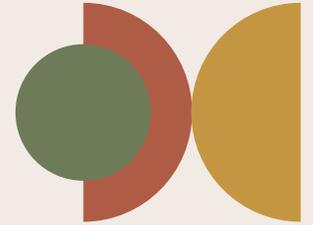


# Agile Factory:



## Ein Zusammenarbeitsmodell in der Praxis

Geschäftsprozesse digitalisieren  
sowie persönliche und digitale Kanäle verbinden.

valiant

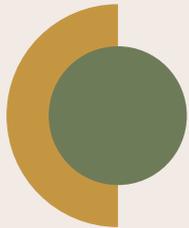


WeEvolve

finnova



swisscom



# Inhalt

**01**

## **Wer.Wie.Was**

Agile Factory – Auftrag  
Architektur & Prozesse

**02**

## **Fachlich**

Spezifikation - Testing &  
Abnahme - Betrieb

**03**

## **Technisch**

Implementation  
Deployment & Releasing

**04**

## **Sonst noch**

Was wir so lernten  
Kontakt - Fragen



01

# Wer.Was.Wie

Agile Factory – Auftrag – Architektur & Prozesse



**Christian Pfister**

IT-Architekt

**valiant**

# Wer ? Agile Factory

**valiant**

**Auftraggeber**

Spezifikation  
Testing  
2nd Level Support



**finnova.**

**BPF / Camunda**

Implementierung  
Prozesse  
3rd Level Support



 **swisscom**

**Betrieb / MAP**

Betreiber  
Entwicklung MAP  
1st Level Support



**WeE**volve



# Was ? Auftrag

Geschäftsprozesse digitalisieren sowie persönliche und digitale Kanäle verbinden.

Kanalübergreifend

Nutzung von Synergien

Effizienzsteigerung

Medienbrüche reduzieren

Datenhaltung in Finnova Core

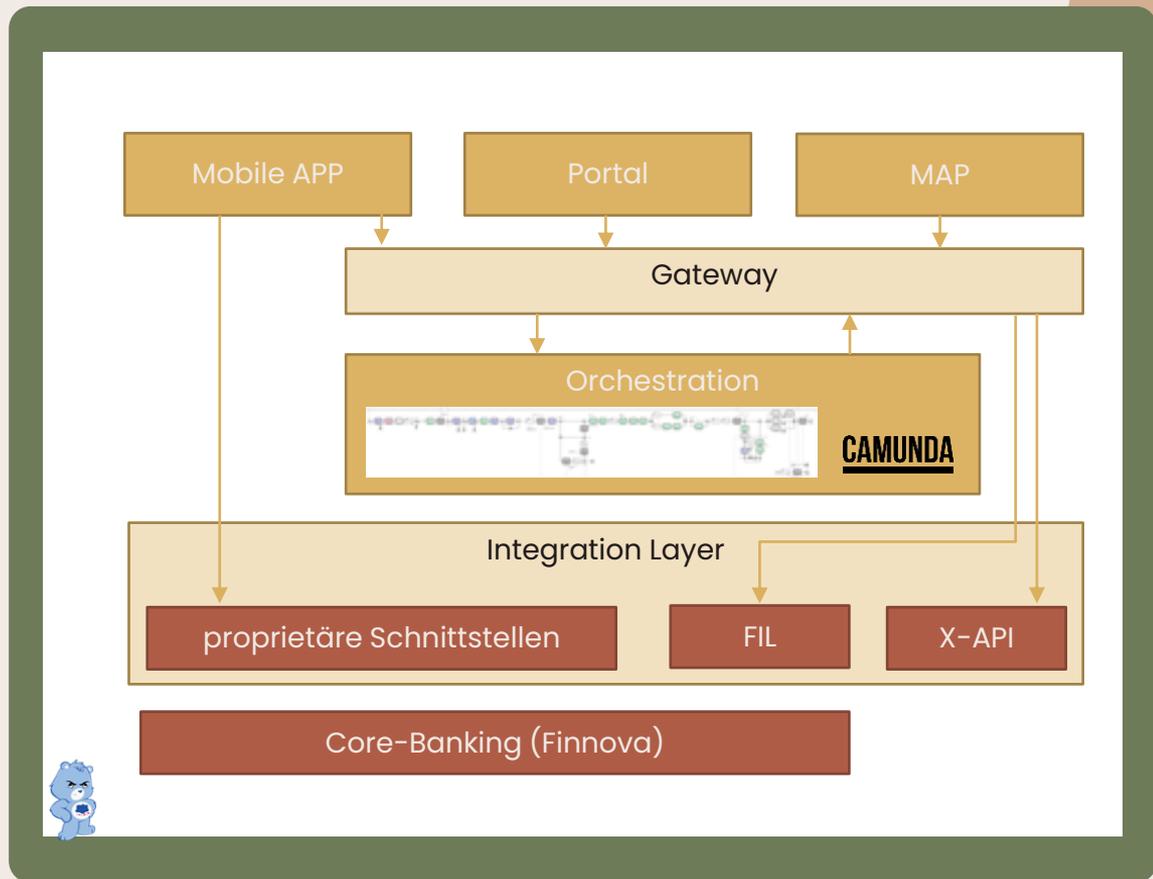
Kundendaten in 360° Sicht



# Wie? Architektur

Konsequenter Einsatz von  
Camunda als  
Orchestrationslayer

Omnichannel fähige  
Prozesse



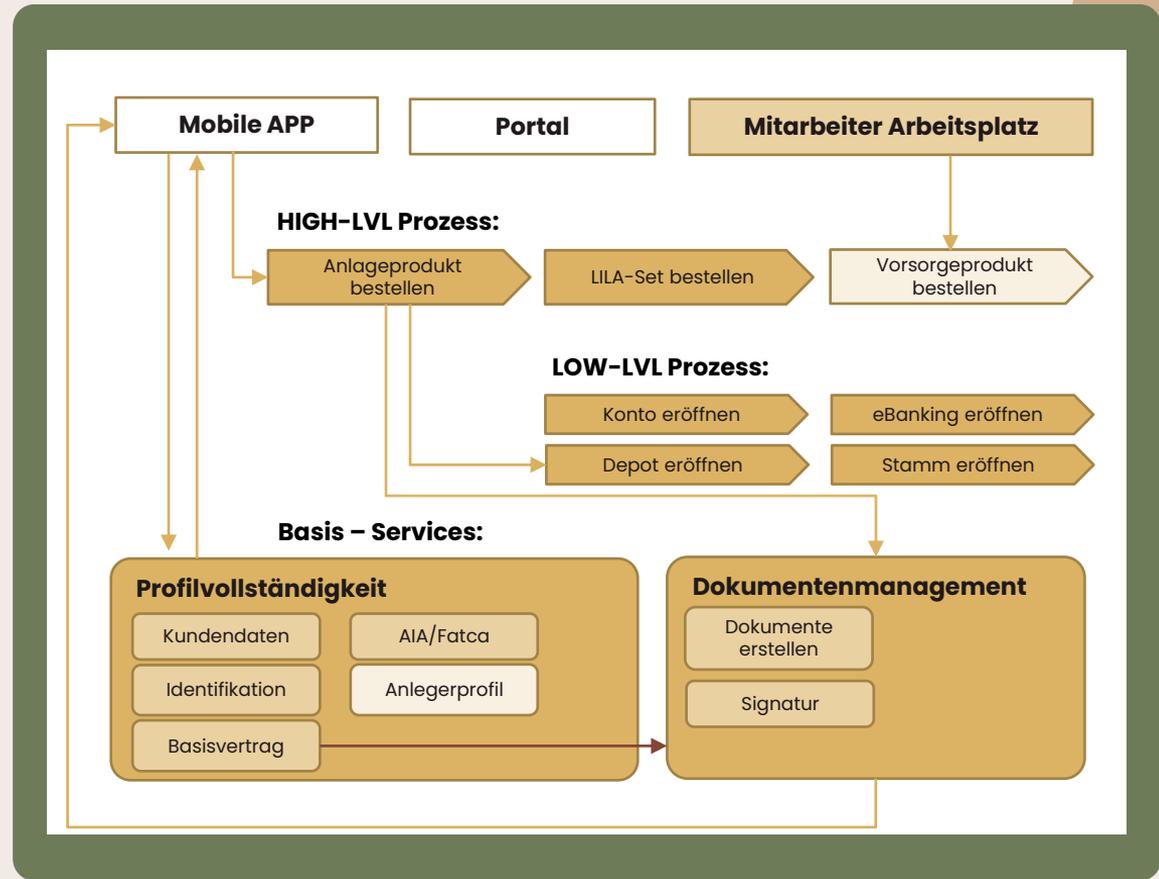
# Wie ? Prozesse

Unterteilung der Prozesse  
in:

Basis-Services

LOW-Level Prozesse

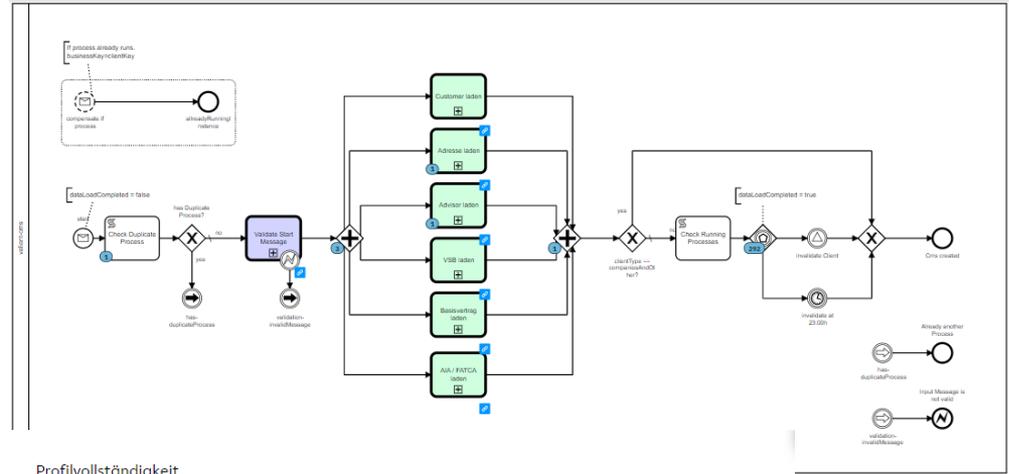
HIGH-Level Prozesse



# Wie? Prozesse

## Basis-Services:

Vorbedingungen & generische Services für das Ausführen von Low- und High-Level Prozessen



Profilvollständigkeit

### Status Kundenprofil



- Kundendaten
- Identifikationsdokument
- Basisvertrag ⓘ
- Selbstdeklaration AIA/FATCA

AKTUALISIEREN

AKTUALISIEREN

# Was ? Erreichtes & Ausblick

Bisherige Projektlaufzeit: 2,5 Jahren

**91 Prozess Definitionen** auf der Produktion

**1,3 Millionen CAMUNDA API** Aufrufe pro Monat

**1,7 Millionen Core Banking API** Aufrufe pro Monat

**4.8 Millionen ausgeführte Activity** Instances pro Monat

Ausblick:

automatisiertes Kunden-Onboarding

Basisservice für Anlageprofil

Basisservice für Vollmachten

und **noch viele weitere spannende Ideen...**



02

# Fachlich

Spezifikation – Testing & Abnahme – Betrieb



**Karin Trachsel**

Solution Architektin

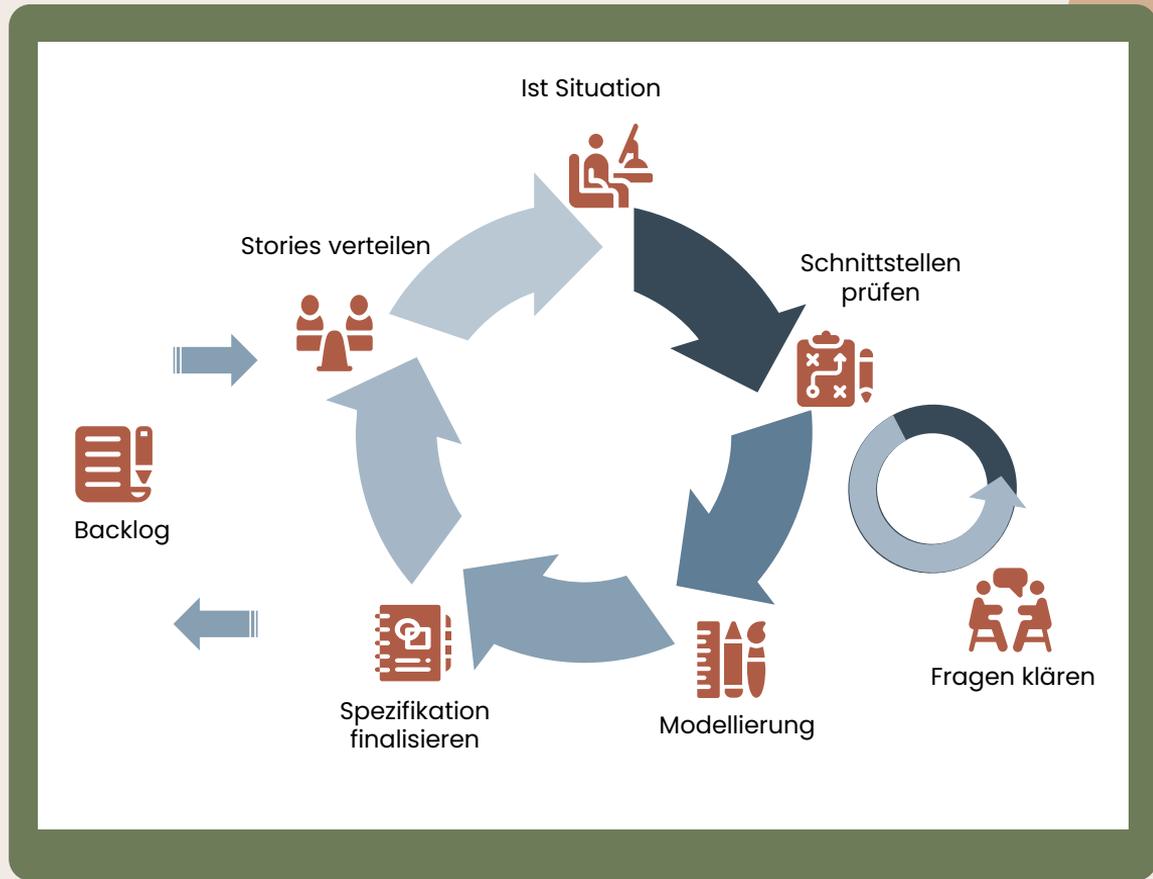


WeEvolve



# Aufgaben

Ist-Prozess aufnehmen  
Schnittstellen prüfen  
Soll-Prozesse modellieren  
Testszenarien schreiben  
Entwicklung begleiten  
Umsetzung abnehmen  
Überführung in Produktion



# Best Practice

Ausgangslage und Scope

Spezifikation UI

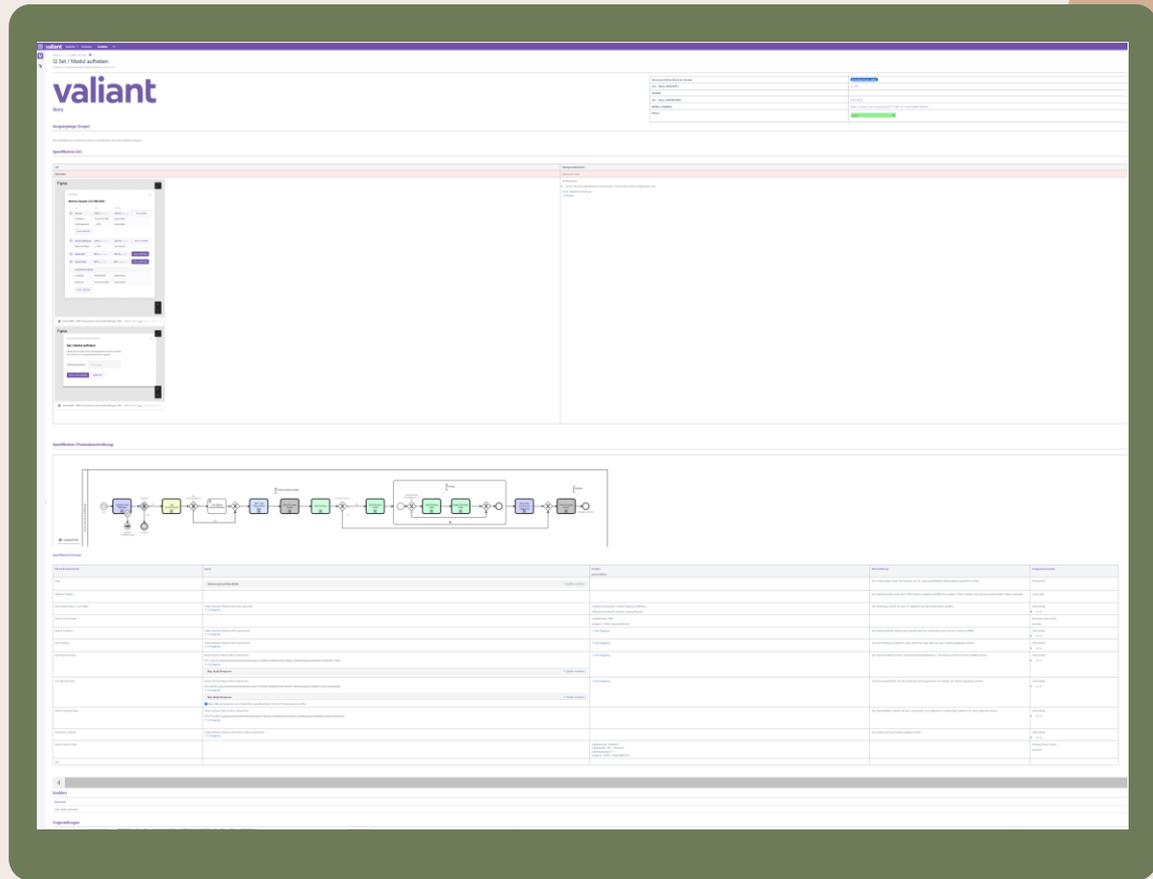
Spezifikation Prozess

BPMN-Modell

Beschreibung BPMN-Modell

Enablers

Fragestellungen



# Spez: User Interface

- UI-Mockups
- Beschreibung Interaktionen
- Beispiel Daten
- Request für Prozess-Start
- Requests für User Tasks

The screenshot displays a specification tool interface with two main panels. The left panel, titled 'UX', shows a Figma mockup of a user interface for 'Marina Kessler (13.786.203)'. The mockup lists various financial modules and products:

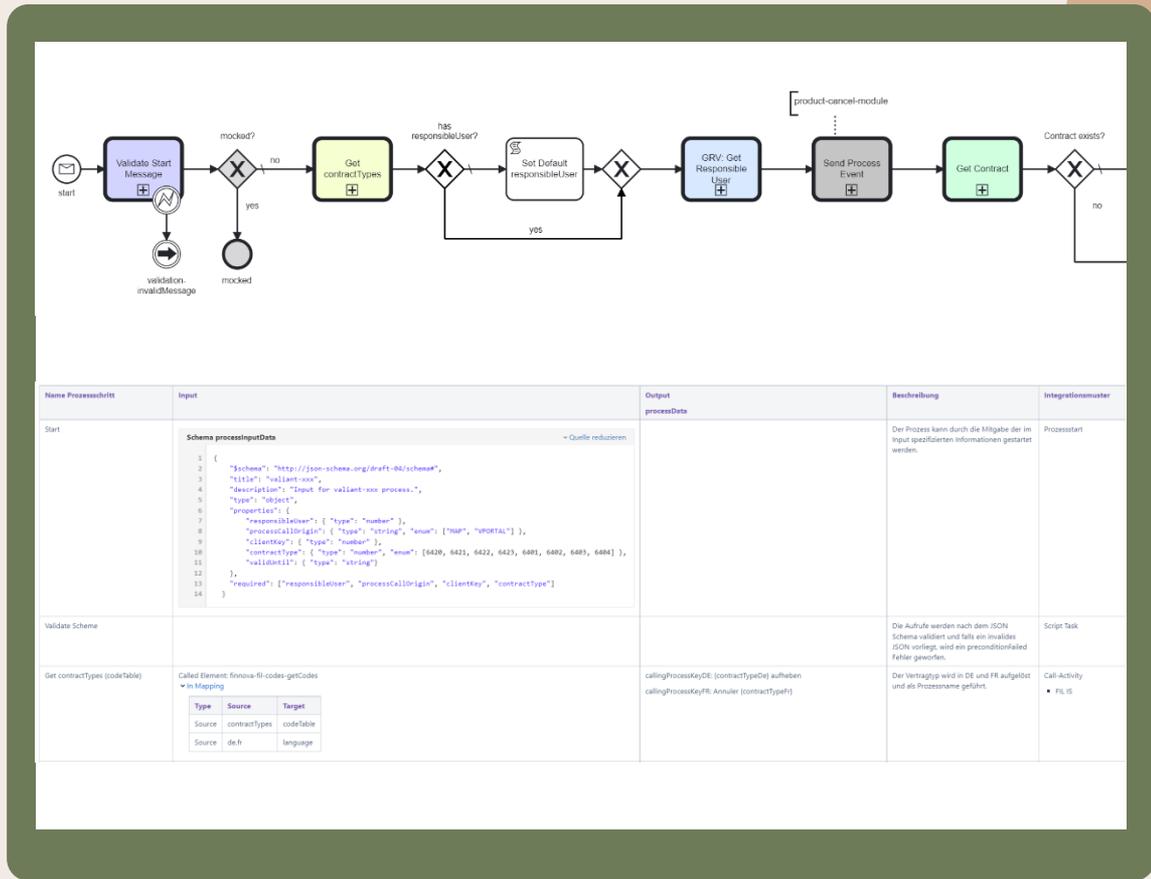
Typ	Preis	Empfänger	Aktion
Lite-Set	CHF 6.000.000.000	CHF 40.000.000.000	SET AUFHEBEN
Privatkonto	15.331.023.758,6	Marino Keller	
Debit Mastercard	...8765	Marino Keller	
KONTO ERÖFFNEN			
Modul Kreditkarte	CHF 6.000.000.000	CHF 40.000.000.000	MODUL AUFHEBEN
Mastercard Silber	...7396	Hans Muster...	
Modul Welt	CHF 3.000.000.000	CHF 10.000.000.000	MODUL ERÖFFNEN
Modul Papier	CHF 3.000.000.000	CHF 10.000.000.000	MODUL ERÖFFNEN
Zusätzliche Produkte			
e-Banking	20ND067666	Marino Keller	
Sparkonto	15.331.023.758,6	Marino Keller	
KONTO ERÖFFNEN			

The right panel, titled 'Akzeptanzkriterien', shows the API request details for the 'SET / Modul aufheben' action:

```
POST: {{bpfuURL}}/message
Payload
{
  "variables": {
    "clientKey": "66238505910423",
    "responsibleUser": "{{responsibleUser}}",
    "processCallOrigin": "MAP",
    "contractType": "6420"
  },
  "businessKey": "Product CancelModule",
  "resultVariableNames": [
    "callingProcessKeyDE",
    "callingProcessKeyFR",
    "endStatus"
  ]
},
"waitTimeoutInMs": 10000
}
```

# Spez: Prozess

- BPMN- & DMN-Modelle
- Input Mapping
- Output Mapping
- Beschreibung
- Integrations Pattern



# Spez: Prozess

BPMN- & DMN-Modelle

Input Mapping

Output Mapping

Beschreibung

Integrations Pattern

Schema processInputData

Quelle reduzieren

```
1  {
2    "$schema": "http://json-schema.org/draft-04/schema#",
3    "title": "valiant-xxx",
4    "description": "Input for valiant-xxx process.",
5    "type": "object",
6    "properties": {
7      "responsibleUser": { "type": "number" },
8      "processCallOrigin": { "type": "string", "enum": ["MAP", "VPORAL"] },
9      "clientKey": { "type": "number" },
10     "contractType": { "type": "number", "enum": [6420, 6421, 6422, 6423, 6401, 6402, 6403, 6404] },
11     "validUntil": { "type": "string" }
12   },
13   "required": ["responsibleUser", "processCallOrigin", "clientKey", "contractType"]
14 }
```

Neuer Service: finnova-fil-is-genericGet

GET: {{sifURL}}/paymentrulesadministration/api/v1/banks:bankNumber/clients:clientKey/paymentRules?  
inclInvalid=false

▼ In Mapping

Type	Source	Target
Source Expression	todo	serviceName
Source	impersonateUserId	impersonateUserId
Source	clientKey	clientKey

03

# Technisch

Implementation - Deployment & Releasing



**Pascal Mengelt**

Software Engineer seit 2003

Camunda Champion

DMN Tester / Camundala

Scala Enthusiast

**finnova**

# Grundsätze

Wiederverwendbarkeit

Wartbarkeit

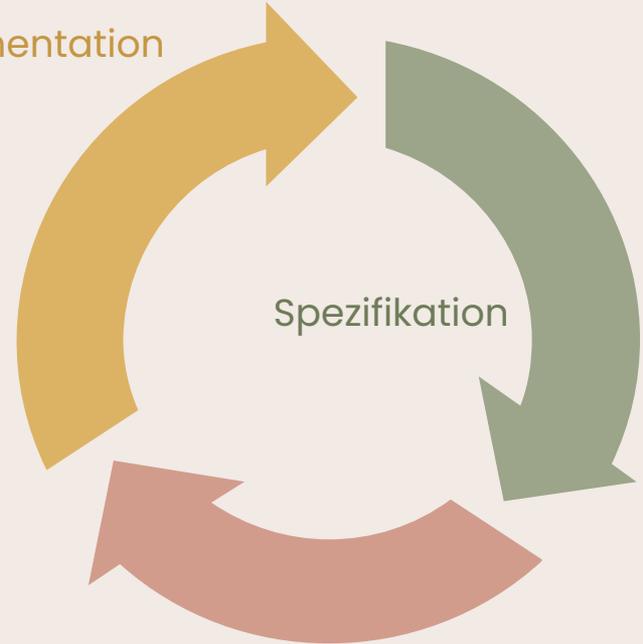
Dokumentierte Schnittstellen

Simuliere Prozess Pfade

Schnittstellen Test Requests

Gemeinsame Entwicklung

Dokumentation



Implementation

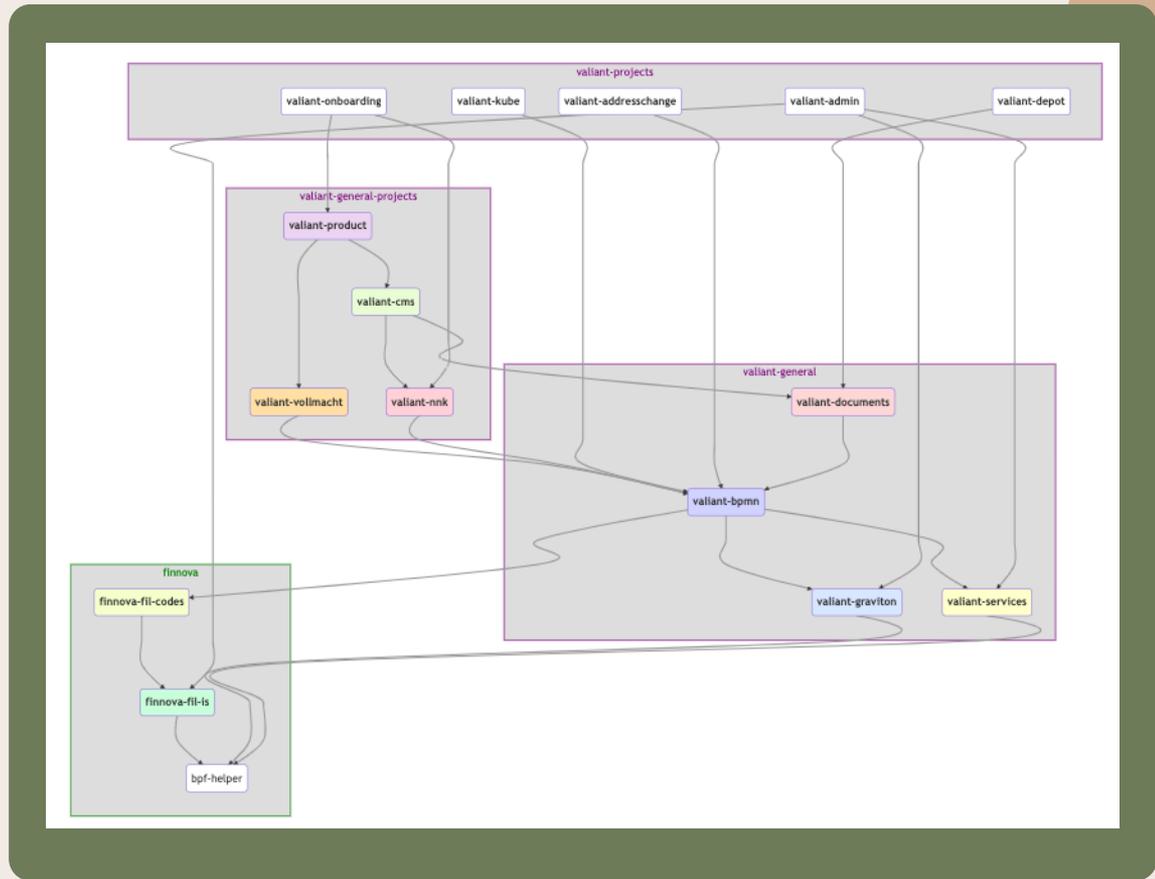


# Wieder verwend barkeit

## Projekte

Pattern

Call Activities



# Wieder verwend barkeit

Projekte  
**Pattern**  
**Call Activities**

## Process Pattern

We try to establish Patterns for doing the same tasks. This documentation lists them and gives you some examples.

### Impersonate User

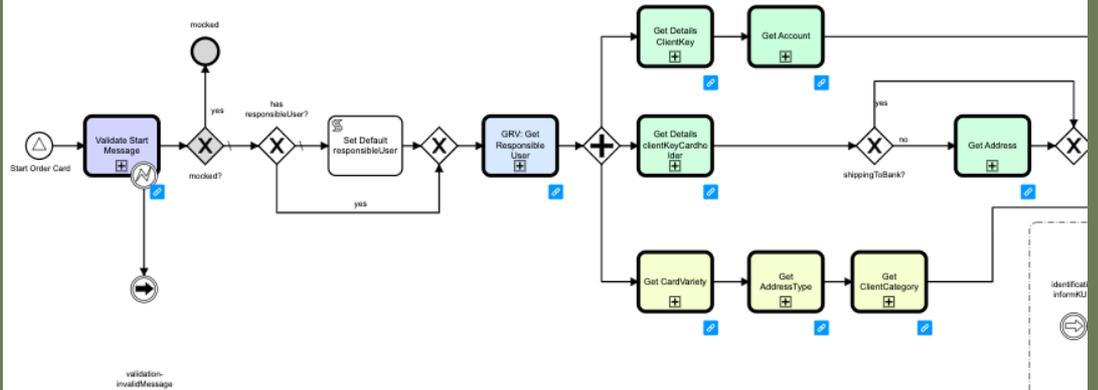
There are Use Cases where you want to run a service under a different user, then the one that last interacted with the process. > In the processes you need to set the `impersonateUserId` if you call another Gateway Service - e.g. FIL-IS Service. > Otherwise the service is called with the technical user.

So you can set the variable `impersonateUserId` in the process manually.

To set it automatically you must use a script that is defined by the variable.

#### Process Pattern

- Impersonate User
- Send Notification
- Inform User
- Validate Process Inputs
- Generic Service Processes
- Caching Orchestrated Variables
  - Getting the variables
  - The orchestration cache
- Issues
- Usage
- Mocking Sub-Processes



# Domain Driven Process Development

Dokumentation

Simulation

Validieren von DMNs

Test Client (Postman)

Zukunft: Code

```
lazy val `Product OrderCard` =  
    process(  
        processName,  
        ProductOrderCardIn(),  
        ProductOrderCardOut()  
    ).withProcessDescr(  
        "Process to order a Card.",  
        postmanIds = "12738322-05e7a5"  
    )  
    lazy val orderCardUT = userTask(  
        id = "Product Order Card",  
        ProductOrderCardUTIn(),  
        ProductOrderCardUTOut()  
    )
```

CAMUNDA  
COMMUNITY  
SUMMIT  
2022

Domain  
Spec

BPMN DSL

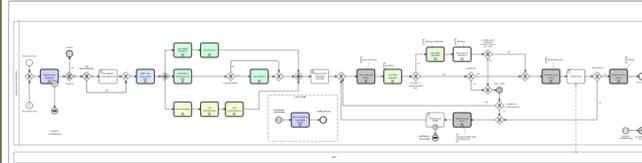
Function DSL

# API Doku mentation

```
document(  
  api(`Product LilaSet`),  
  api(`Product LoadLilaSetModules`)(  
    `invalidate process for client`  
  ),  
  api(`Product ModuleCreditcard`),  
  api(`Product LoadPoas`),  
  api(`Product OrderCard`)(  
    ProductOrderCard.postprocessCardUT,  
    orderCardUT  
  ),  
  api(`Product OrderCreditcard`)(  
    orderCreditcardApprovaUT,  
    orderCreditcardUT,  
    ProductOrderCreditcard.postprocessOrderUT,  
    cancelCreditcardUT,  
  )  
)
```

## Process: valiant-product-orderCard

Process to order a Card.



BPMN.IO

Download: product-orderCard.bpmn

Calling Process: `valiant-product:valiant-product-orderCard`

### Postman Request

#### Used in 1 Project(s) (EXPERIMENTAL)

- valiant-onboarding
  - valiant-onboarding

#### Uses 6 Project(s) (EXPERIMENTAL)

REQUEST BODY SCHEMA: application/json

clientKey	integer <int64>
required	Technical client key of the client
clientKeyCardholder	integer <int64>
accountKey	integer <int64>
required	
responsibleUser	integer <int64>

HEAD /process/ProductOrderCard/valiant-product-order

### Request samples

#### Payload

Content type  
application/json

Copy Expand

```
{  
  "clientKey": 74854564837991,  
  "accountKey": 72856681586075,  
  "responsibleUser": 57441573666194,  
  "cardVariety": 910,  
  "dailyLimit": 3000,  
  "monthlyLimit": 5000,  
  "multiAccountsAssign": false,  
  "shippingToClient": {  
    "validFrom": "2022-08-10",  
    "validUntil": "2299-12-31",  
    "addressType": 11,  
    "addresseeKey": 74854564837991  
  }  
}
```

### Response samples

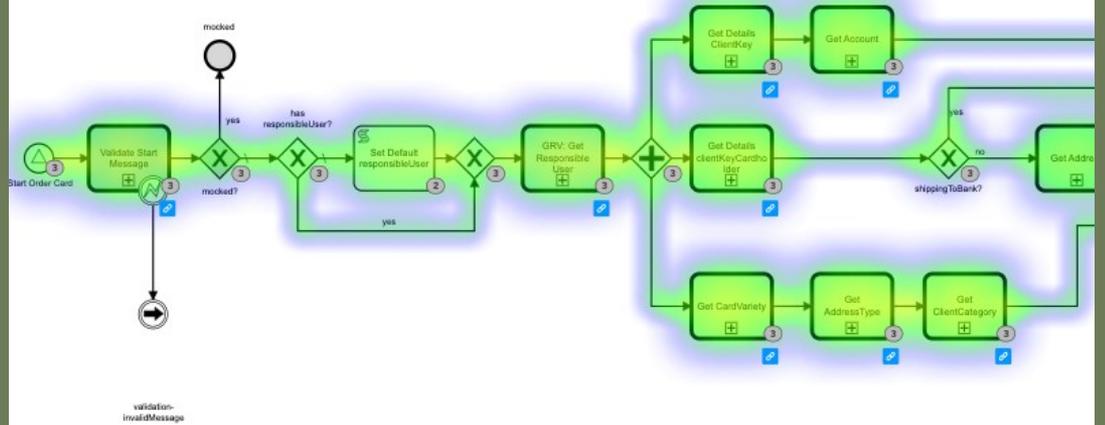
200

Content type  
application/json

```
{  
  "endStatus": "ordered",  
  "callingProcessKeyDE": "Neubestellung DMC",  
  "callingProcessKeyFR": "Nouvelle commande DMC"  
}
```

# Simulation

```
simulate(  
  scenario(`Product OrderCard`)(  
    `postprocessCardUT with cmsMock`,  
    `orderCardUT not ordered`,  
    `postprocessCardUT with cmsMock -  
    orderCardUT`  
  ),  
  scenario(`Product OrderCard shipping`  
    `postprocessCardUT sentToBank with`  
    `orderCardUT shippingToBank`  
  ),  
  scenario(`Product OrderCard differen`  
    `postprocessCardUT with cmsMock`,  
    `orderCardUT`  
  )  
)
```



State	ID	Business Key
Completed	9736c6c0-bf	Product OrderCard shippingToBank
Completed	9736c6cb-bf	Product OrderCard different clientKeyCardholder
Completed	9733925a-bf	Product OrderCard

# DMNs

```
createDmnConfigs(  
    bpmnProductGroupsDmn  
        .testUnit  
        .testValues("productToOpen", 1, 2, 3, 4, 5)  
        .inTestMode,  
    mapActionsFromSubStatusDmn  
        .testUnit  
        .acceptMissingRules  
        .dmnPath(defaultDmnPath(bpmnSubStatusAllDmn))  
        .testValues("subStatusKey", "address-change")  
        .inTestMode,  
    bpmnSubStatusAllDmn  
        .testUnit  
        .testValues("subStatusKey", "all")  
        .inTestMode,  
)
```



valiant-bpmn-productGroups src/main/resources/camunda/bpmn-productGroups.dmn

Unit Test  
Hit Policy: COLLECT

Successful Tests

<input type="checkbox"/>	productToOpen	Dmn Row	productToOpen	productGroups	finnovaProductNumbers	productDescriptionDe	productDescriptionFr
<input type="checkbox"/>	1	1	1, null,-99	1	5010	Execution only	Execution only
<input type="checkbox"/>	2	2	2, null,-99	2	5010,3	Beratung Basic	Conseil Basic
<input type="checkbox"/>	3	3	3, null,-99	3	5010,3,12	Beratung Sélect	Conseil Sélect
<input type="checkbox"/>	4	4	4, null,-99	4	5010,3,12	Beratung Professione..	Conseil Professione..

	When = 1 productToOpen integer	Then productGroups integer	And finnovaProductNumb... string	And productDescriptionDe string	And productDescriptionFr string	
1	1, null,-99	1 = 1	"5010" = 5010	"Execution only" = Executio n only	"Execution only" = Executio n only	-
2	2, null,-99	2	"5010,3"	"Beratung Basic"	"Conseil Basic"	-
3	3, null,-99	3	"5010,3,12"	"Beratung Sélect"	"Conseil Sélect"	-

# Test Clients

```
document(  
  api(`Product LilaSet`),  
  api(`Product LoadLilaSetModules`)(  
    `invalidate process for client`  
  ),  
  api(`Product ModuleCreditcard`),  
  api(`Product LoadPoas`),  
  api(`Product OrderCard`)(  
    ProductOrderCard.postprocessCardUT,  
    orderCardUT  
  ),  
  api(`Product OrderCreditcard`)(  
    orderCreditcardApprovaUT,  
    orderCreditcardUT,  
    ProductOrderCreditcard.postprocessOrderUT,  
    cancelCreditcardUT,  
  )  
)
```

The screenshot displays a REST client interface with a sidebar on the left listing API endpoints under the 'Product OrderCard' folder. The main area shows a selected endpoint: 'POST Product OrderCard: StartProcess Async'. The right panel shows the request configuration, including the method 'POST', the URL '[[bpfURL]]/process-definition/', and the body type 'form-data'. The response body is a JSON object with the following structure:

```
1  {  
2    "variables": {  
3      "clientKey": "{{clientKey}}",  
4      "accountKey": "72856681556075",  
5      "responsibleUser": "57441573680",  
6      "cardVariety": "910",  
7      "dailyLimit": "3000",  
8      "monthlyLimit": "5000",  
9      "multiAccountsAssign": false,  
10     "shippingToBank": false,  
11     "shippingToClient": {  
12       "validFrom": "2022-08-10",  
13       "validUntil": "2299-12-31",  
14       "addressType": "11",  
15       "addresseeKey": "748545648379",  
16     }  
17   },  
18   "businessKey": "Product_OrderCar
```

# Release Project

Erstelle Package

Generiere Dokumentation

Publiziere Package und API

Stelle API Doku bereit

Erstelle Tag in Source Control

## Valiant Product (13 New Lila Sets) (1.3.15)

Download OpenAPI specification: [Download](#)

Die bestehenden Lila Sets wurden überarbeitet und sollen neu als geführter Prozess über den MAP eröffnet werden können.

Created at 13/03/2023, 08:26

[Check Project on Cawemo](#)

### ► Development

Created with:

- [camundala-api v0.14.9](#)
- [mpa-api v0.14.0](#)
- [mpa-helper v0.15.0](#)

## Changelog

► CHANGELOG.md

● Init new Version 1.4.0-SNAPSHOT	 origin & develop	Pascal Mengelt
● Released Version 1.3.15	 origin & master	Pascal Mengelt
● MAP-6560: Added minimal Simulation for ProductOrderEBankingContractSimulation.		Pascal Mengelt
● MAP-6560: Added additional inputs / changed to allPoas in product-orderEBankingContract.bpmn		Pascal Mengelt
● MAP-6831: Changed ageLessEqual28 to ageLessEqual27.		Pascal Mengelt

# Deploy ment

Postman

Lokal

Bankenumgebung

The screenshot displays the Postman interface. The top section shows a REST client with a collection named 'deploy\_manifest'. The selected endpoint is 'HEAD Manage Deploy: TST Va...'. The test script below contains several assertions for version numbers:

```
11 // Project
12 valiantOnboardingVersion = "1.5.4" // new
13 valiantAddresschangeVersion = "3.11.1" // new
14 valiantDepotVersion = "1.14.2" // new
15 valiantKubeVersion = "1.4.2" // new
16 valiantAdminVersion = "1.2.4" // new
17
18 // Project / general Valiant
19 valiantProductVersion = "1.3.17" // new
20 valiantNnkVersion = "1.14.0" // new
21 valiantCmsVersion = "1.8.5" // new
22 valiantVollmachtVersion = "1.3.0" // new
23
24
25 // general Valiant
```

The bottom section shows the 'Valiant - Run results' for a test run on 8 Mar, 2023 at 08:36:12. The runner used is 'Local Docker Env'. The test results table is as follows:

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Local Docker Env	1	17s 677ms	4	1724 ms

Summary: All Tests Passed (4), Failed (0), Skipped (0). The detailed test results for Iteration 1 show a successful POST request to 'FSSO Token' with a status of 200 OK, a response time of 108 ms, and a body size of 2.676 KB.

# Releasing

Monatlich

TST > PRD

Dokumentation

Abhängigkeiten

Change Log

The screenshot shows a web application interface for 'Valiant Process Documentation'. At the top left, it displays 'Version 2023-02' with a dropdown arrow and a home icon. The main content area is divided into a left sidebar and a main panel. The sidebar contains a menu with 'Valiant Process Documentation' (highlighted), 'Dependencies Overview', 'Process Pattern', 'Process Statistics', and 'Contact'. Below this, there are sections for 'RELEASES' (listing releases from 2023-02 to 2022-08) and 'DEPENDENCIES' (listing various packages like bpf-helper, finnova-fil-is, etc.). The main panel features the title 'Valiant Process Documentation' in red, followed by the text 'Updated for Release 2023-02 - created on 09.02.2023'. Below this is a section titled 'BPF Packages' containing a complex dependency diagram. The diagram shows a hierarchy of packages: 'valiant-projects' at the top, which includes 'valiant-embedding', 'valiant-hube', 'valiant-addresschange', 'valiant-admin', and 'valiant-depot'. Below 'valiant-projects' are 'valiant-general-projects' (containing 'valiant-product' and 'valiant-cms') and 'valiant-general' (containing 'valiant-documents', 'valiant-bpmn', 'valiant-graviton', and 'valiant-services'). At the bottom left is the 'finnova' group, including 'finnova-fil-codes', 'finnova-bis-is', and 'bpf-helper'. Arrows indicate dependencies between these packages.



04

# Sonst noch

Was wir so lernten – Kontakt – Fragen

# Lessons learned



## Zusammen

Zusammen geht es schneller



## Testen

Automatisierte Tests - Simulationen sparen Zeit



## Monitoring

Camunda Ressourcen überwachen



## Namen

Namenskonventionen machen es einfacher



## Dokumentieren

Dokumentation kann Spass machen



## Zeit

Verbesserungen - Austausch - Spass

# Weitere Informationen

## **Agile Factory** – Ein Zusammenarbeitsmodell in der Praxis

- <https://www.weevolve.ch>



## **Dmn Tester** – Testing DMN Tables Automatically

- <https://page.camunda.com/ccs-mengelt-testing-dmn-tables-automatically>
- <https://github.com/camunda-community-hub/camunda-dmn-tester>



## **Camundala** – a new Way to develop Camunda BPMNs

- <https://page.camunda.com/ccs2022-domaindrivenprocessdevelopment>
- <https://pme123.github.io/camundala/>





# Danke!

## Fragen?

Christian Pfister, Valiant

- [christian.pfister@valiant.ch](mailto:christian.pfister@valiant.ch)

Karin Trachsel, WeEvolve

- [karin.trachsel@weevolve.ch](mailto:karin.trachsel@weevolve.ch)
- [LinkedIn: karin-trachsel](#)

Pascal Mengelt, Finnova

- [Pascal.Mengelt@finnova.com](mailto:Pascal.Mengelt@finnova.com)
- [pme123 > Github / Camunda Forum](#)

CREDITS: This presentation template was created by [Slidesgo](#), including icons by [Flaticon](#), infographics & images by [Freepik](#)

**valiant**



We**E**volve

**finnova**.



**swisscom**

