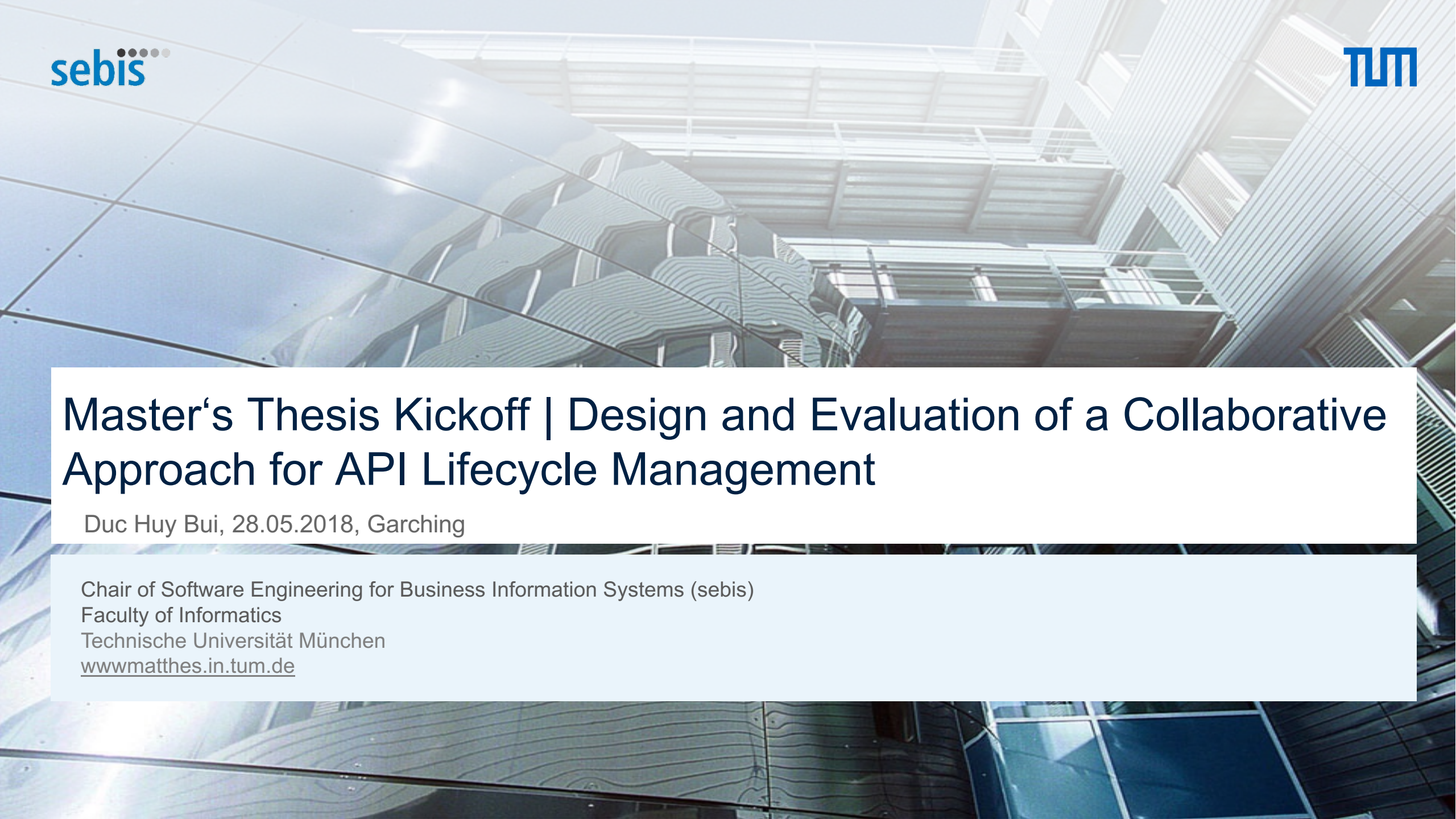


# Master‘s Thesis Kickoff | Design and Evaluation of a Collaborative Approach for API Lifecycle Management

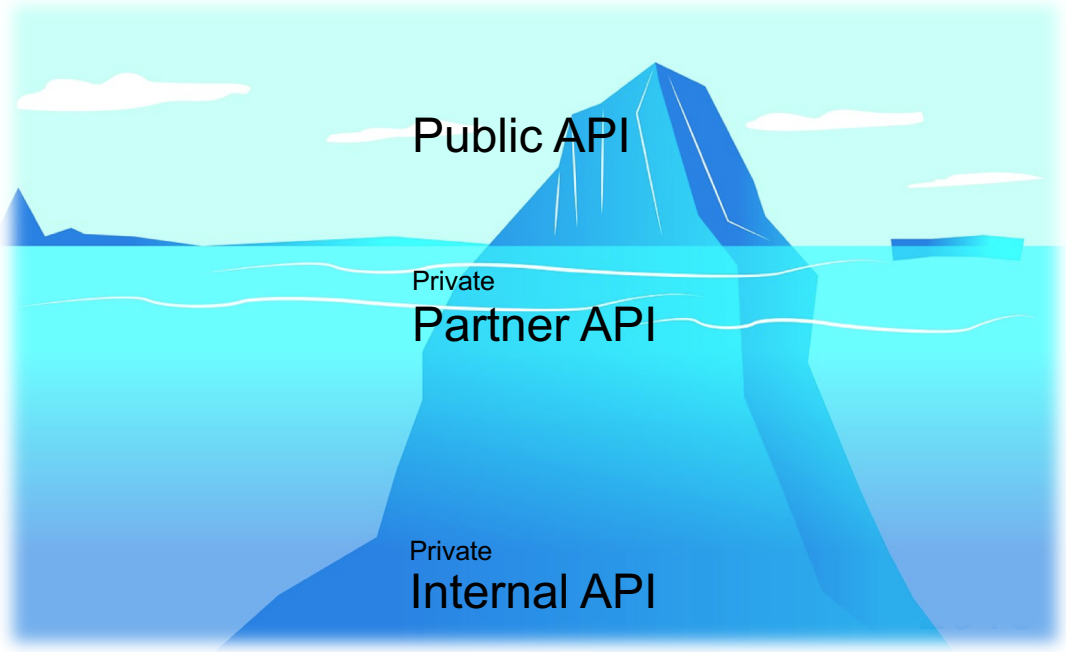
Duc Huy Bui, 28.05.2018, Garching

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

# Outline

- Motivation
- Research Questions and Approach
- First Results
  - Collaborative API Lifecycle Management
  - Use Cases
  - System Design
- Next Steps
- Timeline

# Motivation
## Web APIs become products




- API Economy
  - API becomes a **product**
  - **Innovation driver** for digital businesses (Cloud, Big Data, IoT, etc.)
  - **Competitive advantages**, e.g. Salesforce (50% revenue), Ebay (60% revenue), Expedia (90% revenue) [4]
- API Management (“Game Changer” [6]) including API Lifecycle Management

# Motivation
## Problem Statement and Solution Approach

Problem Statement

- The **API lifecycle in enterprises is not an integrated process** which leads to inefficient processes, manual operative overhead, longer time to market, low API customer satisfaction and API adoption

Solution Approach

- Integrate the API lifecycle process in a **central portal** and use an **automated workflow** to efficiently improve shortcomings by **guiding API providers and API consumers through the API lifecycle process**

# Research Questions and Approach

**RQ1**

What **phases, activities, artifacts and roles** belong to an API Lifecycle and how could a holistic approach for an integrated **API Lifecycle** look like that is **driven by collaboration of participating stakeholders**?

- Literature Review
- Industry Practices
- Expert Interviews

- Requirements
- Conceptual Model

**RQ2**

How can **tools** be used to **support activities** and **automate** parts of the **collaborative API Lifecycle Management**?

- Formal Specification

- Use Cases
- System Design
- Workflow (BPMN)

**RQ3**

How well do **tools support specific use cases** in the API Lifecycle?

- Code Implementation
- Expert Interviews

- Prototype
- Evaluation Results

# Collaborative API Lifecycle Management (CALM)
## Success Factors as Requirements

- Sources for success factors (8)
  - API Management/Lifecycle (industry (2), academic papers (1))
  - Product Development (2)
  - (Enterprise-/IT-) Service Management (2)
  - Agile Software Development (1)

- Structure success factors into categories
  - Business
  - Organizational
  - Process
  - Technical

- Result: Consolidated table of requirements for API Lifecycle

- Example requirement: “Design for UX/DX” or “Top Management Support”

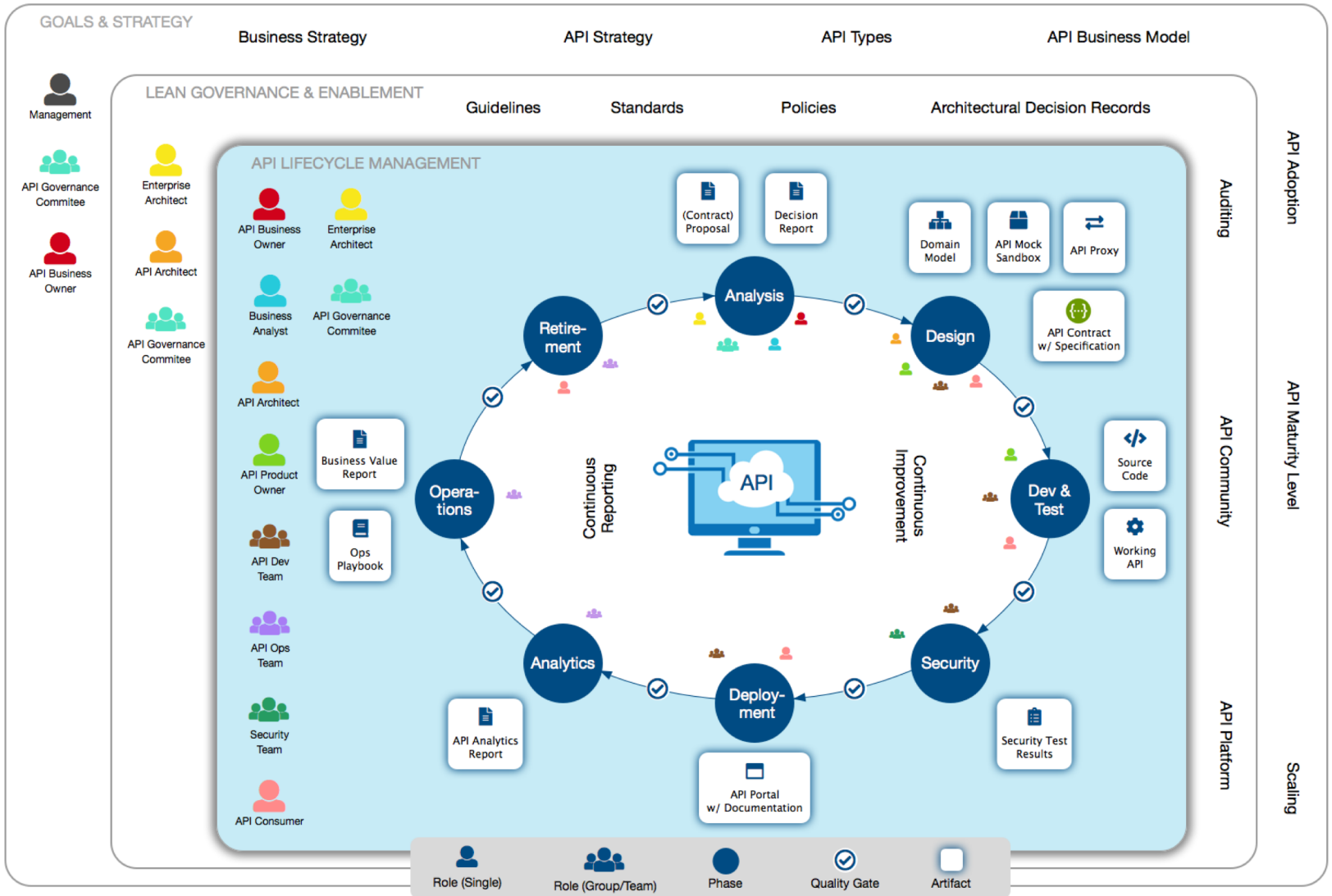
# Collaborative API Lifecycle Management (CALM)

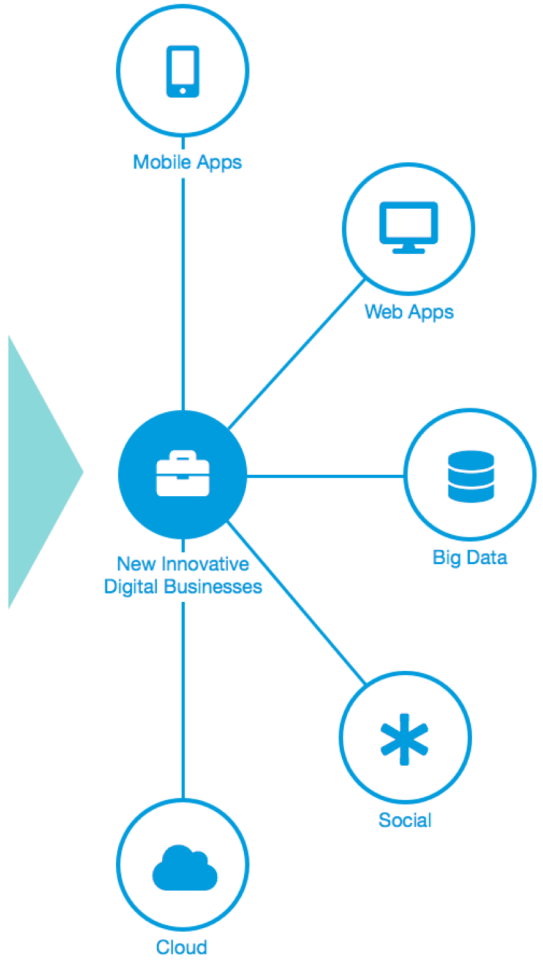
## Conceptual Model

External User

Partner

Internal Employee

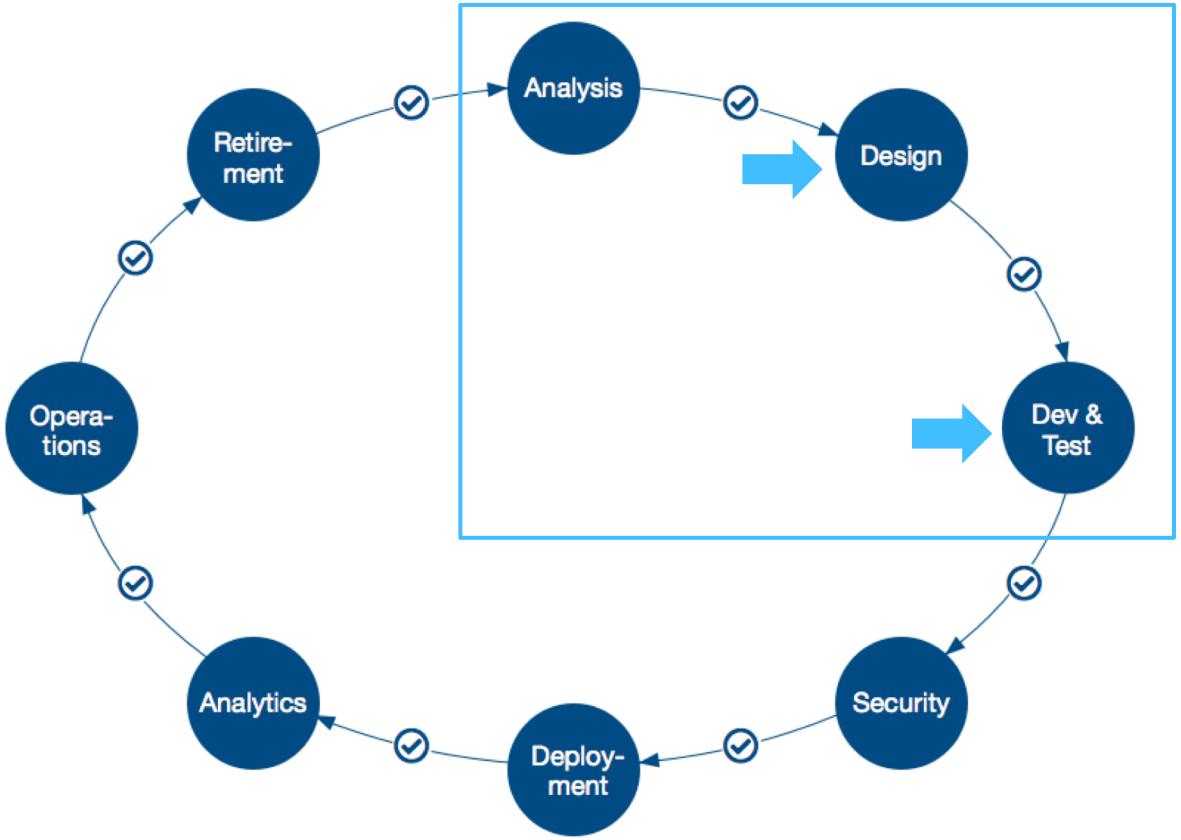


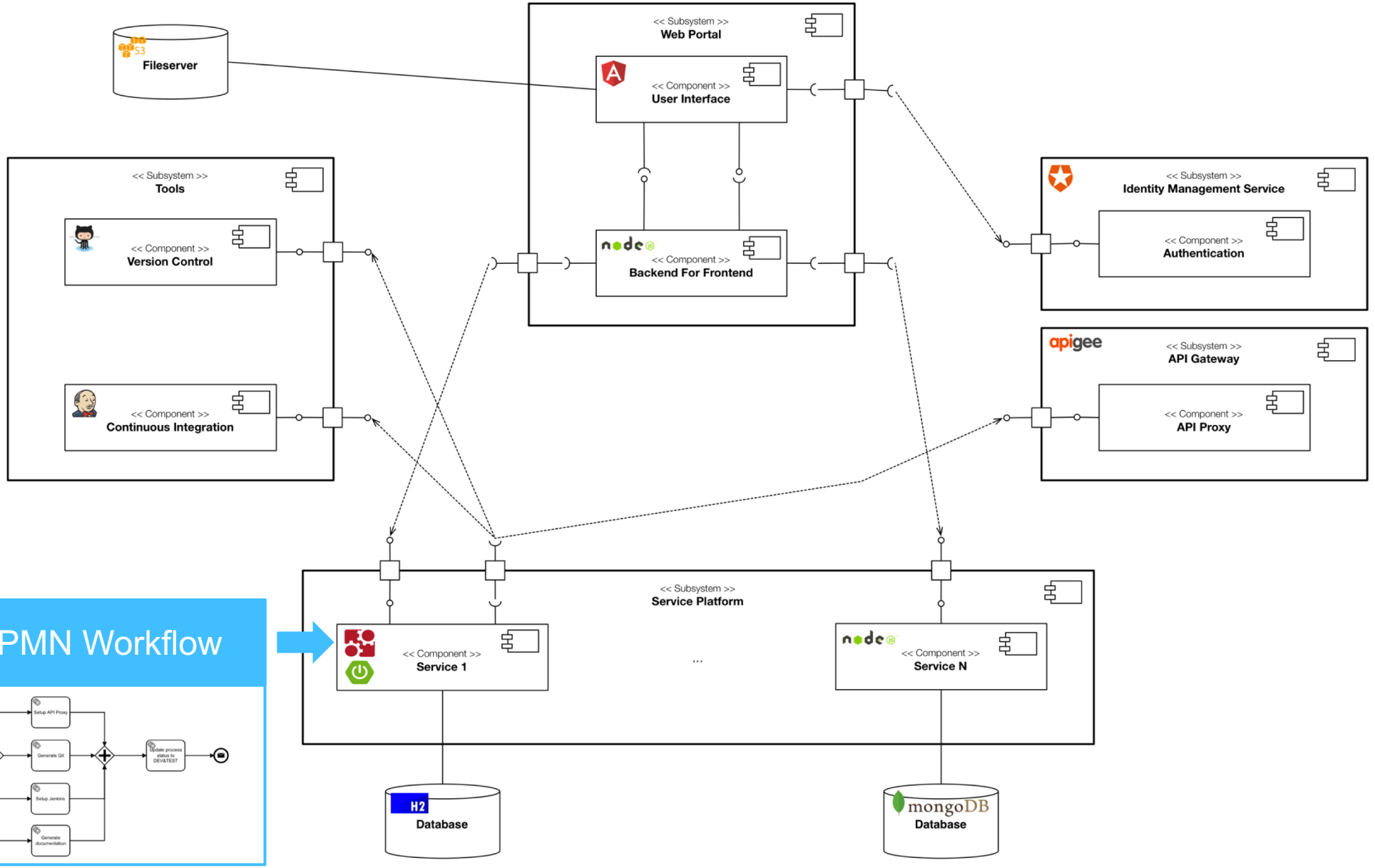
# Use Cases
## CALM Web Portal

**KEY FUNCTIONALITIES**

1. Project Management
2. Proposal Management
3. API Subscription Management
4. Governance Management
5. DevOps Management
6. Security
7. Analytics
8. Version Management

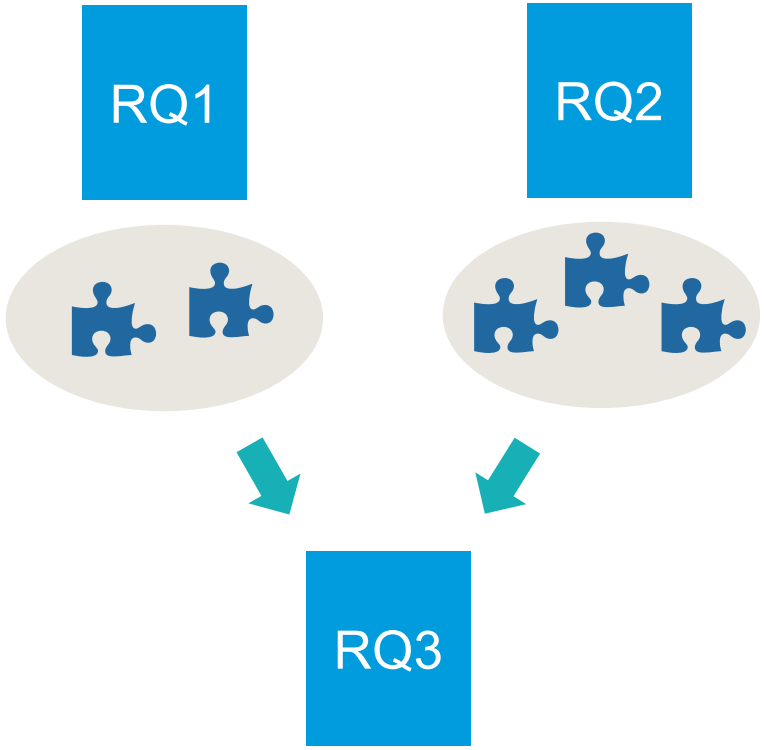


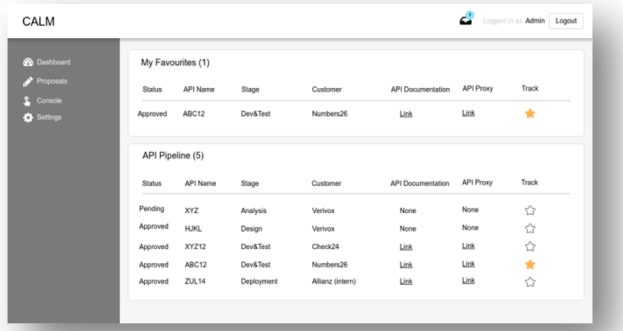
# System Design
## Prototype

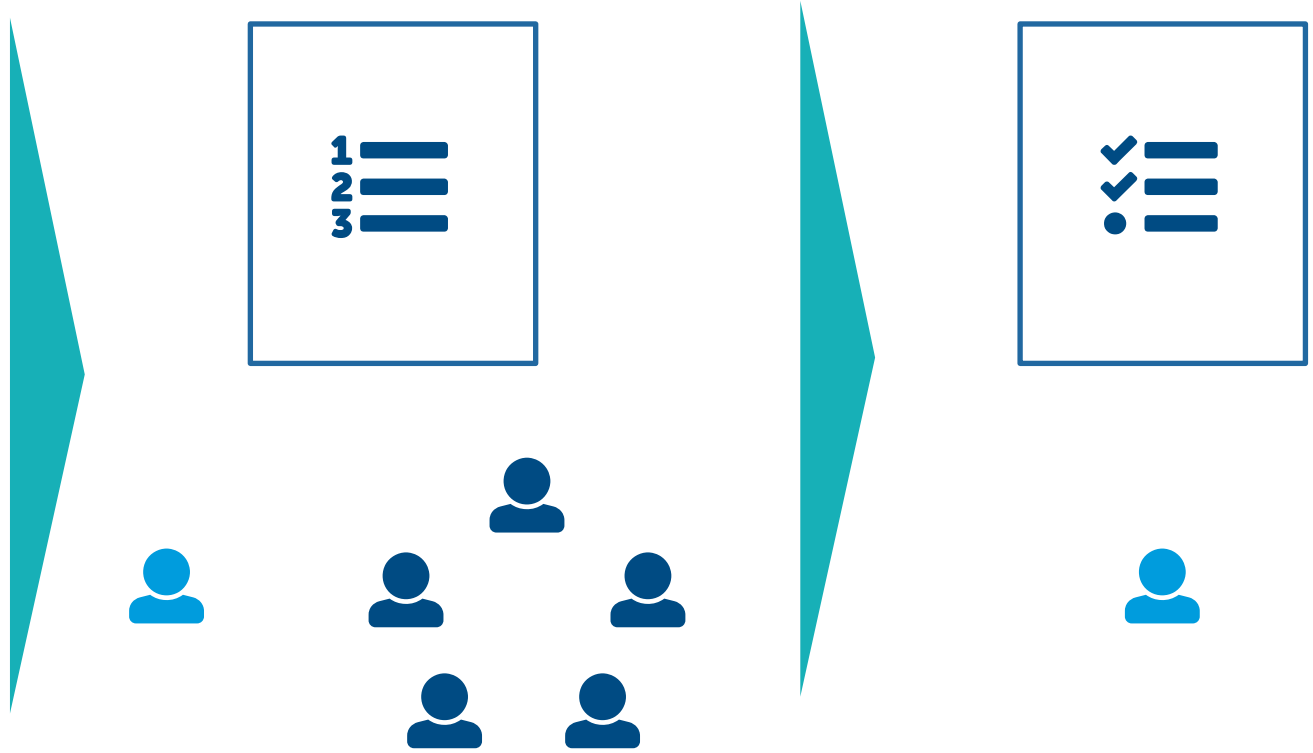


# Next Steps

## Implementation and Evaluation Approach



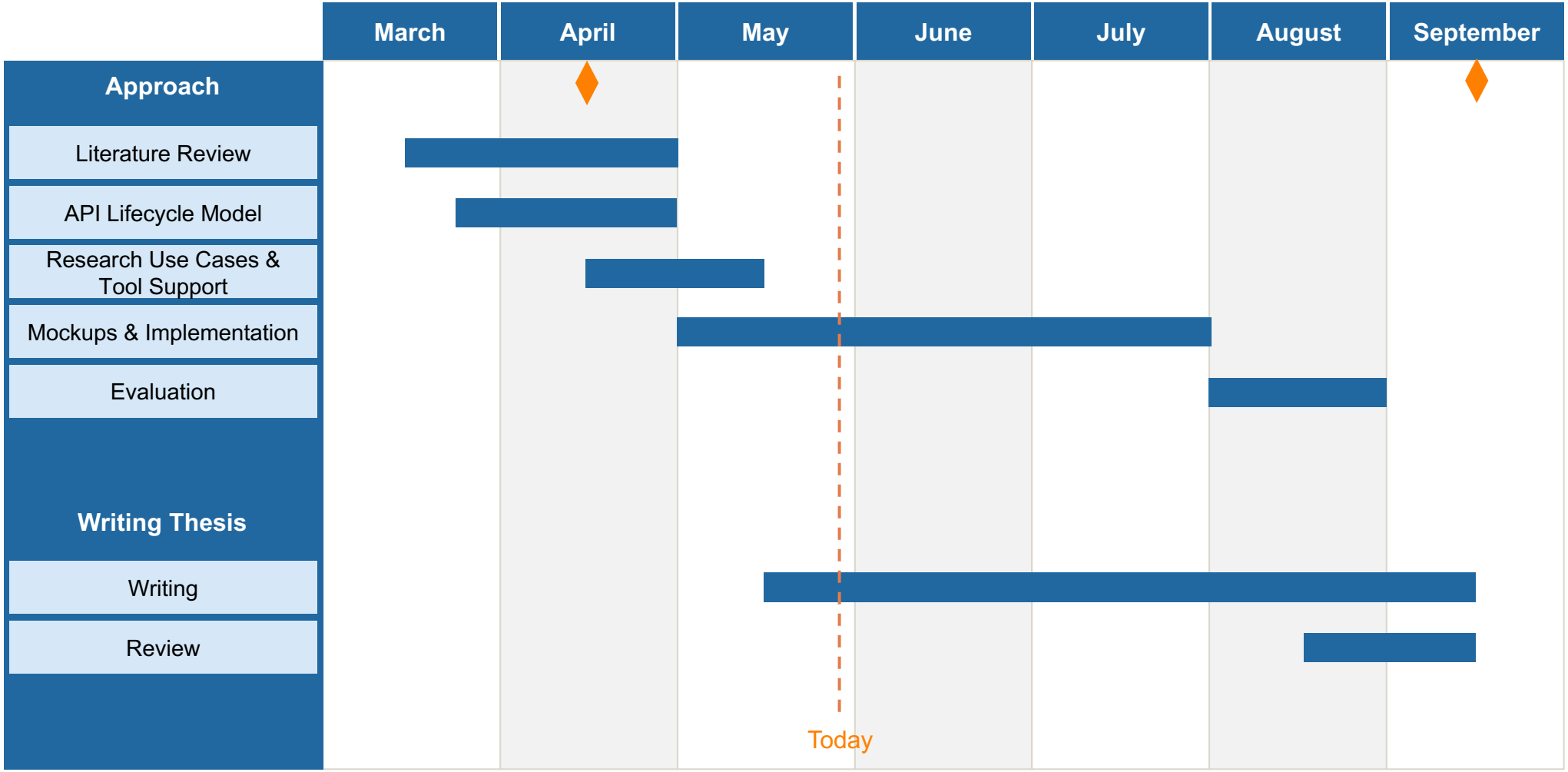




Expert Interviews

Evaluation Results

# Timeline




Registered Date: 15.04.2018

Submission Date: 15.09.2018

Thank you for your attention! ☺


B.Sc.

**Duc Huy Bui**

Technische Universität München
Faculty of Informatics
Chair of Software Engineering for Business Information Systems

Boltzmannstraße 3
85748 Garching bei München

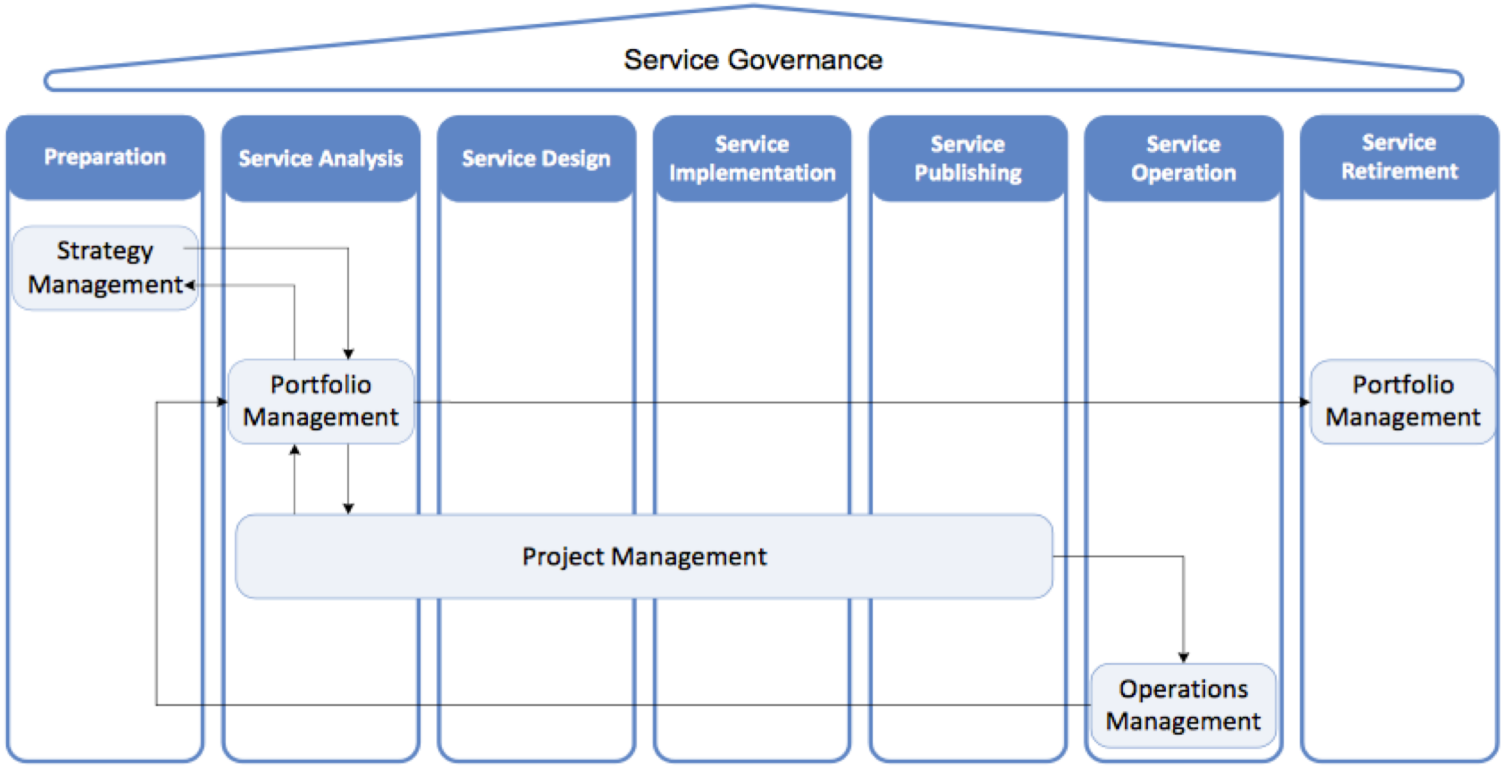
Tel +49.89.289.
Fax +49.89.289.17136

duchuy.bui@tum.de
wwwmatthes.in.tum.de

# Backup

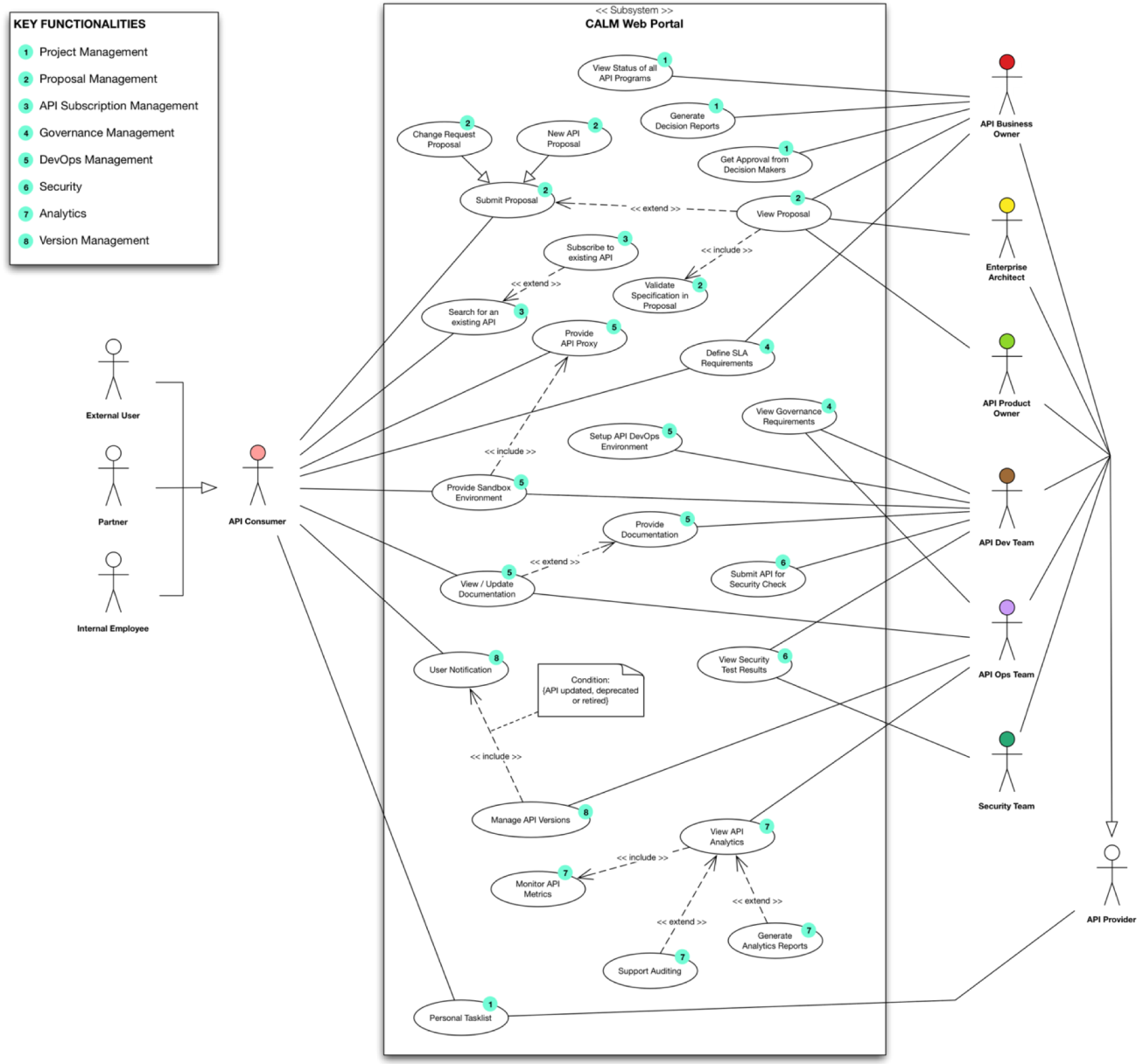
# Service Lifecycle vs API Lifecycle

Both lifecycles look similar, but API Lifecycle has API specific activities and phases like Analytics or Security that need to be highlighted.




Source: Kohlborn, T., Korthaus, A., & Rosemann, M. (2009). Business and Software Service Lifecycle Management (pp. 87–96). IEEE. https://doi.org/10.1109/EDOC.2009.20

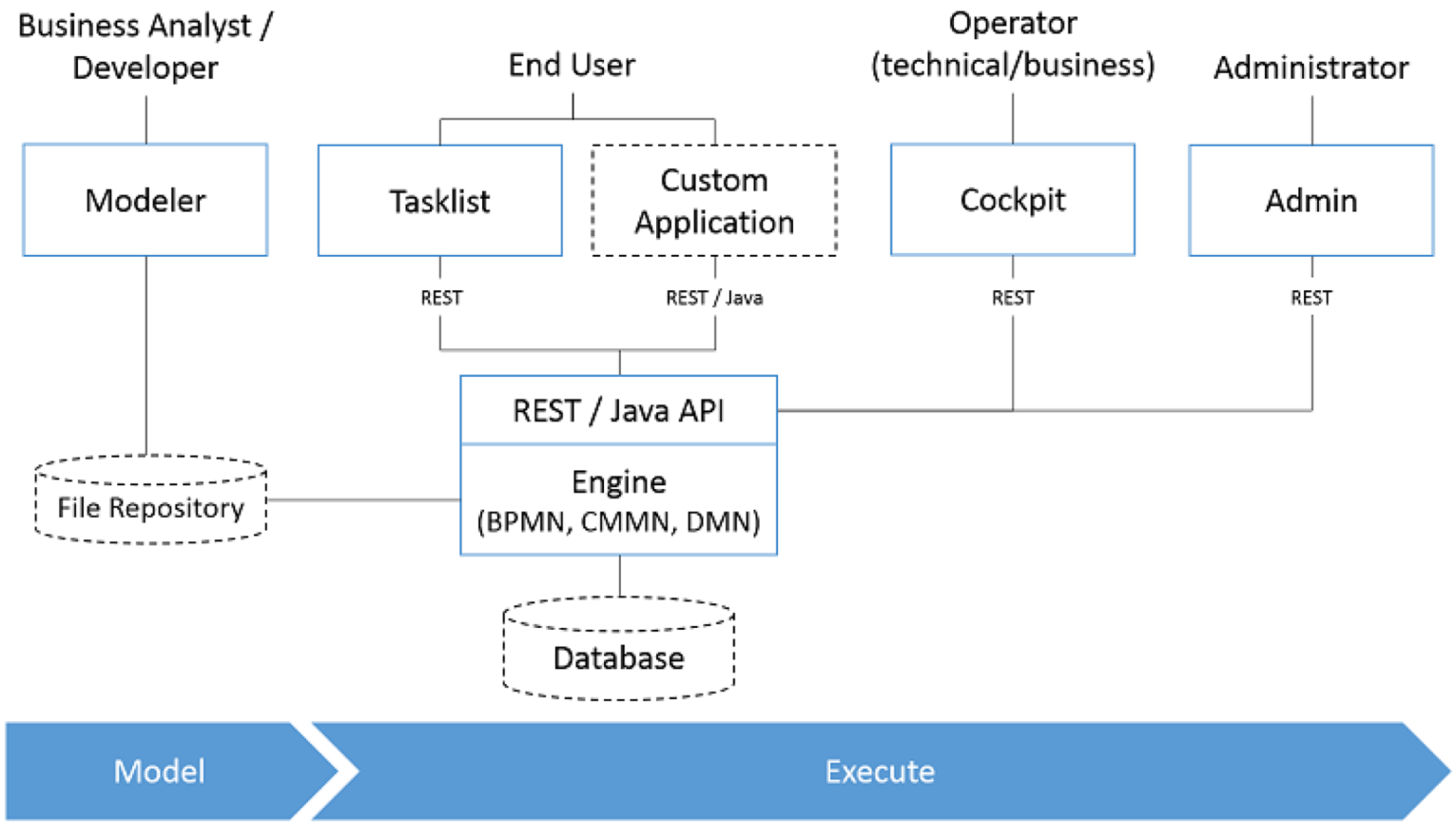
# Use Case Diagram



# Camunda Process Engine

camunda

is an open source platform for workflow and decision automation.



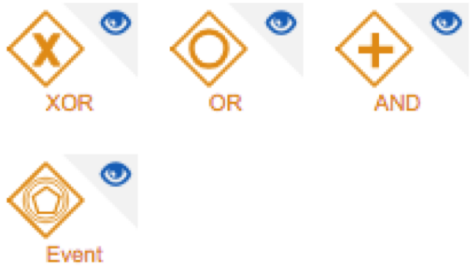

Source: https://camunda.com/

# BPM Diagram

## Basics

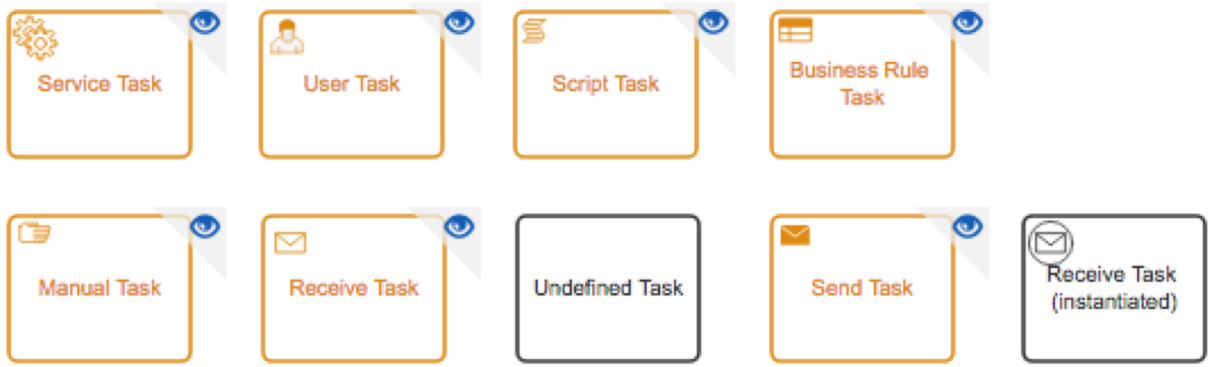
### Participants


### Gateways




### Subprocesses


### Tasks



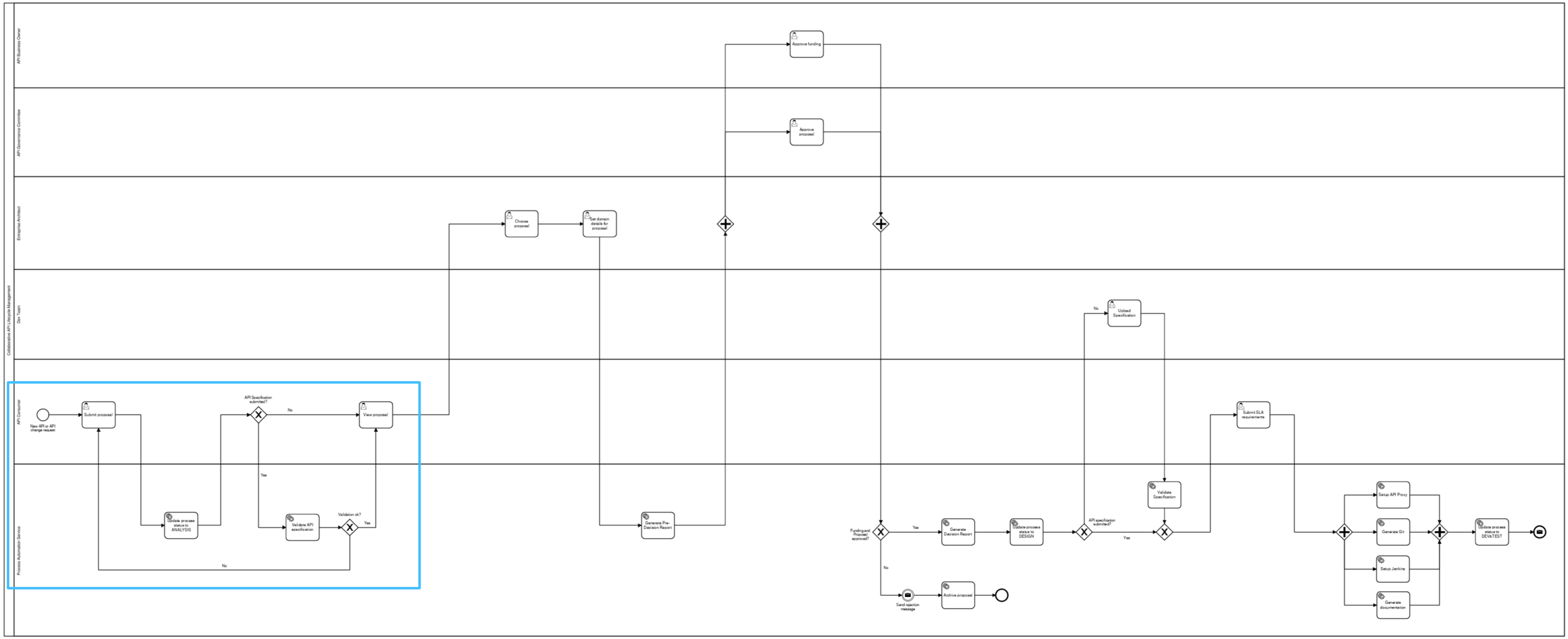

### Events

| Type | Start | | | Intermediate | | | | End |
|---|---|---|---|---|---|---|---|---|
| | Normal | Event Subprocess | Event Subprocess non-interrupt | catch | boundary | boundary non-interrupt | throw | |
| None | ○ | | | | | | ◎ | ○ |
| Message | ✉ | ✉ | ✉ | ✉ | ✉ | ✉ | ✉ | ✉ |
| Timer | 🕐 | 🕐 | 🕐 | 🕐 | 🕐 | 🕐 | | |
| Conditional | ≡ | ≡ | ≡ | ≡ | ≡ | ≡ | | |
| Link | | | | ⇨ | | | ⇨ | |
| Signal | △ | △ | △ | △ | △ | △ | ▲ | ▲ |
| Error | | ⚡ | | | ⚡ | | | ⚡ |
| Escalation | | Ⓐ | Ⓐ | | Ⓐ | Ⓐ | Ⓐ | Ⓐ |
| Termination | | | | | | | | ● |
| Compensation | | ⏪ | | | ⏪ | | ⏪ | ⏪ |
| Cancel | | | | | ⊗ | | | ⊗ |

Source: https://docs.camunda.org/manual/7.8/reference/bpmn20/

# BPM Diagram

## Prototype Workflow



# BPM Diagram
## Example Detail