



Reusing process definitions with Call Activity Templates

Ben Fürnröhr – CGI Germany – LBBW

Process Enterprise Orchestration at LBBW



Ben Fürnröhr
Senior BPM
Consultant



Landesbank Baden-
Württemberg
6th largest bank in
Germany



GCI Germany

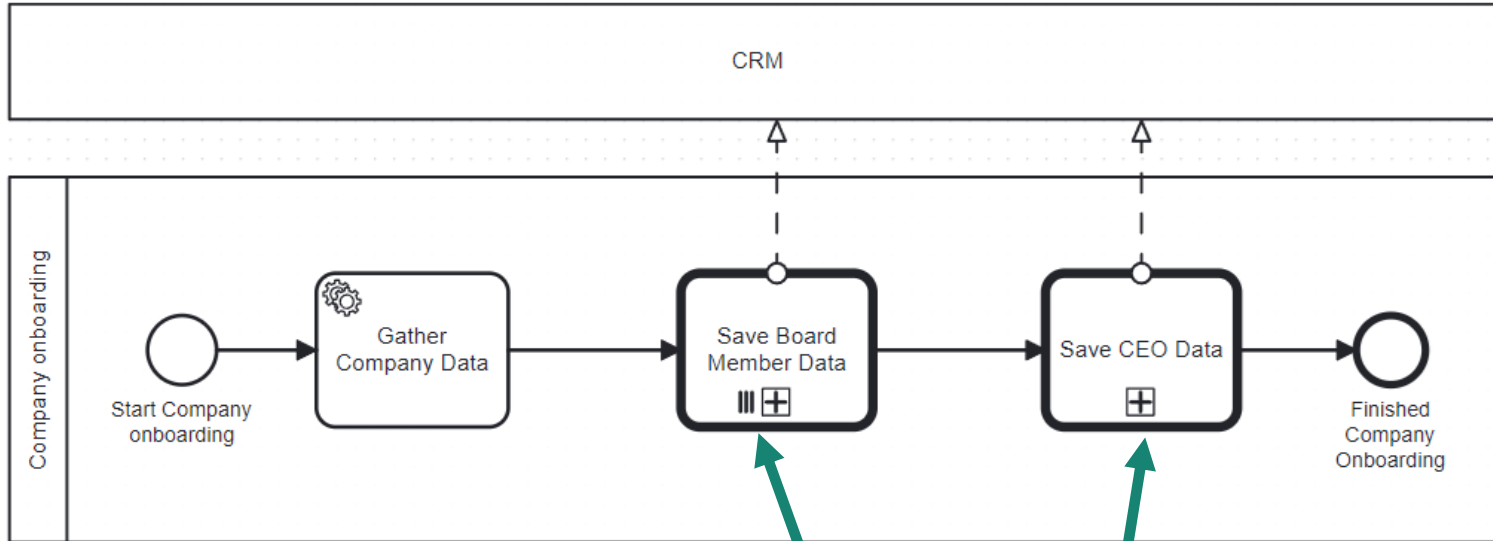


Camunda
Platinum Partner



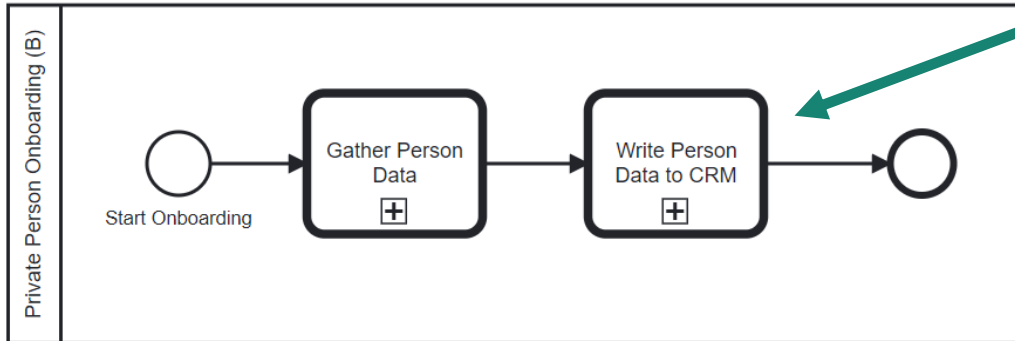
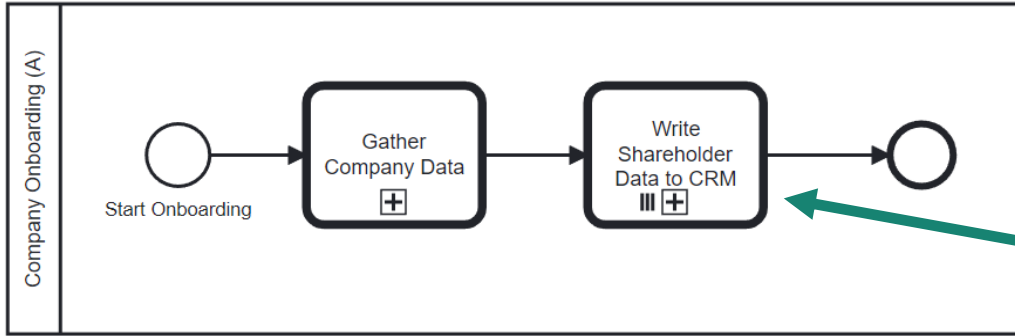
Enterprise Process
Orchestration
All based on and
around Camunda 7
& 8

Scenario 1: Flavoring



Same Process, called with different parameters

Scenario 2: Context A and Context B

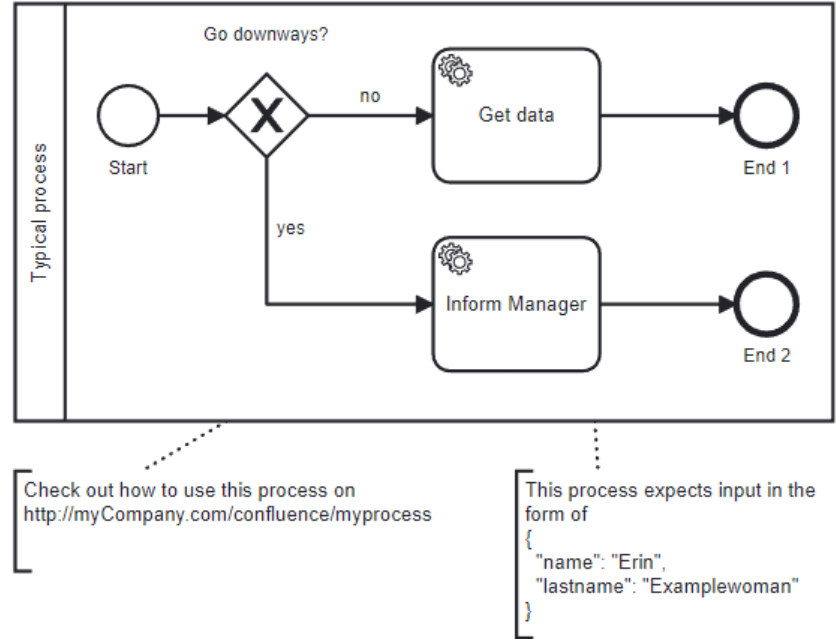


Same Process, called from two different contexts

But with what data?

Calling a Process is comparable to calling an API or a Java method, but unfortunately...

- No API specification
- No structured documentation
- No data model definition linked to process model definition



Mapping Concept 1: All Variables

In mapping propagation

☒ Propagate all variables

☐ Local

**Terrible practice and
rightfully removed in
Camunda 8!**



- Lack of control: called process gets it all
- No separation of models: model of called process is now tightly coupled to model of calling process
- Has been known to break Camunda when combined with Out mapping propagation „all“

Mapping Concept 2: One big Variable

Inputs + 1

▼ person

Local variable name

person

Variable assignment value €*

= personData

Possible in Camunda 8 but bad practice and terrible for reusability!



- Parent variable may contain unnecessary data
- No separation of models: model of called process is now tightly coupled to model of calling process
- When trying to reuse, an unrelated process needs to use the same model as parent process

Each Process deserves its own data model!

Data Point, Process A

```
{
  "company": {
    "name": "Camundanzia AG",
    "shareholders": [
      {
        "firstName": "Jake",
        "lastName": "Friend"
      },
      {
        "firstName": "Bernhard",
        "lastName": "Mover"
      }
    ]
  }
}
```

Data Point, Process B

```
{
  "person": {
    "name": "Michael",
    "middleName": "B.",
    "lastname": "Jordan"
  }
}
```

The called process should use neither of those data models but rather a model of its own!

Mapping Concept 3: One ID to rule them all

Inputs

+

1

▼ personStorageId

Local variable name

personStorageId

Variable assignment value ⓘ

= personStorageId

**Possible in
Camunda 8 but
worse than it was in
Camunda 7!**



- Relies on retrieval of previously stored data in external system
- Similar Data model separation issues
- Process now has a (possibly undocumented) precondition

Mapping Concept 4: to primitives

Inputs

+ 3

> lastName

▼ name

Local variable name

name

Variable assignment value ↻

= person.firstName

> isCompany

**This works – but
it's not pretty!**



- Called process has its own model – made up of primitives
- Data points from calling processes may be mapped to this model
- Reusability is now possible, but not practical

Mapping Concept 5: mapping with FEEL

Inputs

+ 1

▼ personData

Local variable name

personData

Variable assignment value ☺*

```
= {  
  name: person.firstName,  
  familyName: person.lastName  
}
```

Almost there!



- Called process has its own model – with all kinds of possible elements
- Data may be constructed via FEEL
- Still no documentation and still not practical for more complex data

Mapping Concept 6: wrapping with Template

Template Applied ▼

Custom properties •

FirstName

First name of the person. Empty if it is a company

Name

Name of the Company or Lastname of the person

Called Process

The Process Id of called process

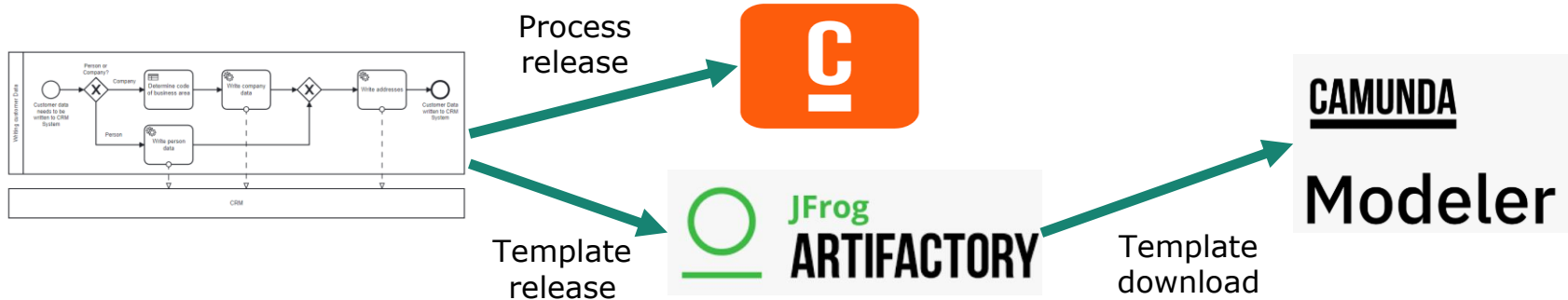
This is the way!



- Input is now user friendly
- Data structure of called process may be constructed in the templates inputMappings
- Template acts as API documentation and may be shared across users.

Let's sum it up

1. Define a data model for every process
2. Map data with feel and mask it with templates
3. Release the templates along with the process





is now part of **CGI**

Ben Fűrnrrohr

Senior Consultant

Mobil: [+49 \(175\) 5114 223](tel:+491755114223)

E-Mail: Benjamin.Fuernrohr@novatec-gmbh.de

Novatec Consulting GmbH

Bertha-Benz-Platz 1

D-70771 Leinfelden-Echterdingen

T. +49 711 22040-700

info@novatec-gmbh.de

www.novatec-gmbh.de

