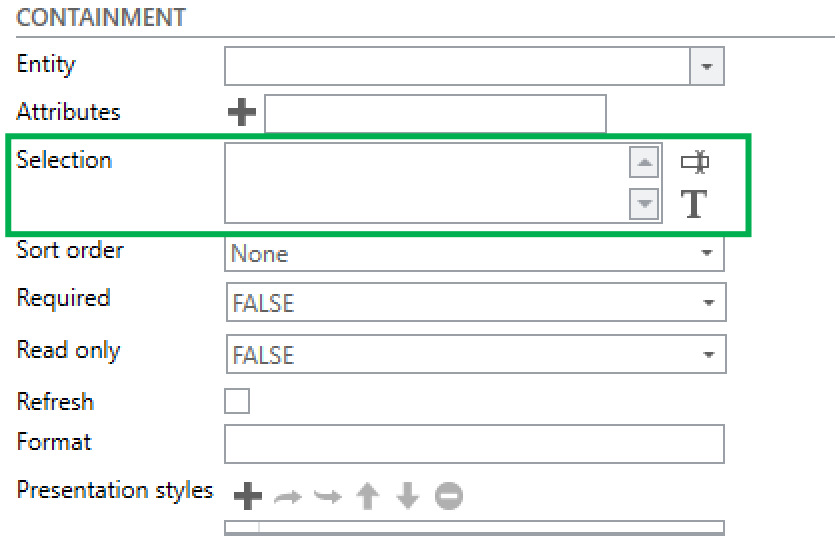
Removed

InstanceLinker selection expression

No longer plain text.

Data types

No longer just string type. Especially handy when returning justifications from BAAS instances.



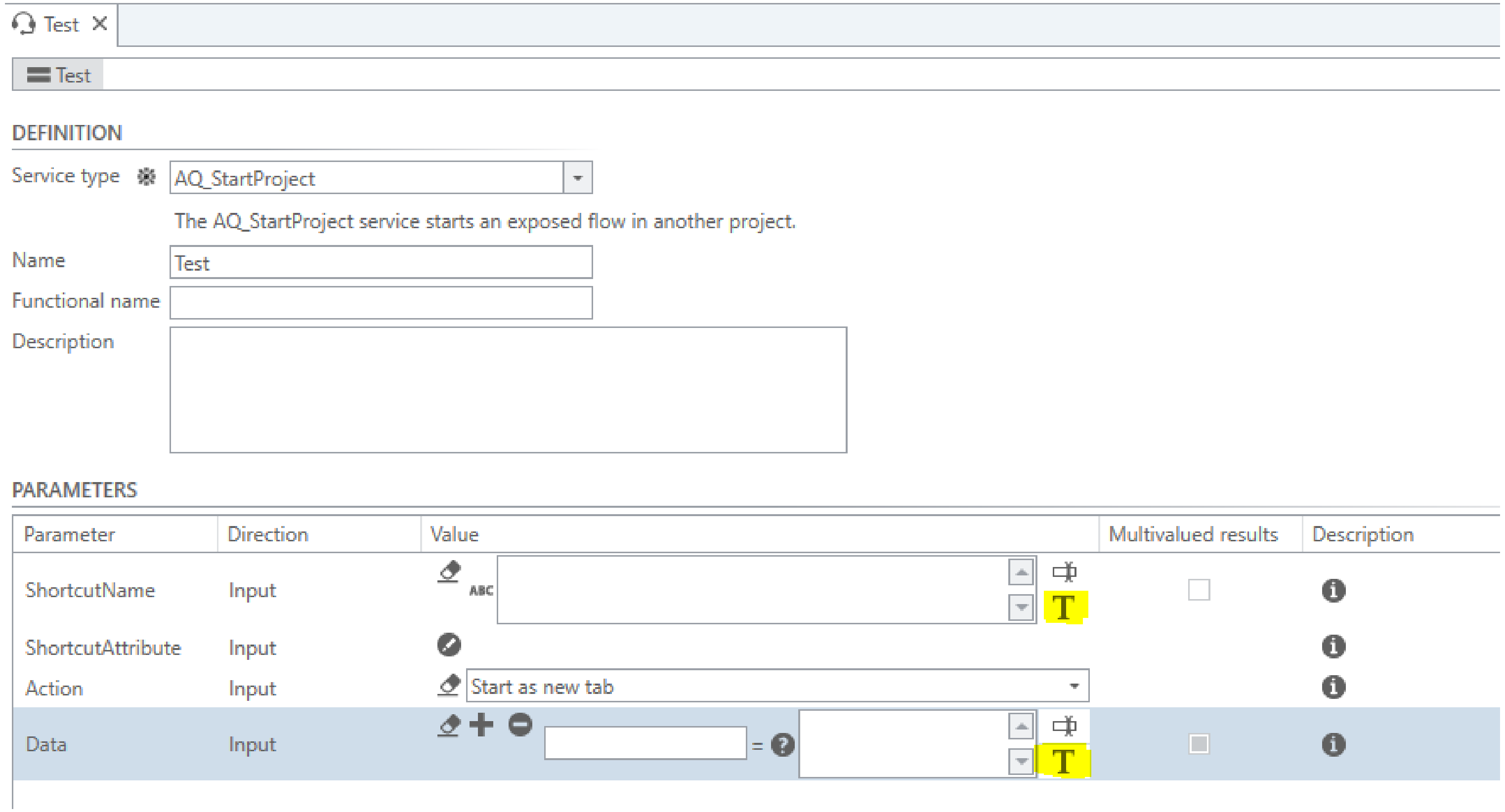
CONTAINMENT

Entity	
Attributes	
Selection	
Sort order	None
Required	FALSE
Read only	FALSE
Refresh	□
Format	
Presentation styles	

12.0

Servicecall expression toggle

When configuring expressions in a service call before, you had no option to toggle to plain text.



Test

Test

DEFINITION

Service type: AQ_StartProject

The AQ_StartProject service starts an exposed flow in another project.

Name: Test

Functional name:

Description:

PARAMETERS

Parameter	Direction	Value	Multivalued results	Description
ShortcutName	Input	ABC T	□	ℹ
ShortcutAttribute	Input			ℹ
Action	Input	Start as new tab		ℹ
Data	Input	= ? T	□	ℹ

12.0

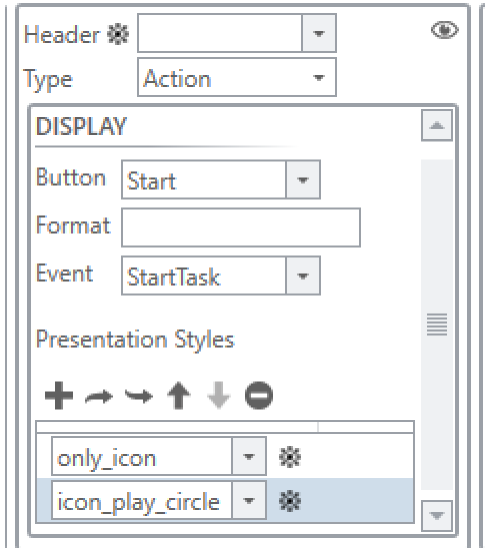
blueriq

Presentation styles in more components

When configuring expressions in a service call before, you had no option to toggle to plain text.

Columns

AQ_WorkList & CaseList



Header
Type: Action

DISPLAY

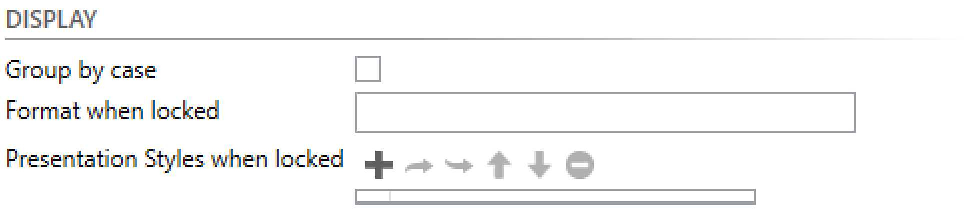
Button: Start
Format:
Event: StartTask

Presentation Styles

only_icon
icon_play_circle

Locked case

AQ_WorkList

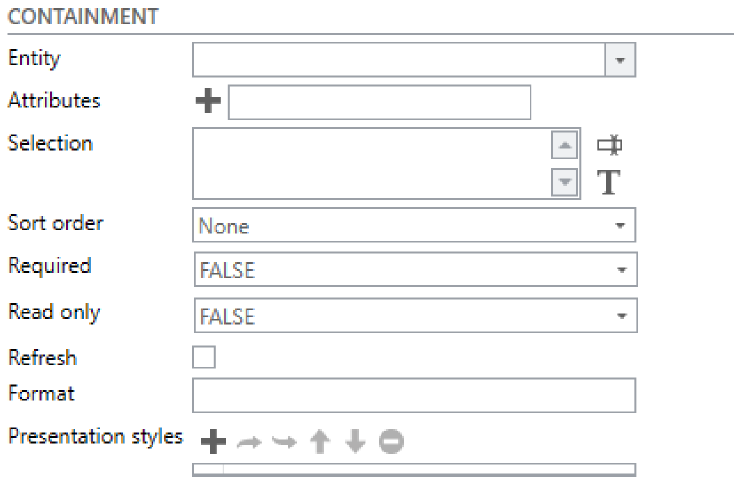


DISPLAY

Group by case □
Format when locked:
Presentation Styles when locked

Instance linking

AQ_InstanceLinker



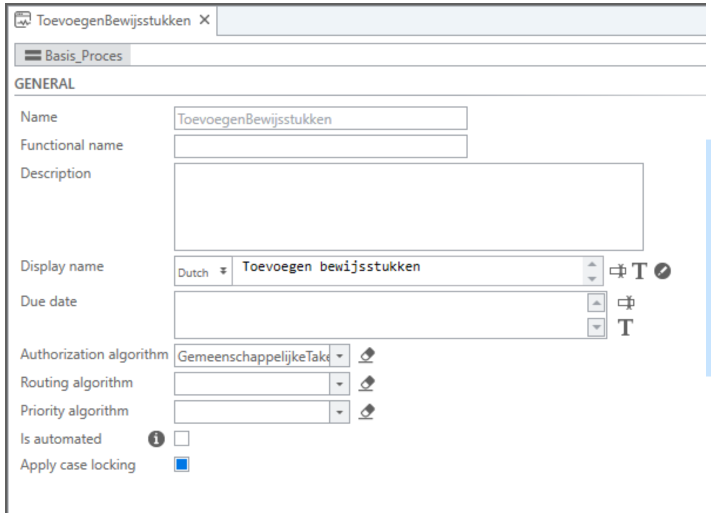
CONTAINMENT

Entity:
Attributes:
Selection:
Sort order: None
Required: FALSE
Read only: FALSE
Refresh □
Format:
Presentation styles

12.0

Renaming tasks

When you rename a task and execute a task with the "old" name, the persistency ID is used to identify the new task name and execute the corresponding task.

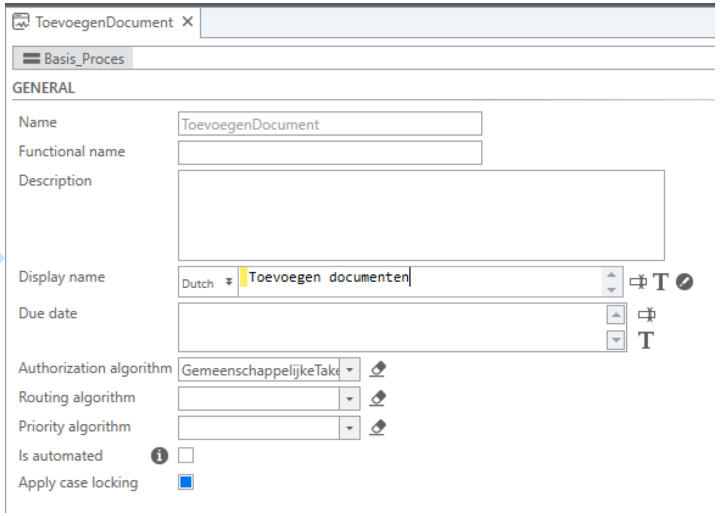


ToevoegenBewijsstukken ✕

Basis_Proces

GENERAL

Name	ToevoegenBewijsstukken
Functional name	
Description	
Display name	Dutch – Toevoegen bewijsstukken
Due date	
Authorization algorithm	GemeenschappelijkeTake
Routing algorithm	
Priority algorithm	
Is automated	□
Apply case locking	☑



ToevoegenDocument ✕

Basis_Proces

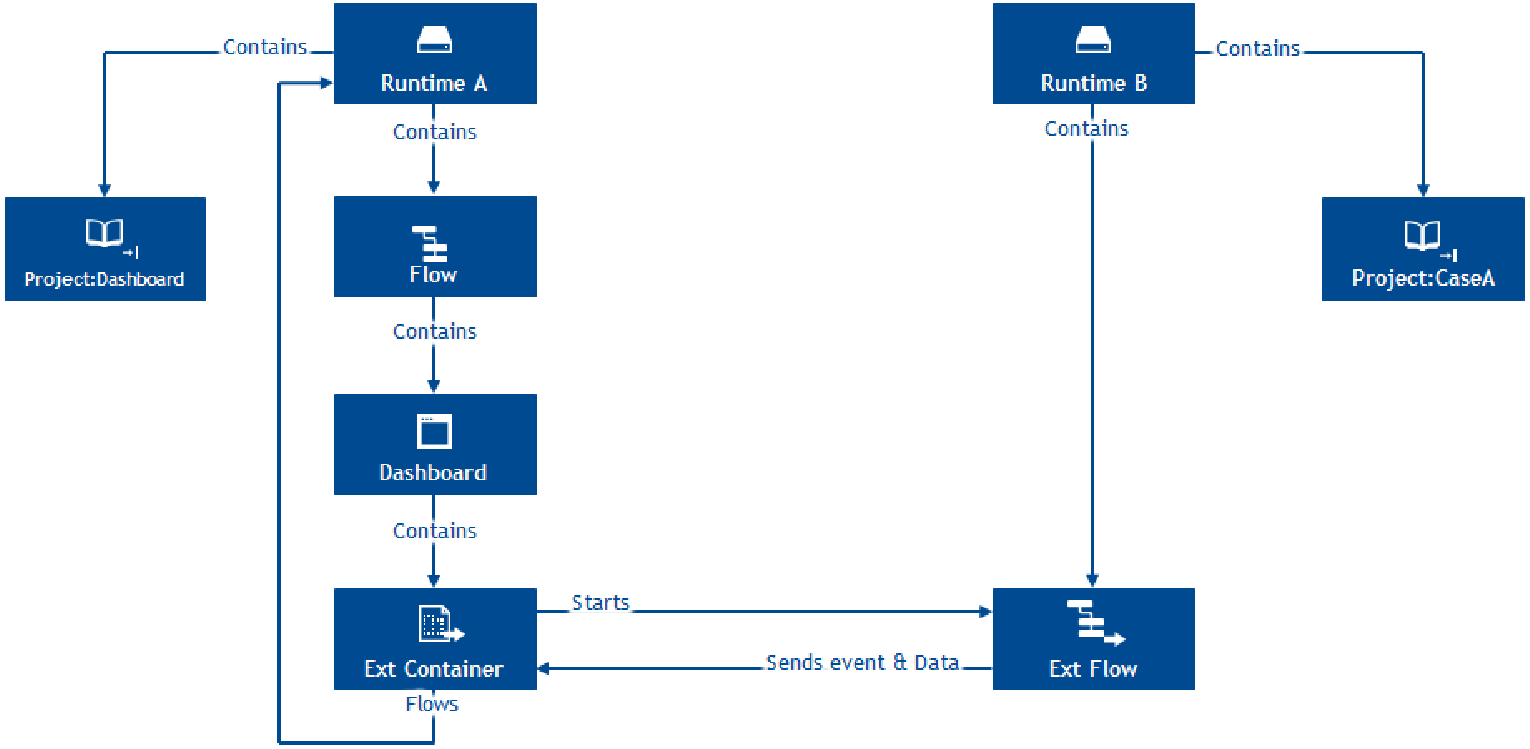
GENERAL

Name	ToevoegenDocument
Functional name	
Description	
Display name	Dutch – Toevoegen documenten
Due date	
Authorization algorithm	GemeenschappelijkeTake
Routing algorithm	
Priority algorithm	
Is automated	□
Apply case locking	☑

12.0

External flows

With this feature it becomes possible to execute a model that is present on a different Runtime instance. A **contract module is needed** to enable communication between two Runtimes in this manner.



12.5

External flows

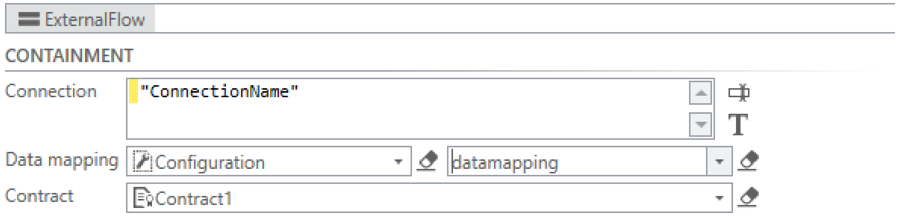
With this feature it becomes possible to execute a model that is present on a different Runtime instance. A **contract module is needed** to enable communication between two Runtimes in this manner.

Pro tip!

Ensure that the **host** and **target** Runtime have the **same major** and **minor** versions

AQ_ExternalFlow

Both a new type of container, **AQ_ExternalFlow**, and flow, **External Flow**, are needed.



ExternalFlow

CONTAINMENT

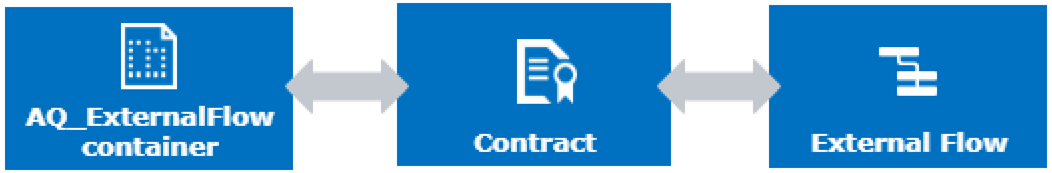
Connection: "ConnectionName"

Data mapping: Configuration | datamapping

Contract: Contract1

Creating a contract

A contract module needs to be created and implemented in both the host and target application.



12.5

Test paths

Introducing the possibility to create a path for testing across different versions. This eliminates the need to create unnecessary versions for projects in which nothing will be changed.

Setting the default

In order to use this functionality you'll need to define the **default version** to be used.

Creating a test path

Test paths can be created through the **Runtime Dashboard**. Select a branch and give the path a name.

Running a test path

Test paths are listed amongst the **shortcuts** and can be executed as one as well.

If available the version corresponding to the test path will be used.

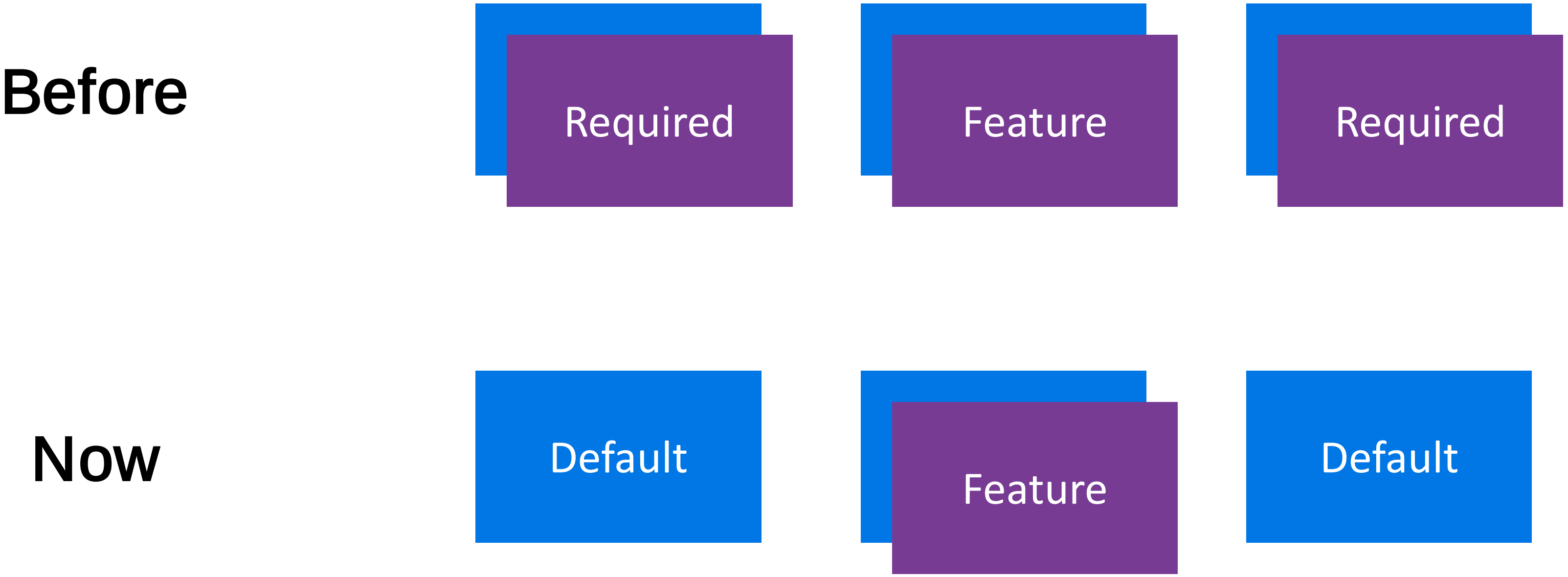
Learn more!

Our community has more information about this topic.

12.9

Test paths

It is no longer needed to create additional branches when trying to test or validate functionality with test paths. You can now resort to a **default** branch.



Before

Required

Feature

Required

Now

Default

Feature

Default

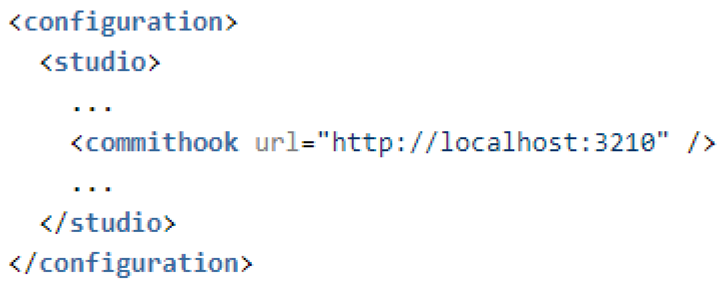
12.9

Commit hook for CI/CD

This feature allows you to trigger build pipelines from within Blueriq Studio.

Studio service config

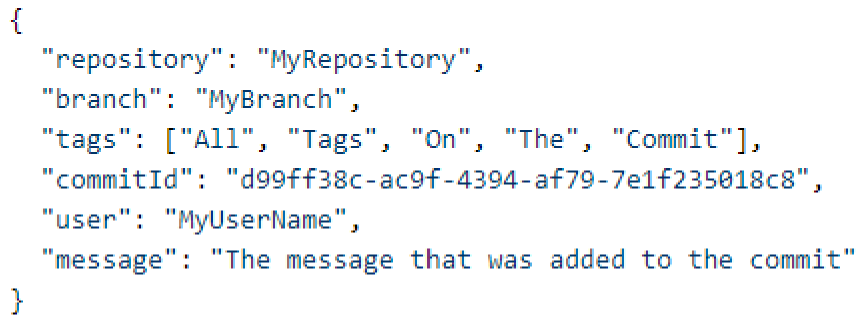
Configure an endpoint to receive a HTTP request whenever a commit is performed.



```
<configuration>
  <studio>
    ...
    <commithook url="http://localhost:3210" />
    ...
  </studio>
</configuration>
```

Resulting HTTP request

Receive metadata at the endpoint that can be used further down the pipeline.



```
{
  "repository": "MyRepository",
  "branch": "MyBranch",
  "tags": ["All", "Tags", "On", "The", "Commit"],
  "commitId": "d99ff38c-ac9f-4394-af79-7e1f235018c8",
  "user": "MyUserName",
  "message": "The message that was added to the commit"
}
```

12.7

IFUNKNOWN function

In case of readability of the model, the IFUNKNOWN function is introduced. This offers a new alternative of writing functions that validate if something is unknown.

Simplify long expressions

Especially **bigger expressions** can become hard to oversee when they contain one or more **=?** statements.

Using the **IFUNKNOWN** function increases readability of these functions, while preserving the result.

```
expression IFUNKNOWN value
```

12.4 ✋

Splitting up the Studio

The Studio has been needing an update for a while now. The first steps in doing so, is looking at everything that is done underneath the hood.

Old

Studio Service

Studio

Changing up the core

The structure underneath the studio has been changed up.

Why

- Evolving from modelling application to a modelling platform
- Multiple visualizations of the same model
  - XBAP Studio
  - Model Mapper
- SDF reached it limits
- Current way of versioning is embedded deeply into Studio causing complexity and scalability constraints

Version management system

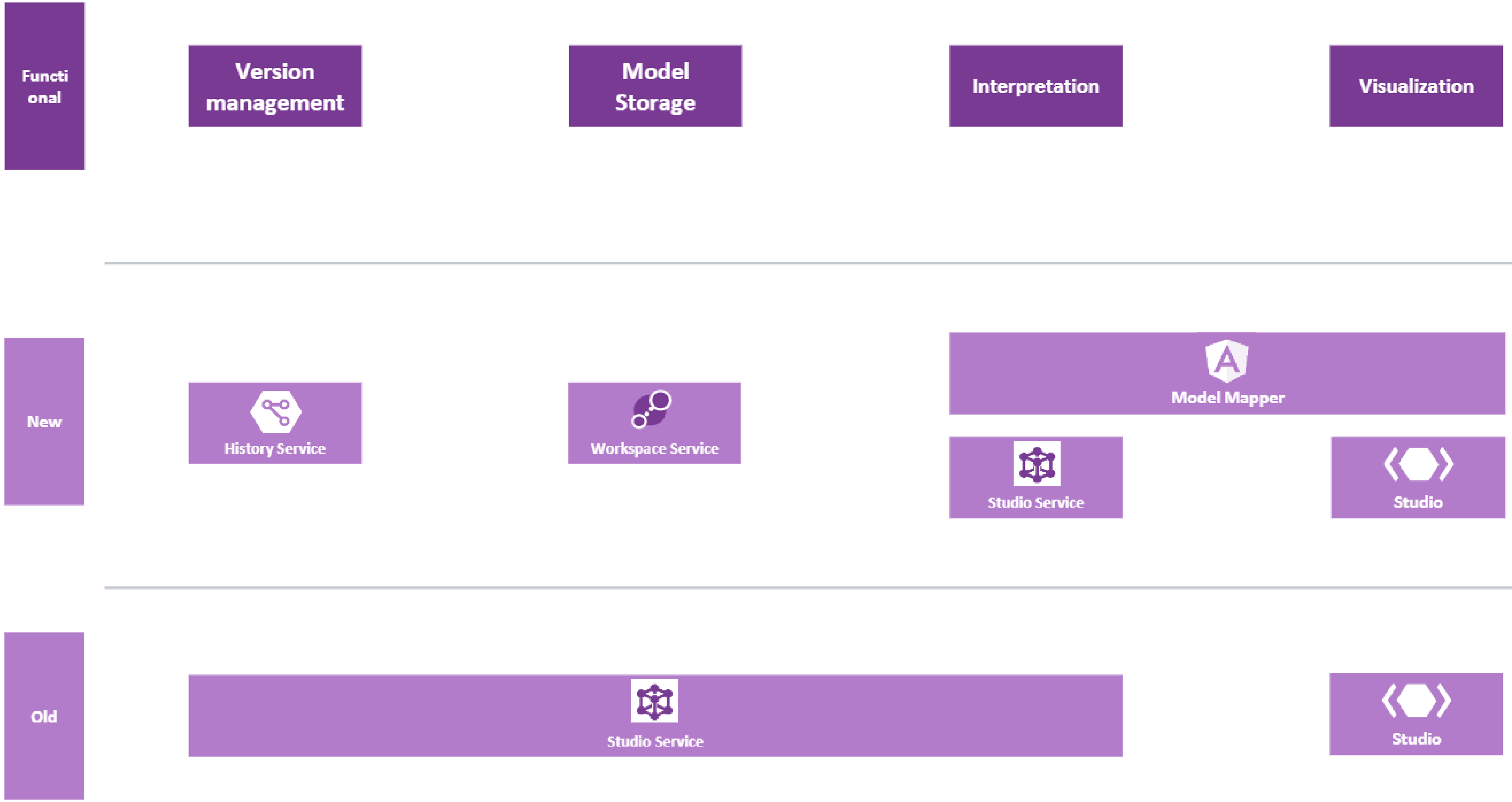
Model data storage

Which steps did we take?

1. Use of **PostgreSQL & Git** instead of SQL CE
2. Introducing an **opt-in mechanism** for both features
3. SDF **migration tool** is made available

12.9

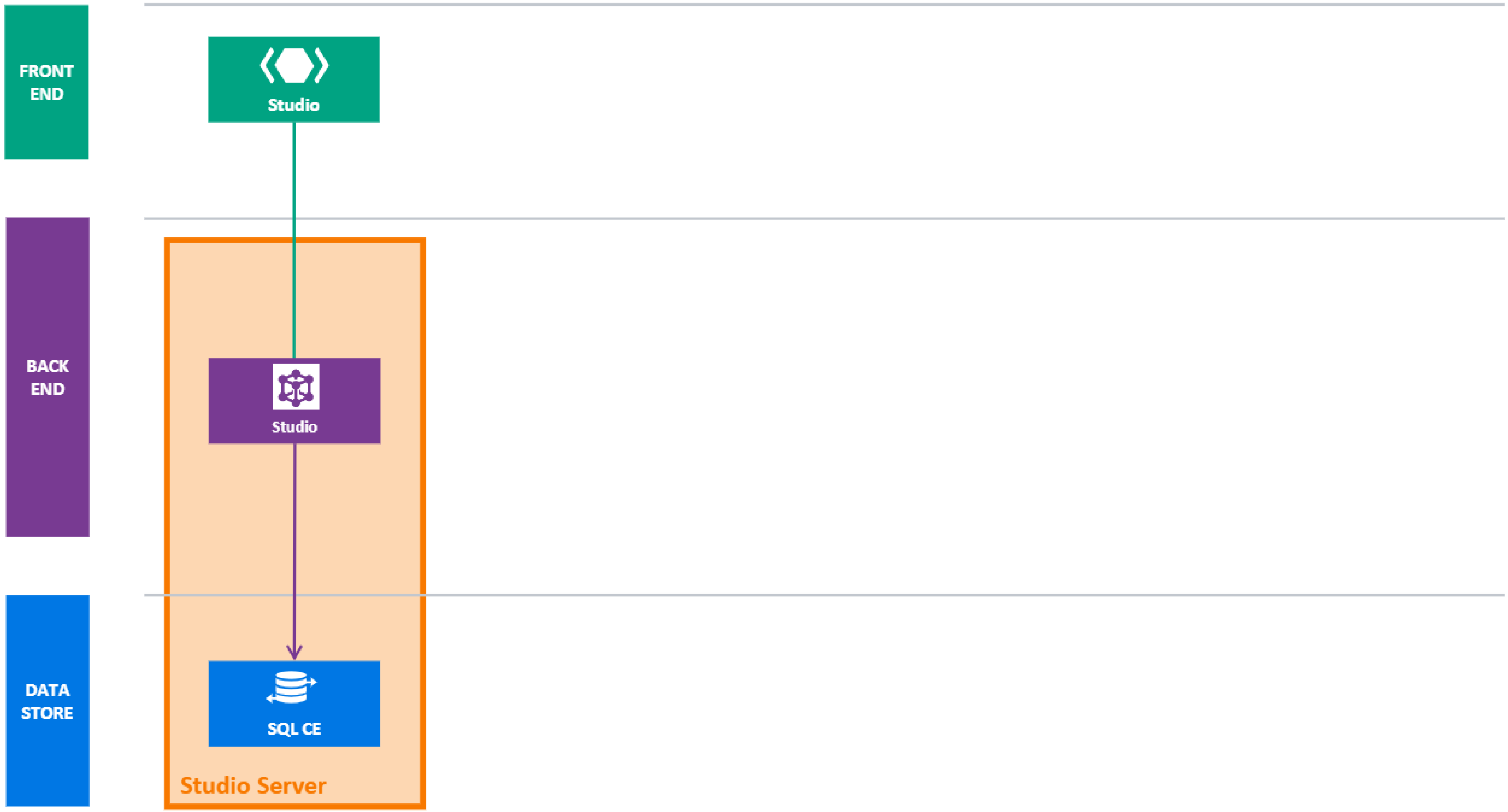
A comparison



Functional	Version management	Model Storage	Interpretation	Visualization
New	History Service	Workspace Service	Model Mapper Studio Service	Studio
Old	Studio Service			Studio

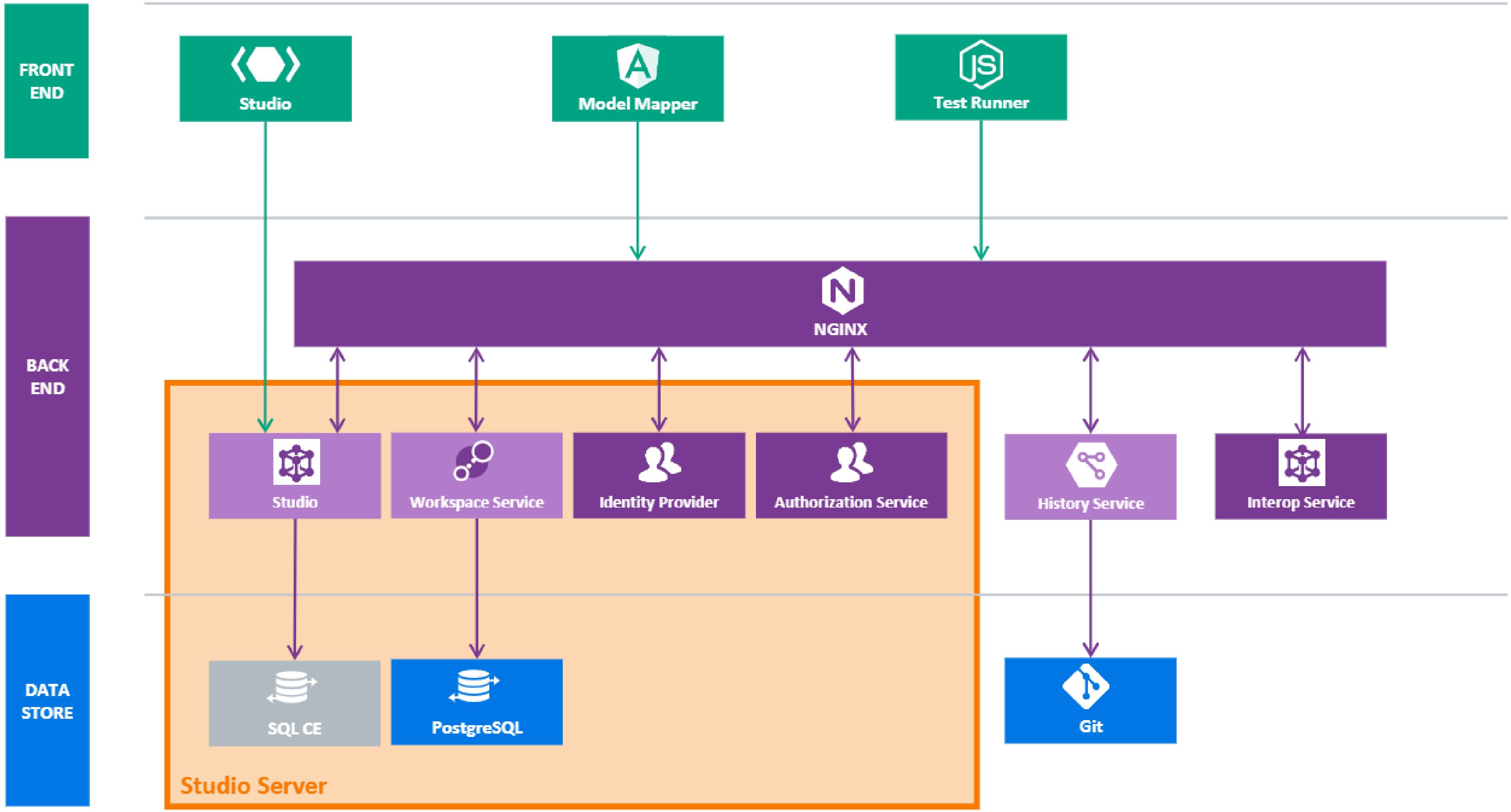
12.9

Current composition



12.9

New composition



Blueriq 12

Studio

- New features
- Enhancements
- Usability
- Unit testing

Runtime

New products

Front-end

Removed

Encore

As you might have heard we're developing the next generation design environment for Blueriq models called; **Encore**.

1 on 1 with a little extra

Encore will have the **same functionality** as the current Studio.

In addition separate products, like the **Model Mapper**, will be **integrated** as part of Encore.

XBAP no more!

Encore will be a web application **no longer** based on **XBAP** technology.

Say goodbye to Internet Explorer (finally)

New styling

For Encore a **new design system** will be used to give all the components a coherent look and feel.

Username
Password
Login
Forgot password?