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

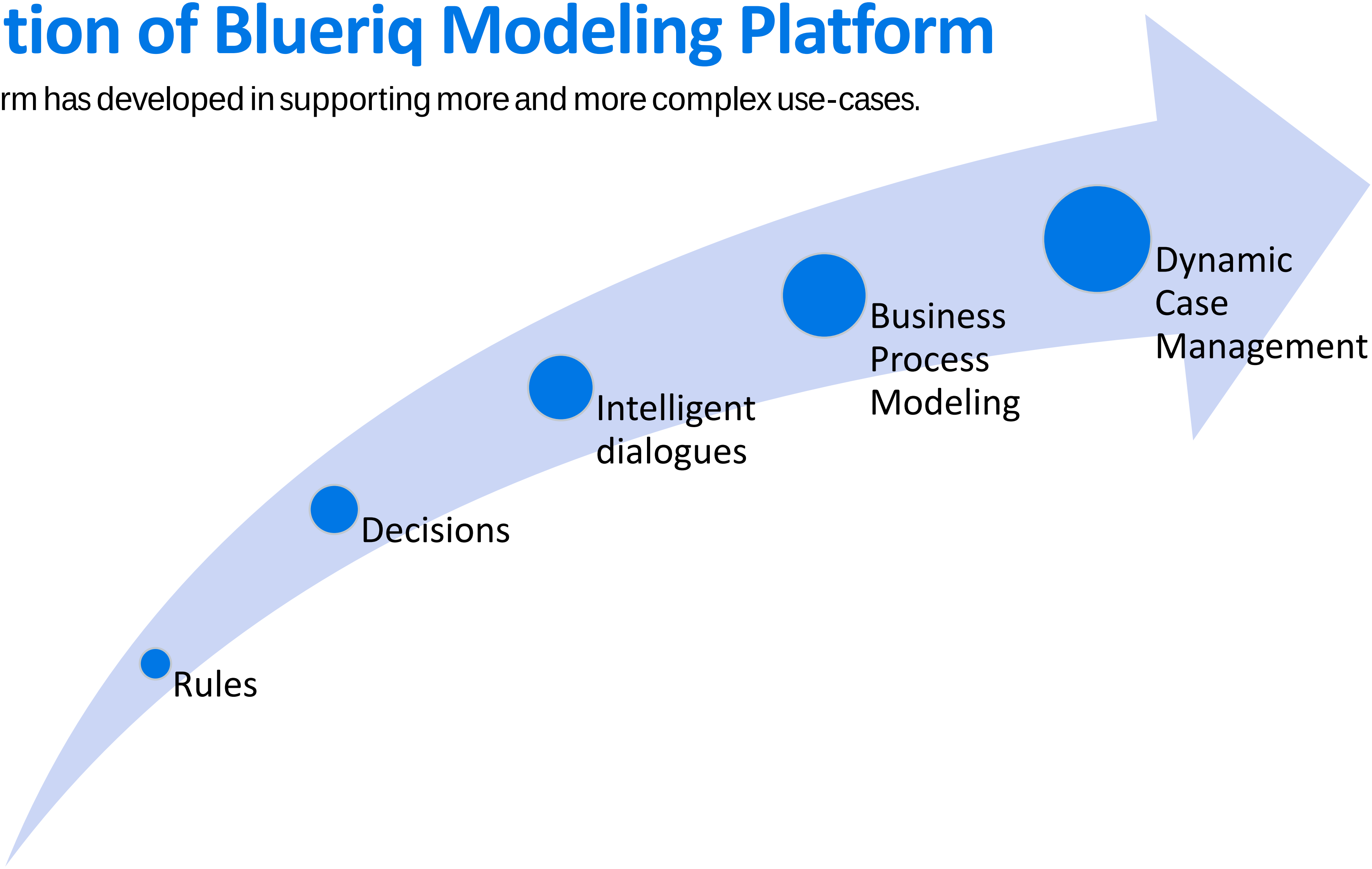
Releases

- 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.



Rules

Decisions

Intelligent dialogues

Business Process Modeling

Dynamic Case Management

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 13

Studio

- New features
- Enhancements
- Usability
- Unit testing

Runtime

New products

Front-end

Removed

Justification Tree 2.0

The old justification tree but with more integration options when using the functionality when returning values to

Data types

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

Function or technical name

It’s up to you. Let Blueriq return the functional name of the attribute or the traditional name.

More assets!

Combine multiple asset types instead of a singular type.

13.12

Oauth2 for the REST service client

Security is an increasingly hot topic. You’ll no longer need to struggle with incorporating the Oauth2 authentication scheme.

(Con-)Figure it out yourself!

Combine the existing connectivity capabilities with the **Oauth2** functionality.

```
blueriq.connection.<connectionName>.http.authentication = oauth2
blueriq.connection.<connectionName>.http.oauth2-token-endpoint
blueriq.connection.<connectionName>.http.oauth2-client-id
blueriq.connection.<connectionName>.http.oauth2-client-secret
blueriq.connection.<connectionName>.http.oauth2-grant-type
blueriq.connection.<connectionName>.http.oauth2-scope
```

13.12

Blueriq 13

Studio

- New features
- Enhancements
- Usability
- Unit testing

Runtime

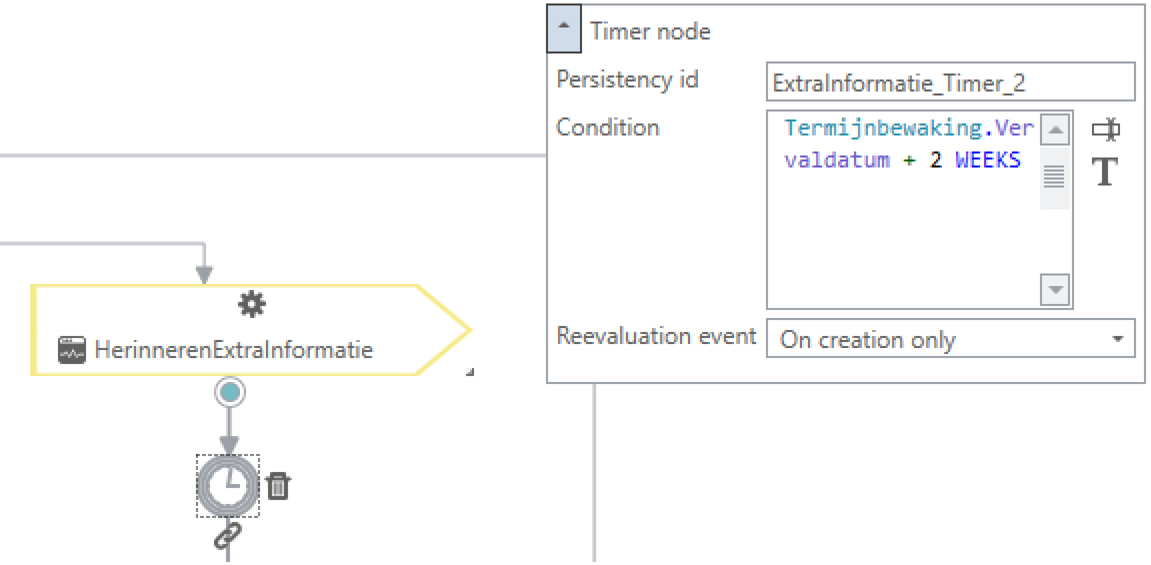
New products

Front-end

Removed

Timer re-evaluation

Recalculate timer nodes through the “**reevalution**” reevaluation event. No longer the need to create complex workarounds to achieve the same behaviour.



Timer node

Persistency id: ExtraInformatie_Timer_2

Condition:
```
Termijnbewaking.Ver
valdatum + 2 WEEKS
```

Reevaluation event: On creation only

On creation only

The condition will be evaluated **only once**, when the timer node is created.

On domain change

The condition will be evaluated **each time the result of the condition changes**, during a domain update.

13.12

Blueriq 13

Studio

- New features
- Enhancements
- Usability
- Unit testing

Runtime

New products

Front-end

Removed

OpenAPI is here

Having to work wit a lot of other services having to consume your webservice? Blueriq has had a Swagger (OAS2) implementation for a while now. It has been migrated to OpenAPI (OAS3), but you can already use Swagger in earlier versions.

Swagger OAS2 is already available

The Runtime will load the models directly after it has started.

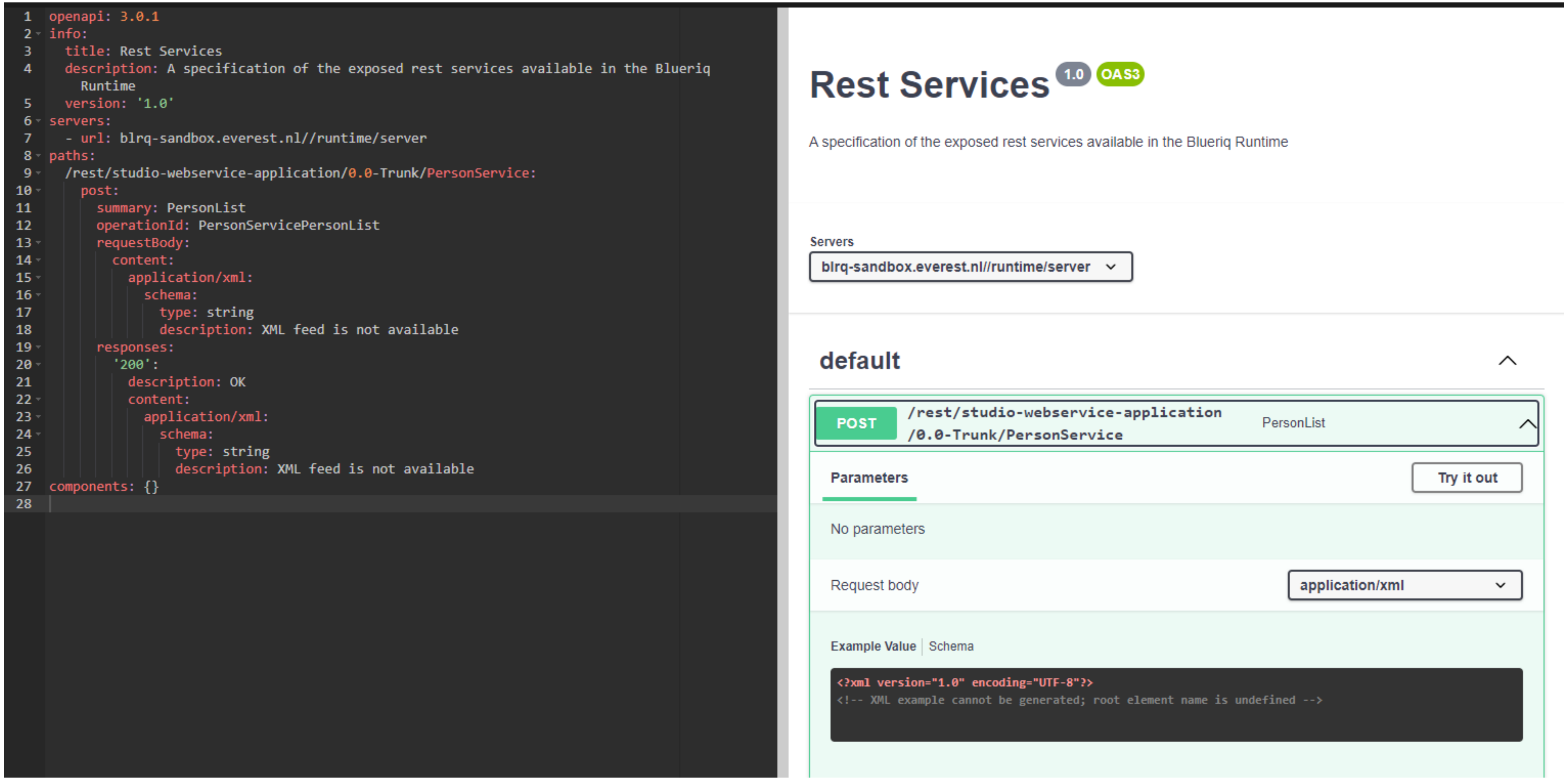
Interface for developers

You can change the details of the preloading in the settings.

```
{runtime-url}/server/rest/docs?project={project}&version={version}
```

13.0

OpenAPI (OAS3) is here



```yaml
openapi: 3.0.1
info:
  title: Rest Services
  description: A specification of the exposed rest services available in the Blueriq Runtime
  version: '1.0'
servers:
  - url: blrq-sandbox.everest.nl//runtime/server
paths:
  /rest/studio-webservice-application/0.0-Trunk/PersonService:
    post:
      summary: PersonList
      operationId: PersonServicePersonList
      requestBody:
        content:
          application/xml:
            schema:
              type: string
              description: XML feed is not available
      responses:
        '200':
          description: OK
          content:
            application/xml:
              schema:
                type: string
                description: XML feed is not available
components: {}
```

Rest Services 1.0 OAS3

A specification of the exposed rest services available in the Blueriq Runtime

Servers
blrq-sandbox.everest.nl//runtime/server

default

POST /rest/studio-webservice-application/0.0-Trunk/PersonService PersonList

Parameters | Try it out

No parameters

Request body: application/xml

Example Value | Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- XML example cannot be generated; root element name is undefined -->
```

13.0

Preloading the model

The first time you access a model through the Runtime after start-up Blueriq used to load the model. Resulting in increased initial loading times that could impact the experience of the end users.

Modify behaviour

You can change the details of the preloading in the settings.

Load at startup

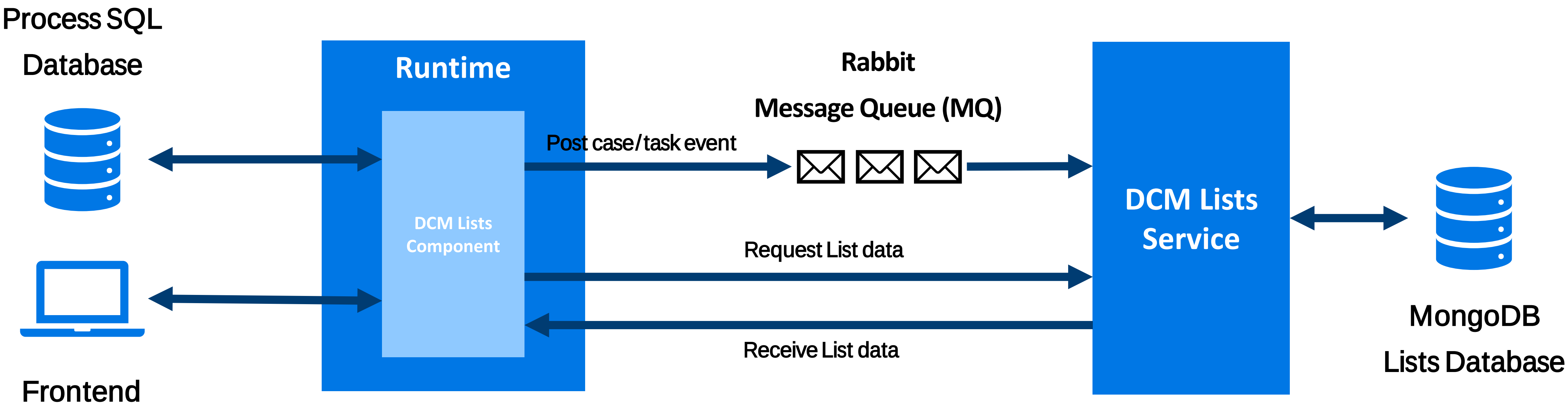
The Runtime will load the models directly after it has started.

```
blueriq.modelpreload.enabled           = TRUE/FALSE
blueriq.modelpreload.before-requests   = TRUE/FALSE
blueriq.modelpreload.models            = {modelnaam}
```

13.8

DCM Lists Service

Overhauling the way case and task events are published, stored and retrieved. This enables more decoupled handling of events.



13.0