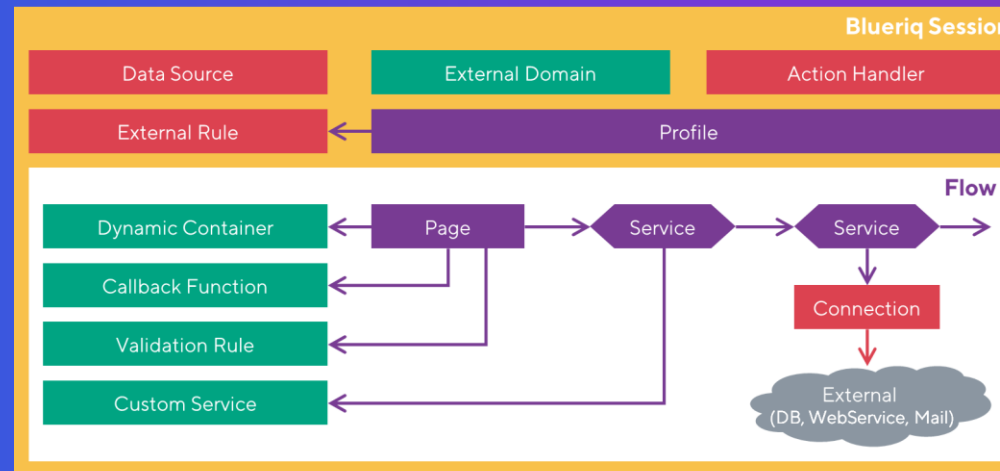


Technical Foundation

Technical engineers customize and configure Blueriq to extend it with custom functionality



blueriq

Program

Day 1

Introduction

- Blueriq Runtime process
- Blueriq Components
- Blueriq Architecture

Development environment

- Create a Blueriq project
- Create plugins
- Build plugins
- Deploy and debug plugins

Services

- Profile
- Error handling
- Multilingual applications

Day 2

Services continued

- Introduction to unit testing

Containers

- Composer
- Callback Functions

Day 3

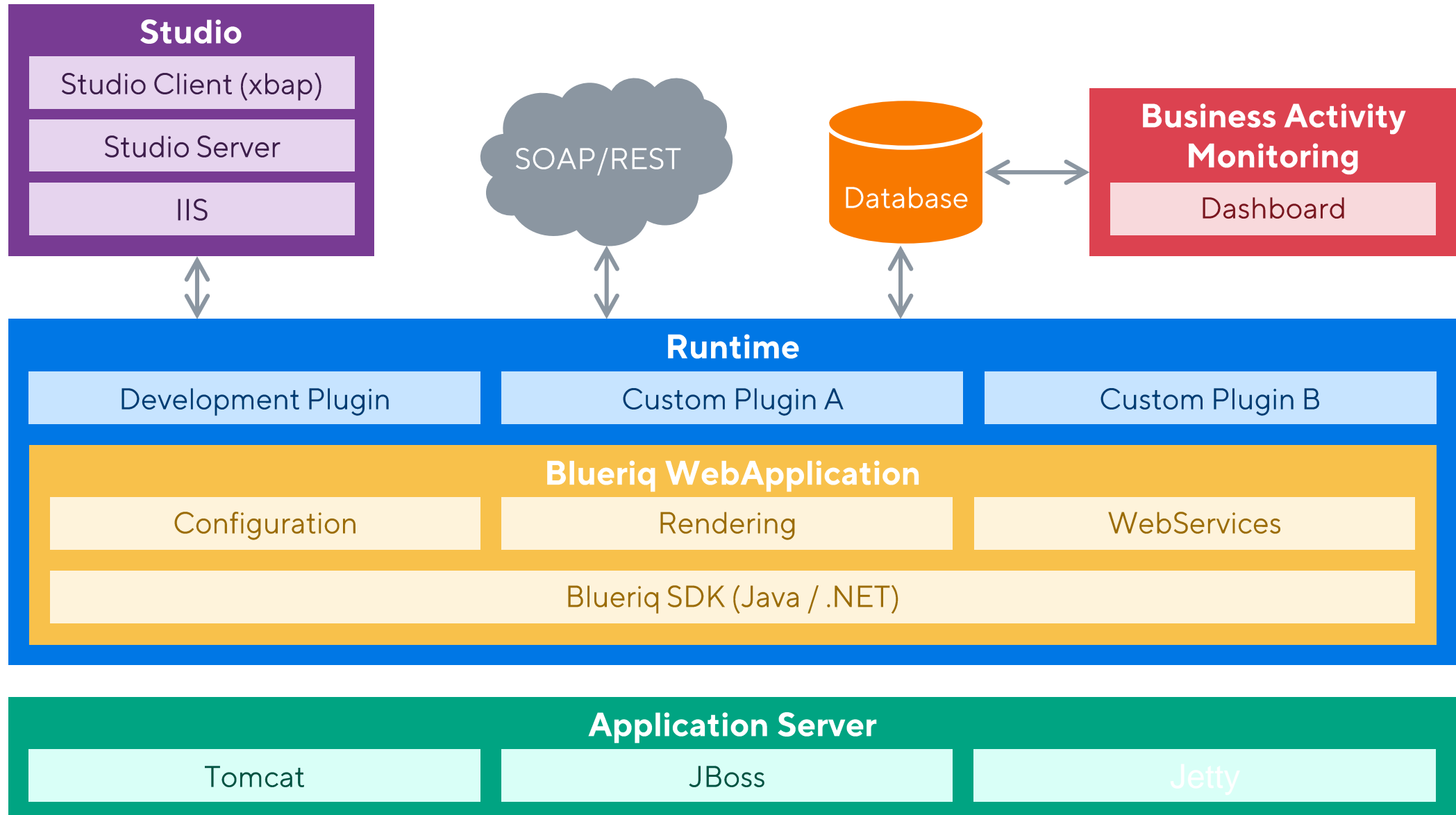
Containers continued

- Introduction to unit testing

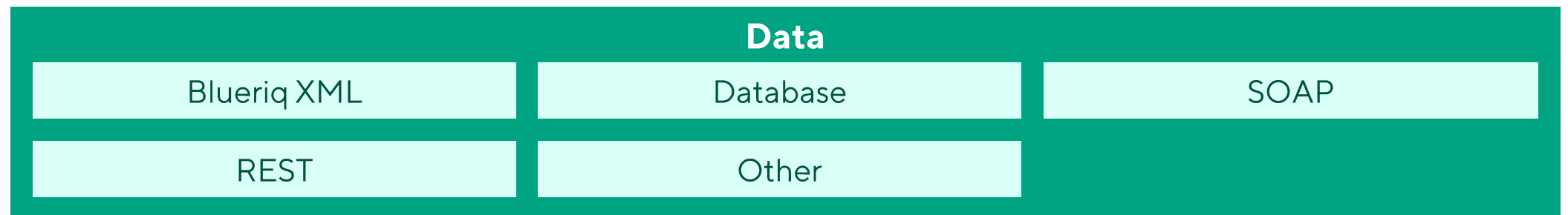
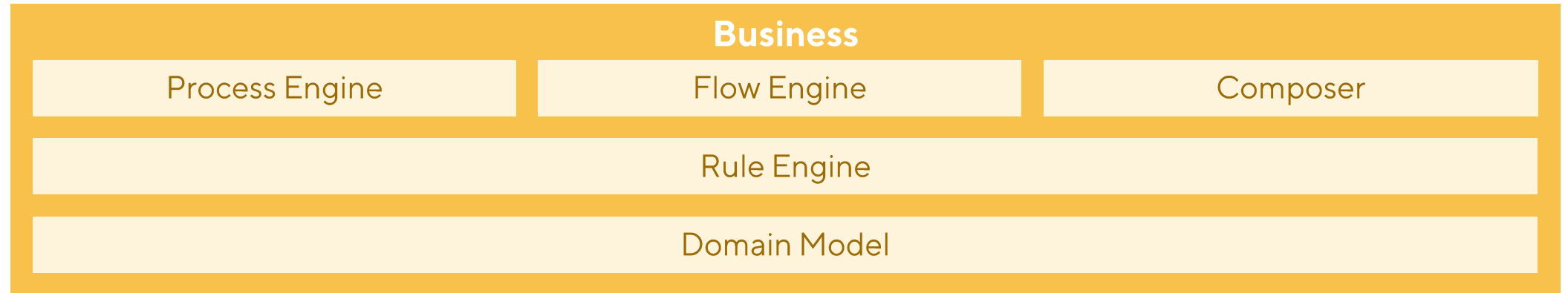
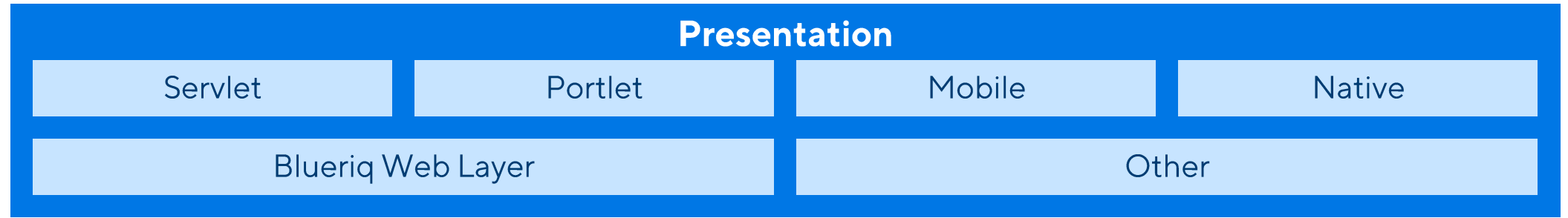
External domains

Validation rules

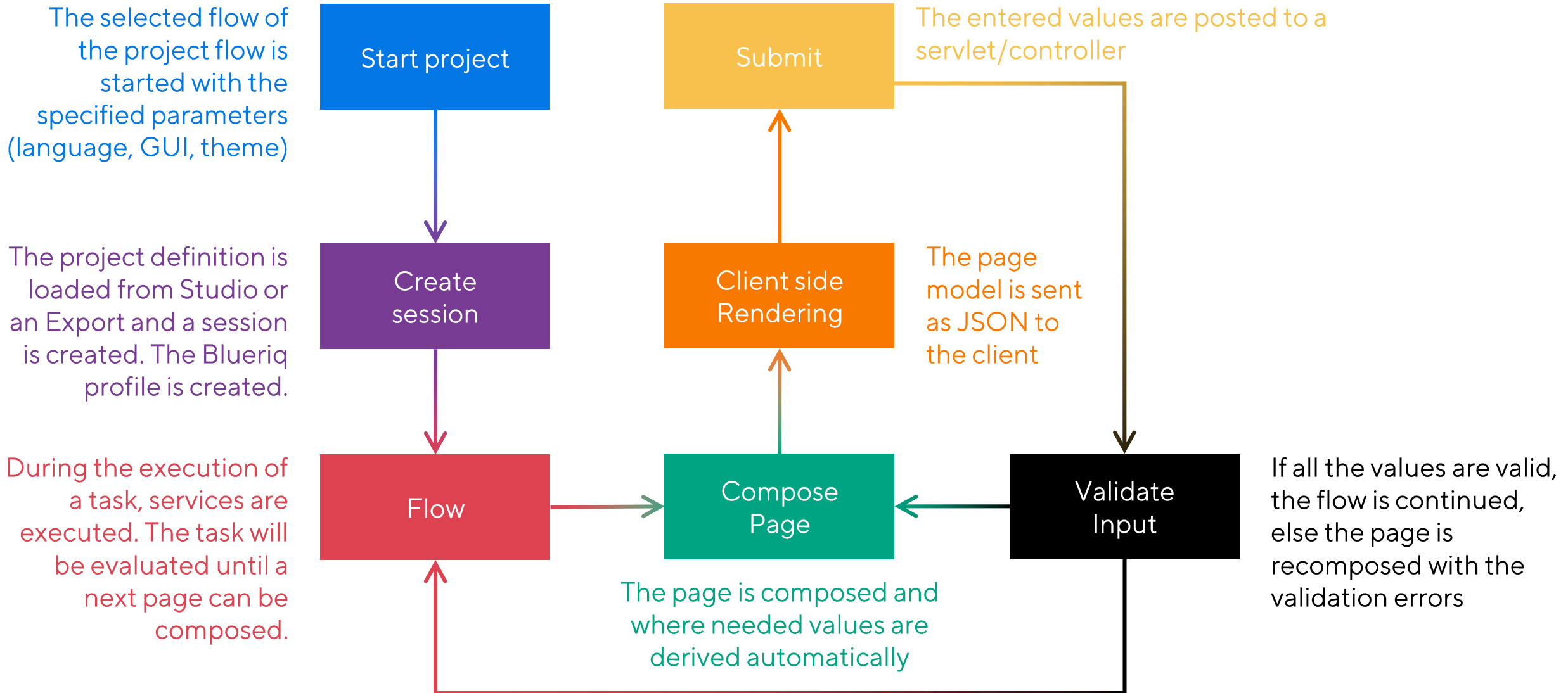
Blueriq overview



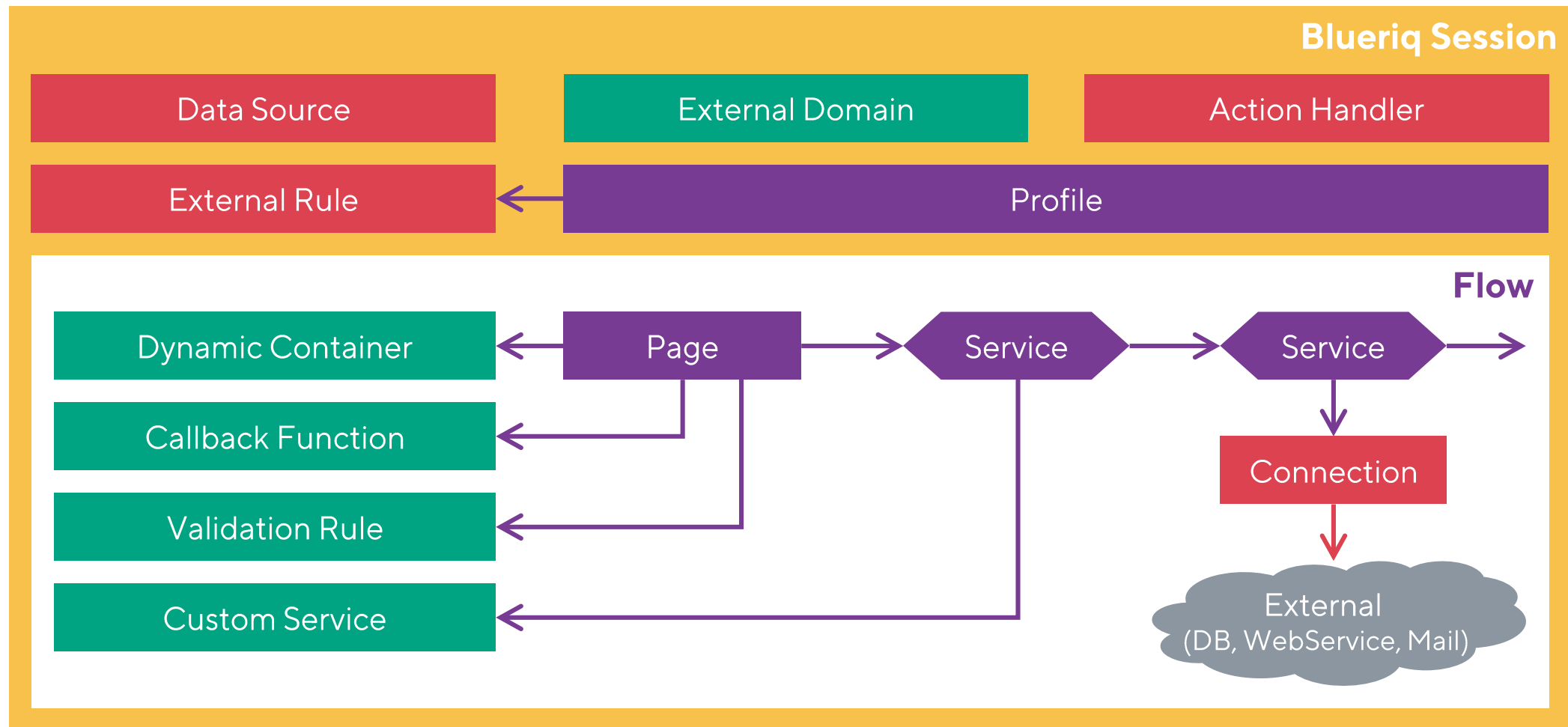
Layers



Runtime process



Runtime – Custom plugins



Endpoint; in this course

Endpoint; not in this course

No endpoint; modeled in Blueriq

External

Maven

- Java Build tool
- pom.xml files describe projects and dependencies
- We use a multi-module Maven structure
 - Each Maven module produces its own *.jar
- Root pom.xml only specifies all modules
- Parent pom.xml manages all dependency versions
- Other modules use the parent module as their parent






Spring & Spring Boot

- Java Application Framework
- Dependency Injection
 - *Spring Beans* are injection “candidates”
 - Nowadays configured in code, mainly using annotations
 - *@Autowired* will automatically select appropriate *Spring Bean* to inject
- Spring Boot used for auto-configuration of components



Blueriq Project Starters

- Maven blueprints to easily create custom Blueriq Runtime
- Distributed for all supported Application Servers
 - You choose which variant to use
- Manage what should be included in the war using Maven

Desktop > blueriq-10.5.1.1891 > Runtime > Java > Webapp > Development			
Name	Date modified	Type	Size
 projectstarter-jboss-10.5.1.1891.zip	12-12-2017 13:02	Compressed (zipped) Folder	7 KB
 projectstarter-tomcat-10.5.1.1891.zip	12-12-2017 13:02	Compressed (zipped) Folder	5 KB
 projectstarter-was-10.5.1.1891.zip	12-12-2017 13:02	Compressed (zipped) Folder	5 KB

Blueriq plugin package

- A plugin is a single component
or

- A set of related components

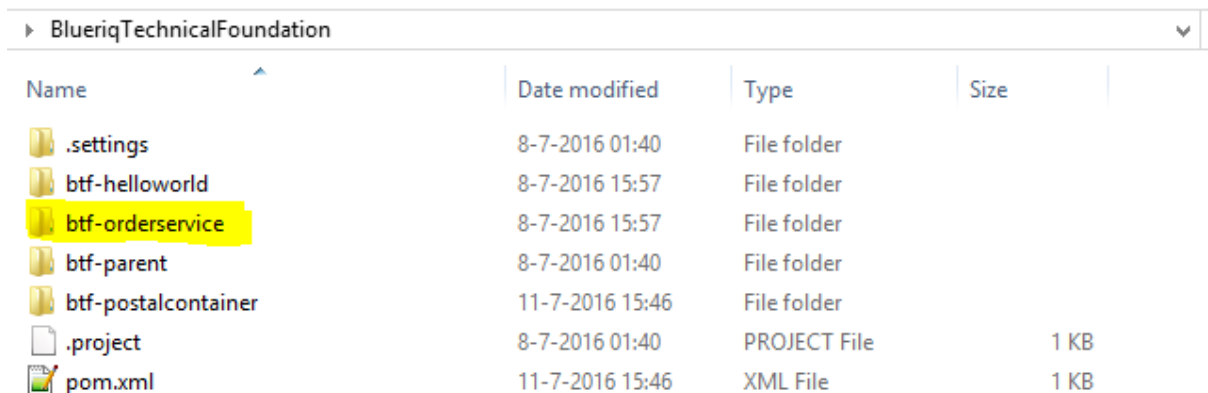


- CreateOrUpdateOrderService
- LoadOrderService
- StoreOrderService
- DeleteOrderService

- Convention: A single jar contains **a single** plugin

Blueriq plugin package

- A plugin is a single Maven module!
- On disk



BlueriqTechnicalFoundation				
Name	Date modified	Type	Size	
.settings	8-7-2016 01:40	File folder		
btf-helloworld	8-7-2016 15:57	File folder		
btf-orderservice	8-7-2016 15:57	File folder		
btf-parent	8-7-2016 01:40	File folder		
btf-postalcontainer	11-7-2016 15:46	File folder		
.project	8-7-2016 01:40	PROJECT File	1 KB	
pom.xml	11-7-2016 15:46	XML File	1 KB	

- In Eclipse



Plugin deployment

- Plugins are packaged inside the war file



- Deployment as single deliverable
 - a single WAR containing the entire application

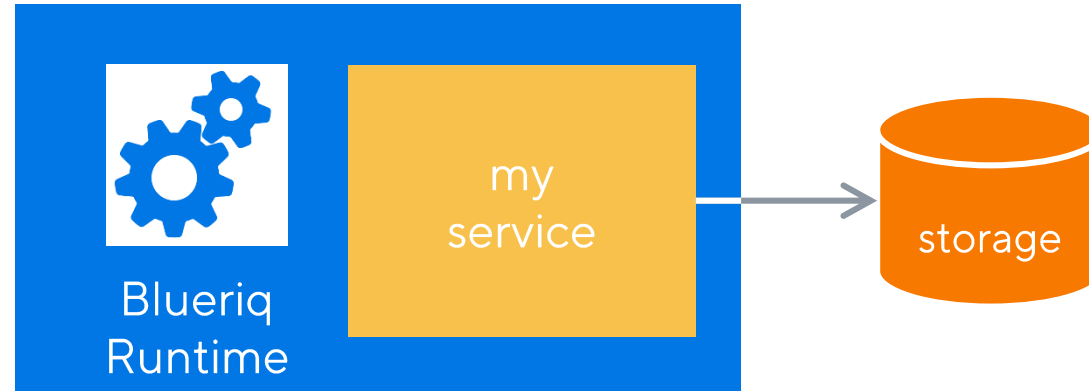
Spring Tool Suite

- Eclipse based IDE
- Useful shortcuts:
 - Ctrl + Space Code Completions
 - Ctrl + 1 Quick Actions
 - Ctrl + 2 + L Assign to variable
- In case of strange errors, using Alt + F5 to update the Maven project typically helps
 - Do not run Maven from the command line and use STS at the same time, they interfere

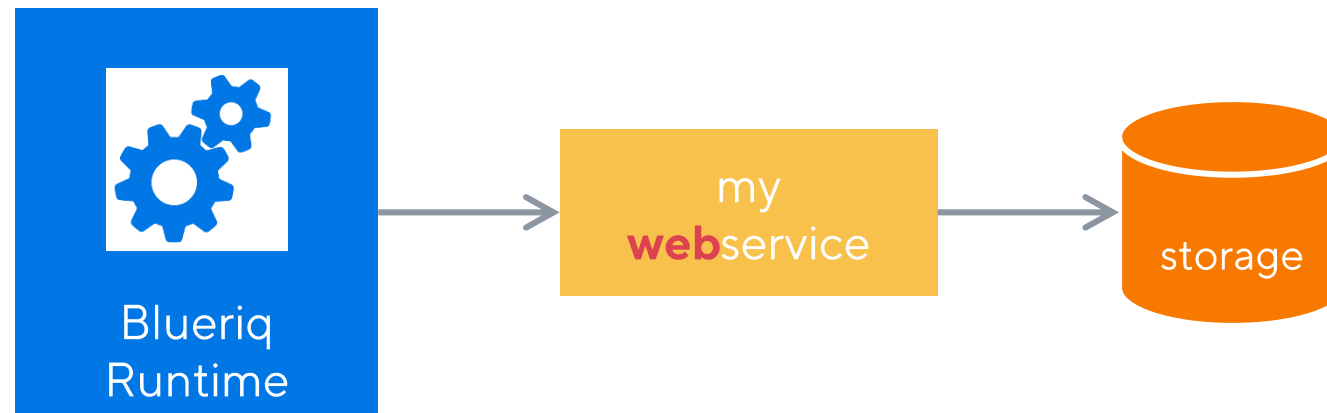


Services - architecture

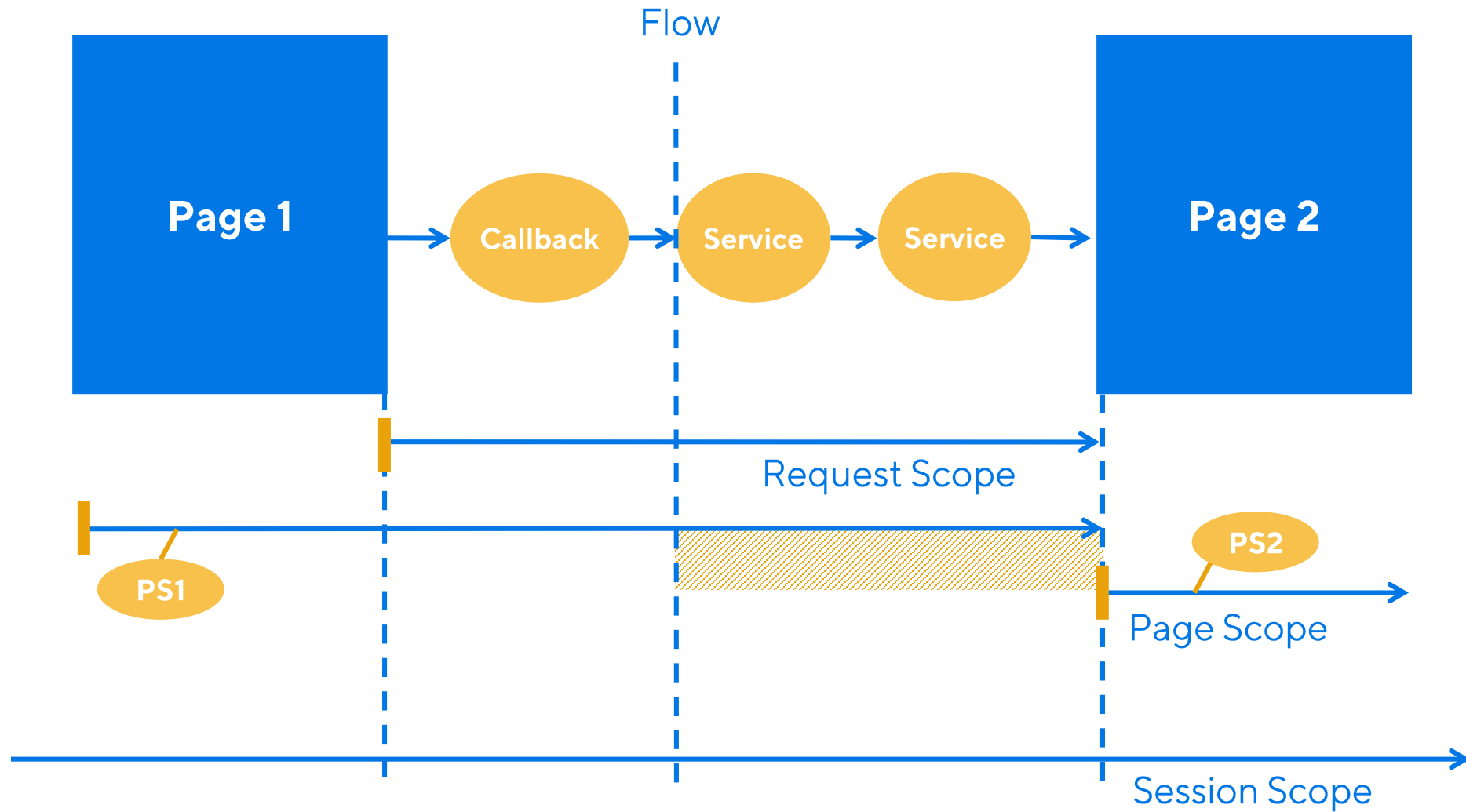
- Tactic #1



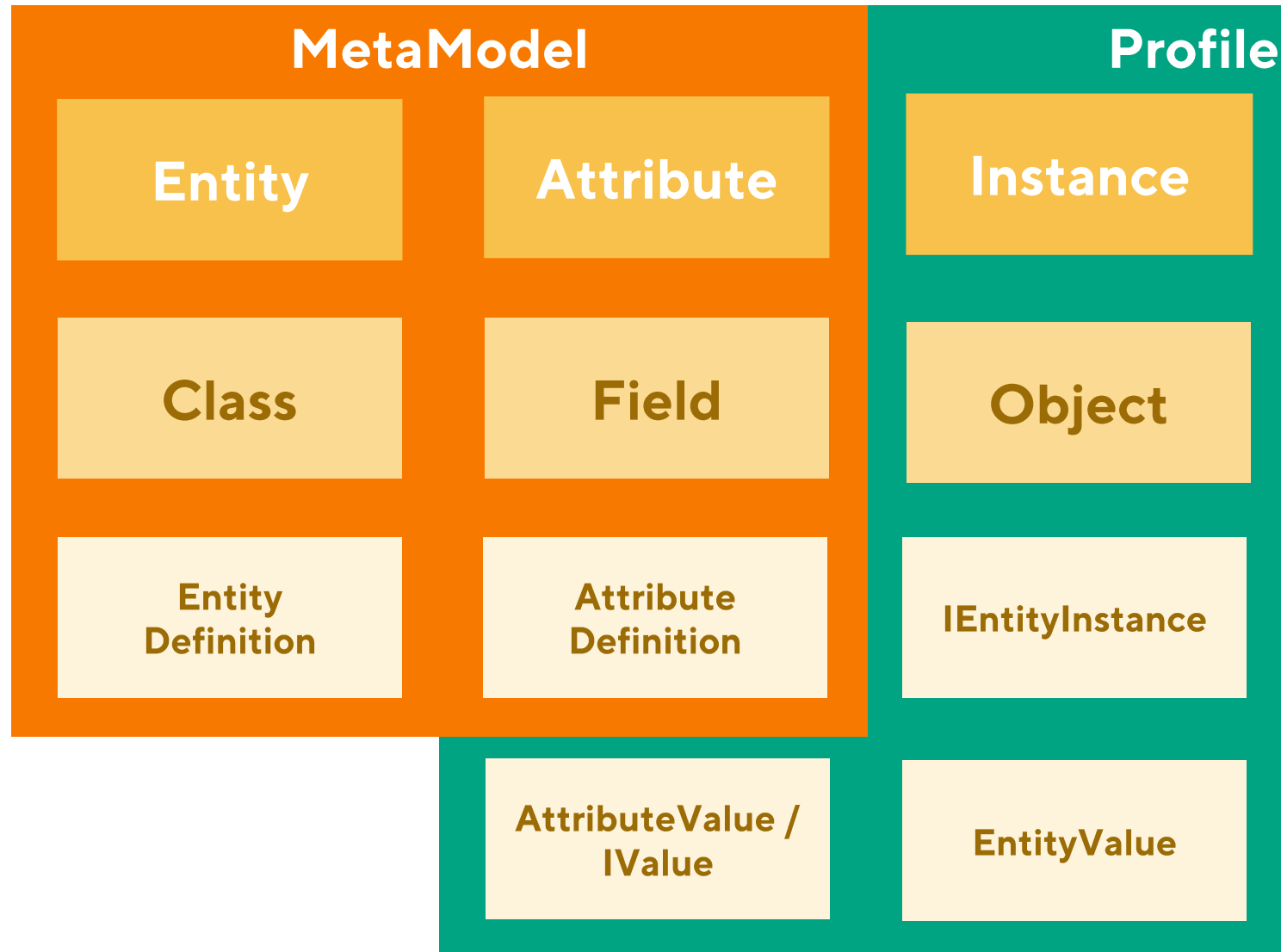
- Tactic #2



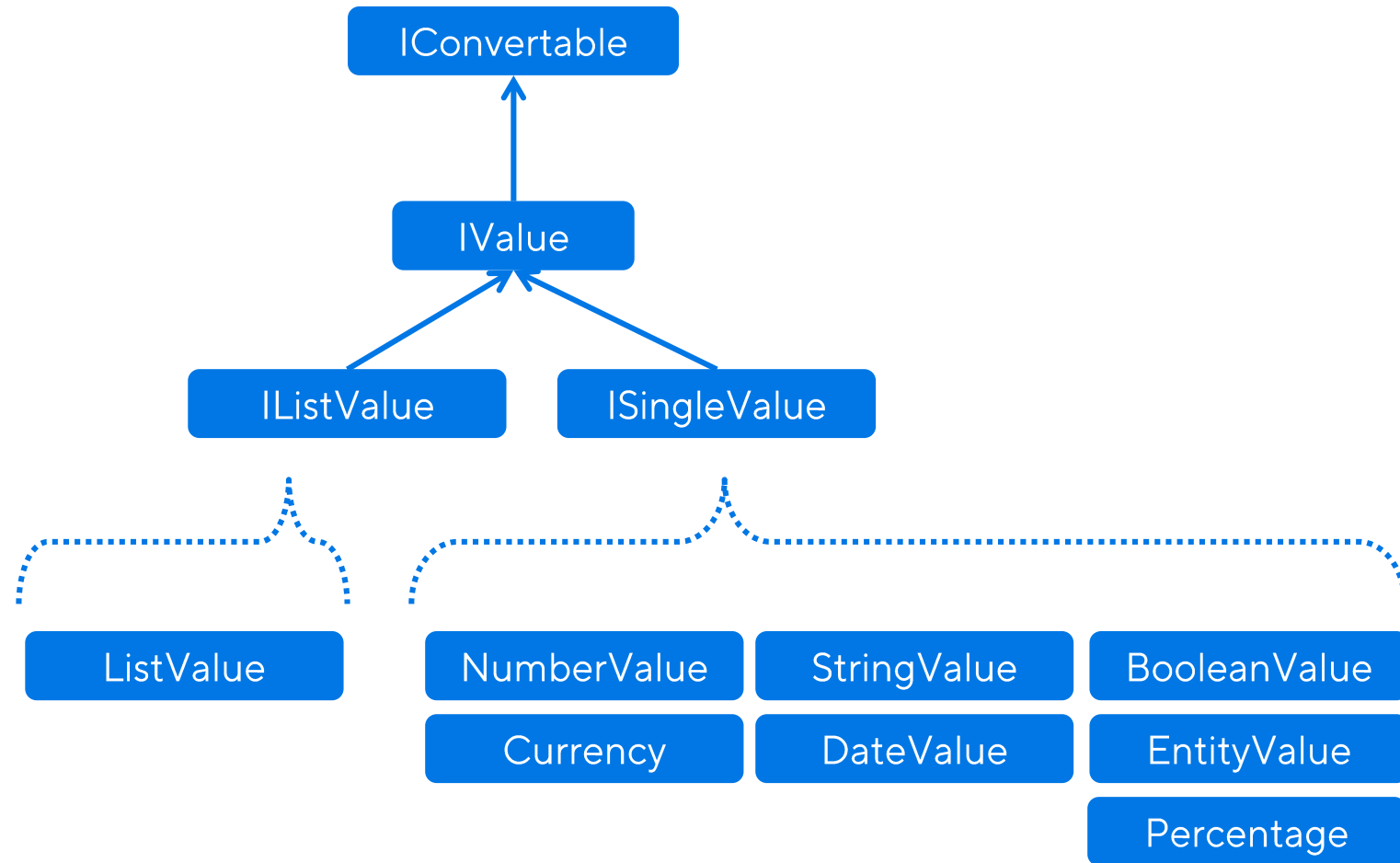
Store state in scopes



Profile – Comparison to ‘Objects’

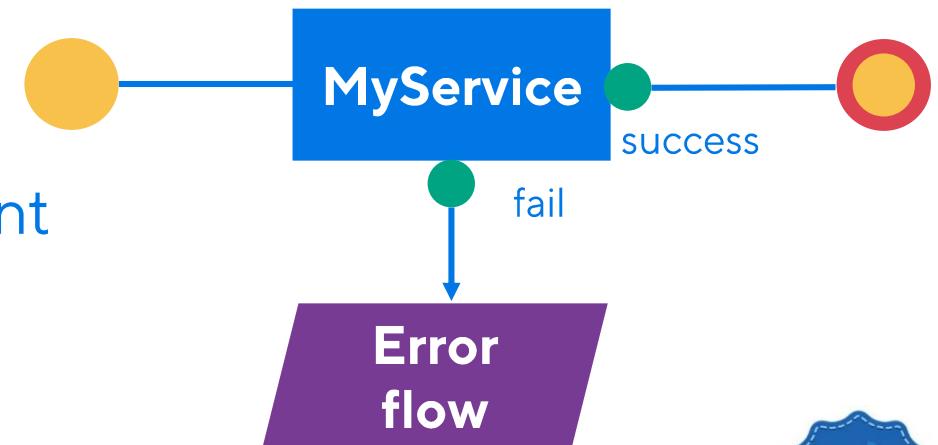


IValue - Structure

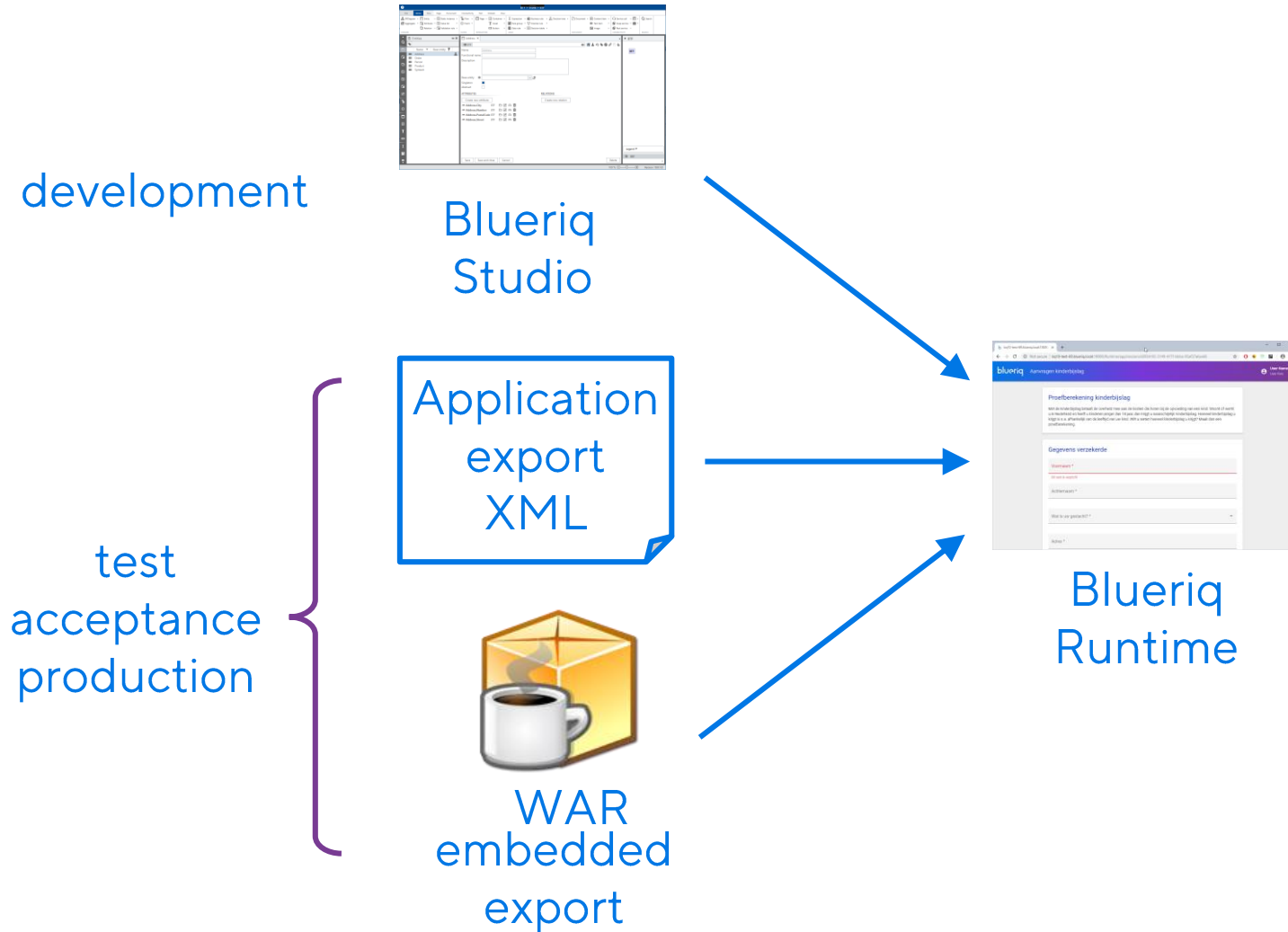


Best practice: Error handling

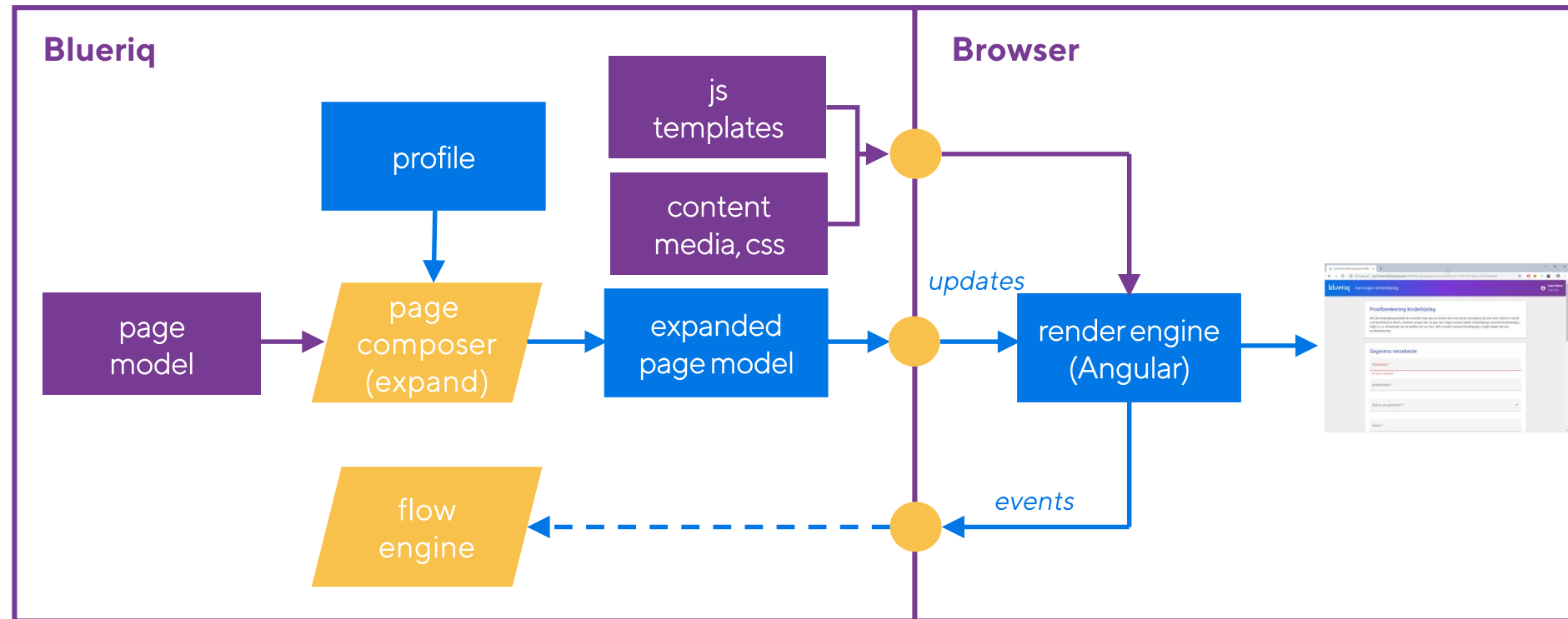
- Services must return an IServiceResult = exit event name
- Exceptions are hard to handle in Blueriq
 - (to some extend not possible)
- Business engineer can only act on service result in flow
- Rule of thumb:
 - Errors which can be handled, throw an exit event
 - negotiate with business engineer
 - Other (technical) errors → exception → crash
 - should be covered in testcases!



Blueriq modeling - Studio

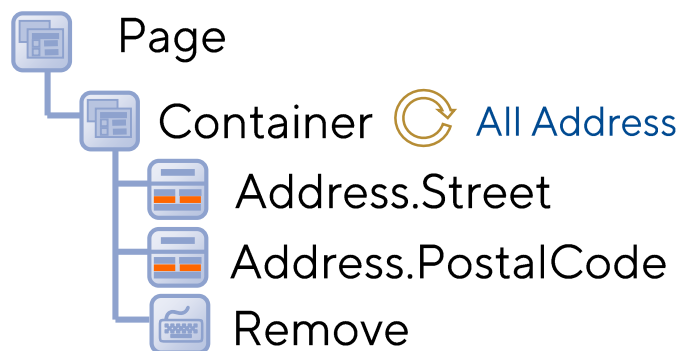


Blueriq MVC page generation – simplified



Page model

Page definition



Visual representation

The visual representation shows two address entries, each enclosed in a dashed box. The first entry has the following fields:

Street	Veemarktkade
Postal code	5222AE
<button>Remove</button>	

The second entry has the following fields:

Street	Reitscheweg
Postal code	5232BX
<button>Remove</button>	