Avoiding Redundancy in the Management of Technical Documentation and Models: Requirements Analysis and Prototypical Implementation for Enterprise Architecture Management
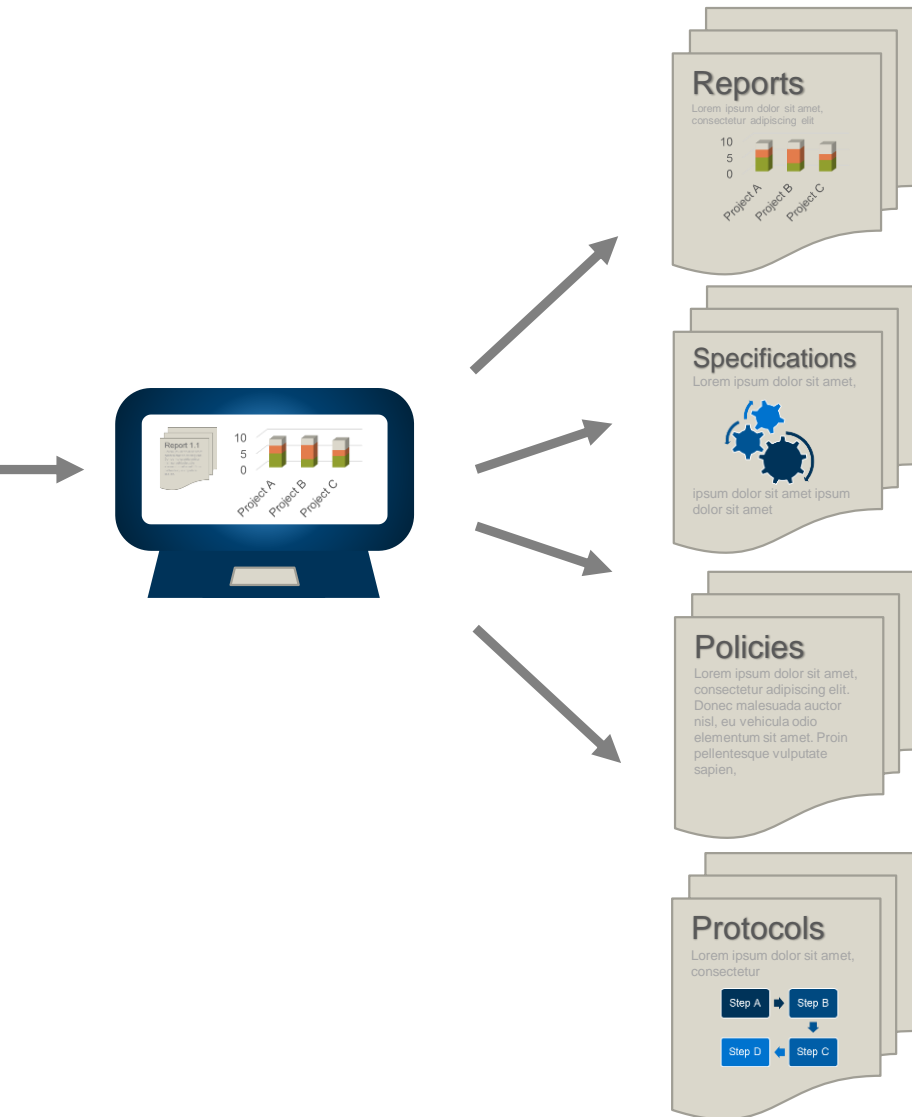
September 19th
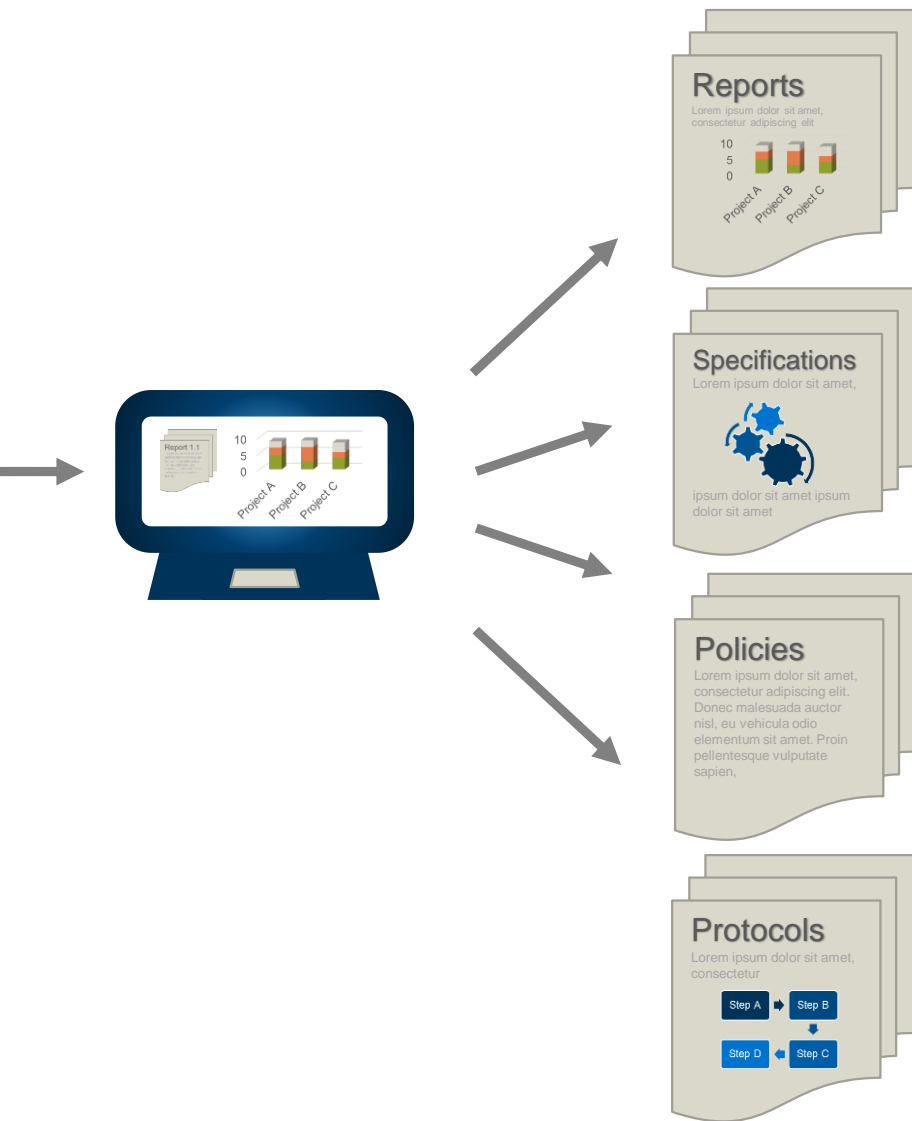
Peter Velten – Master's Thesis – Final Presentation

Software Engineering for Business Information Systems (sebis)
Department of Informatics
Technische Universität München, Germany

wwwmatthes.in.tum.de

Agenda

sebis

sebis

Reports
Lorem ipsum dolor sit amet, consectetur adipiscing elit

Specifications
Lorem ipsum dolor sit amet,

ipsum dolor sit amet ipsum dolor sit amet

Policies
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec malesuada auctor nisl, eu vehicula odio elementum sit amet. Proin pellentesque vulputate sapien,

Protocols
Lorem ipsum dolor sit amet, consectetur

Step A → Step B
Step D ← Step C

Reports
Lorem ipsum dolor sit amet, consectetur adipiscing elit

Specifications
Lorem ipsum dolor sit amet,

ipsum dolor sit amet ipsum dolor sit amet

Policies
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec malesuada auctor nisl, eu vehicula odio elementum sit amet. Proin pellentesque vulputate sapien,

Protocols
Lorem ipsum dolor sit amet, consectetur

Step A → Step B
Step D ← Step C

Present Problems

- **Redundancy leads to Inconsistencies**
- **Formatting issues**
- **No proper repository**
- **Manually processing**
- **Maintenance**

sebis

Maintenance of redundant information. Output and input of this information is laborious and error-prone.

Reports often created manually due to lack of technology. (e.g. using Word-templates). This is error-prone and cost-intensive. In addition immutability may be desired.

Existing solutions lacking in critical functionality, such as a suitable (graphical) output.

Demand in Enterprises for consistent and redundancy-free reporting and documentation tools

sebis

> **"** Avoiding Redundancy in the Management of Technical Documentation and Models: Requirements Analysis and Prototypical Implementation for Enterprise Architecture Management **"**
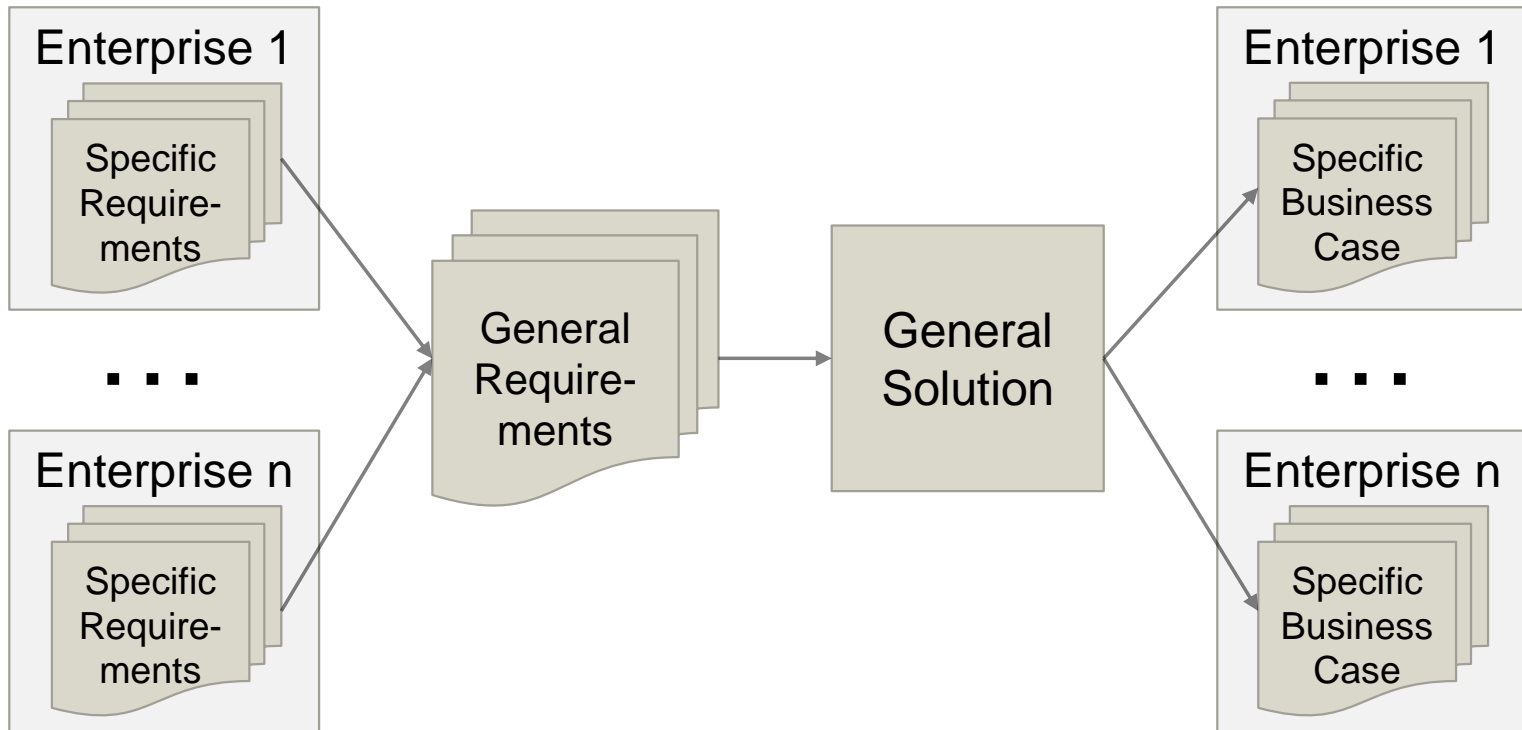
Inquiry and analysis of concrete requirements from industrial partners to a reporting tool

Abstraction of the requirements and conception of a general-purpose solution

Prototypical implementation of the solution approach with subsequent evaluation

sebis

ID	Requirement
R-1	Inclusion of simple parameters
R-2	Inclusion of lists
R-3	Combine text-parts to comprehensive documents
R-4	Support for PDF
R-5	Support for Office-formats
…	…
R-24	Serial generation of reports

Requirement	Priority
Reuse of sub-components	High
Usage of parameters and types	High
Interface to present data sources	High
Integration of formatting functions	High
Serial generation of documents	Medium
Integration of different file formats	Medium
Publishing workflow	Low
Document versioning mechanism	Low
Integration of visualizations	Low

sebis

| Template Generation | → | Configuration Generation | → | Report Generation |
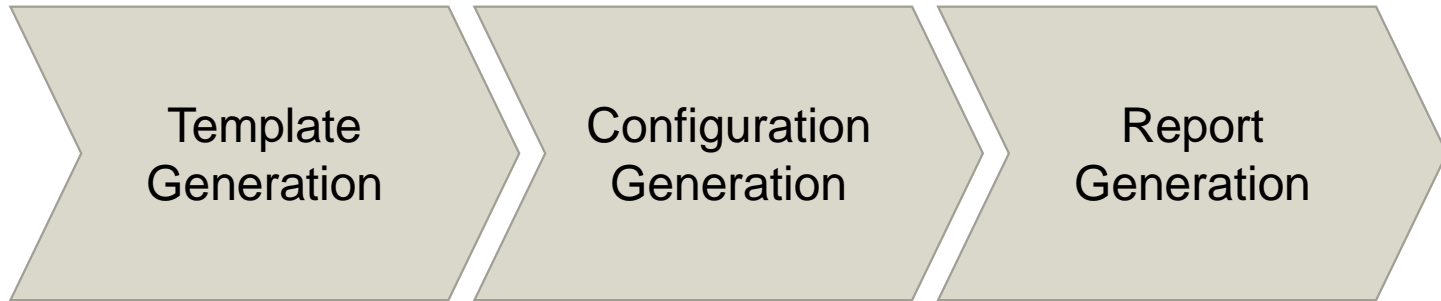
3-Step procedure to finally create the required document

Goal: Usage of desktop-based tools in combination with the reporting tool to avoid problem areas

Every step corresponds to a task in the application

SocioCortex as repository: Each step creates an entity within SocioCortex

Template and Configuration is only done once. End user only creates final Report

sebis

| Template Generation | Configuration Generation | Report Generation |
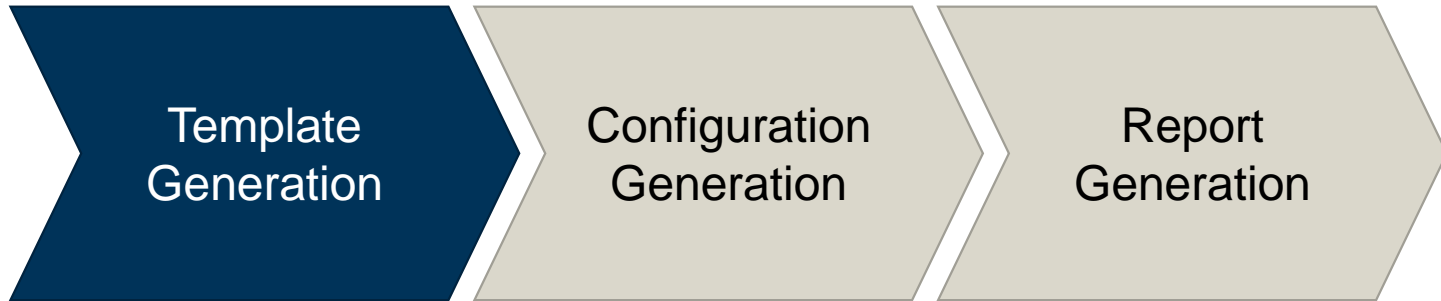
Template as basis for further steps

Creation of a template-file within desktop-tools of Microsoft Office (.docx / .pptx)

Define parameters as placeholders instead of concrete content from SocioCortex

Velocity as template-language allows further functionalities like conditions and loops

Template-file gets uploaded within the application for further processing

| Template Generation | Configuration Generation | Report Generation |
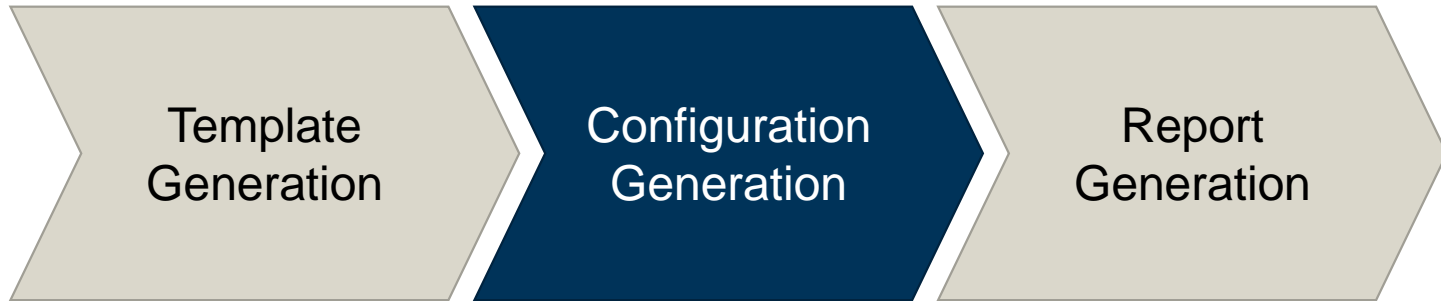
A configuration is based on a created template

Define MxL-queries for each parameters from the template

Queries are only validated, but not executed

Multiple Configurations for one Template

Example: Template for technical documentations of applications. One Configuration of strategic applications, another for operational applications

sebis

| Template Generation | Configuration Generation | Report Generation |
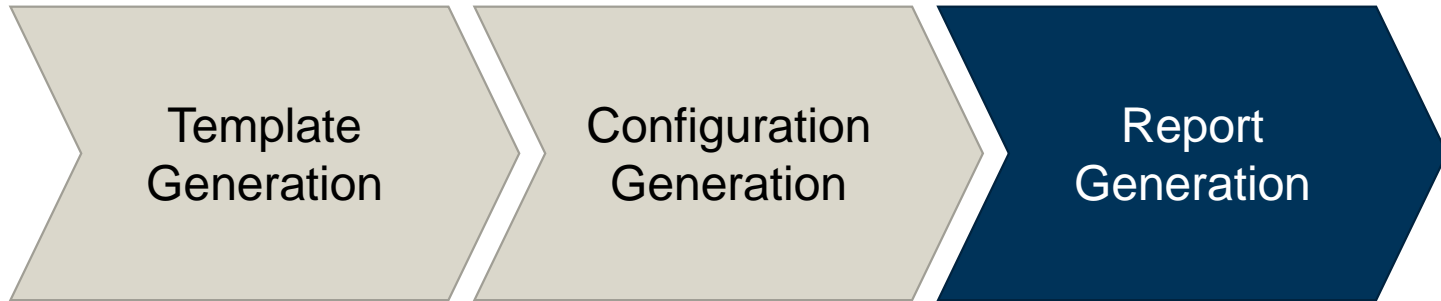
Execution of MxL-queries

Replace parameters with retrieved results and provide final documents to end-user
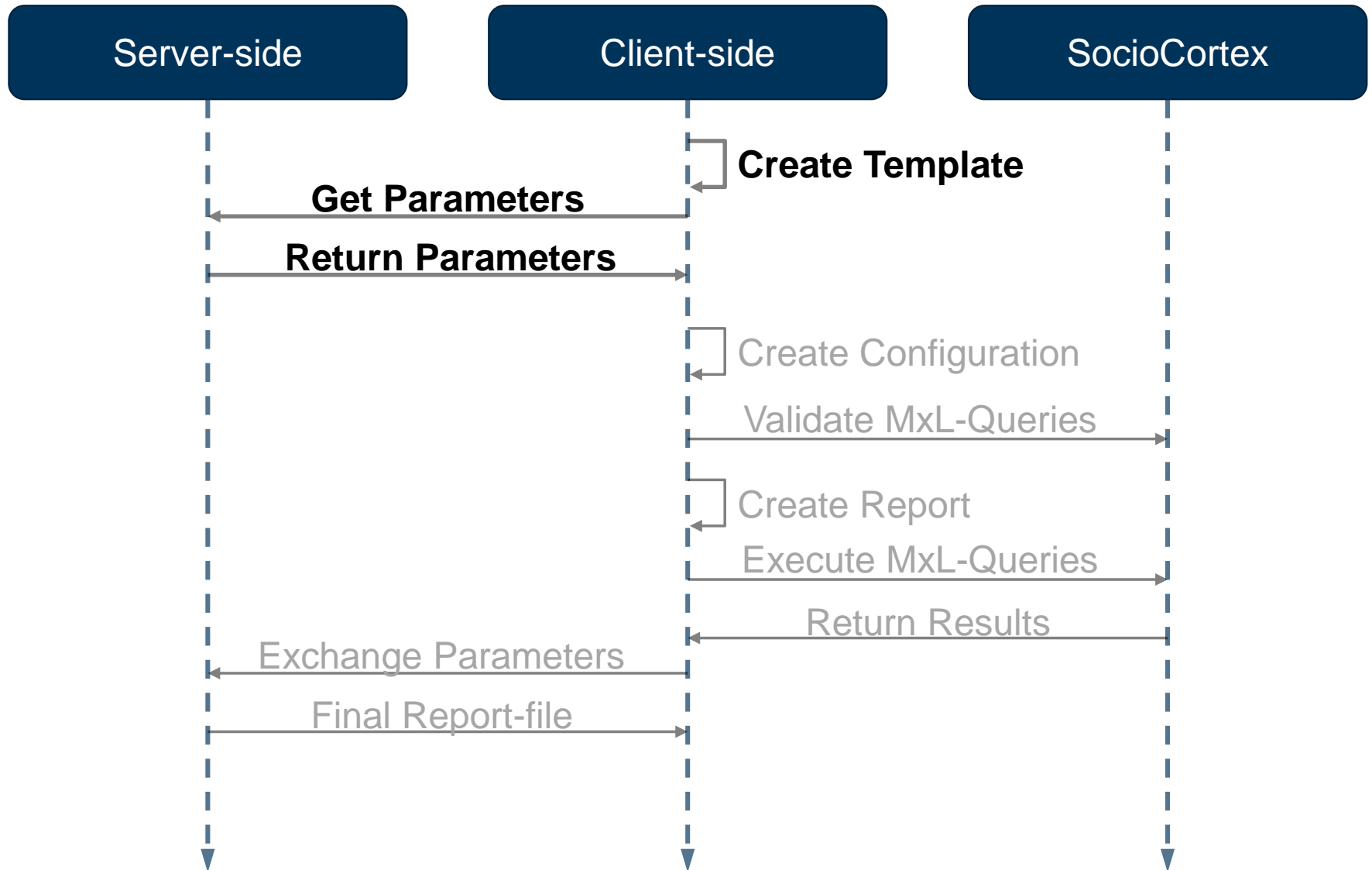
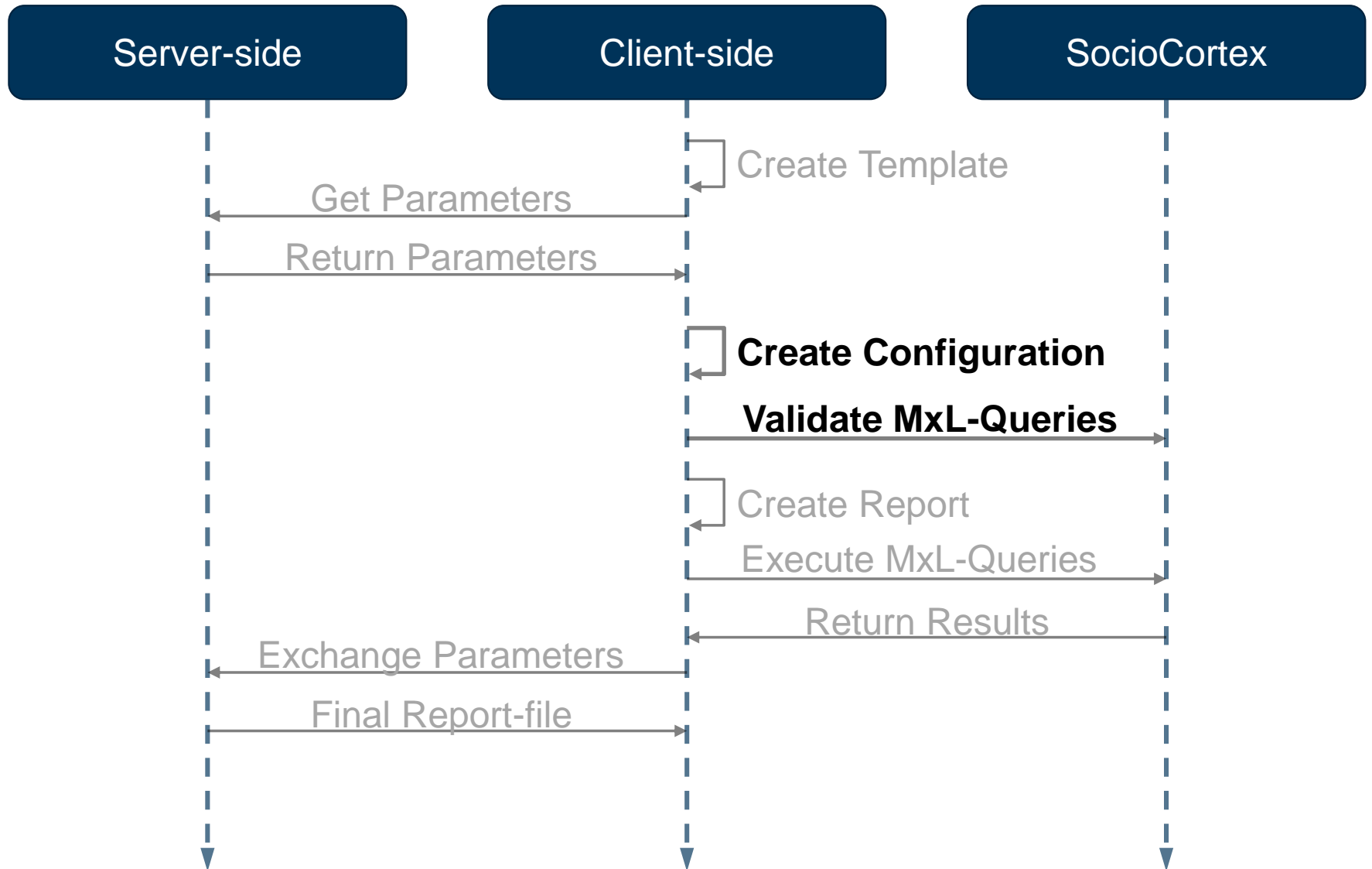Report always created with up-to-date data

Conversion from .docx to .pdf

Multiple Reports for one Configuration

sebis

Server-side	Client-side	SocioCortex

Create Template

Get Parameters

Return Parameters

Create Configuration

Validate MxL-Queries

Create Report

Execute MxL-Queries

Return Results

Exchange Parameters

Final Report-file

```
Server-side          Client-side           SocioCortex

                     ┌─ Create Template
        Get Parameters ◄─┘
        Return Parameters ──►

                     ┌─ Create Configuration
                     ◄─┘
                     Validate MxL-Queries ──►

                     ┌─ **Create Report**
                     ◄─┘
                     **Execute MxL-Queries** ──►
                     **Return Results** ◄──
    **Exchange Parameters** ◄──
    **Final Report-file** ──►
```

4. Evaluation

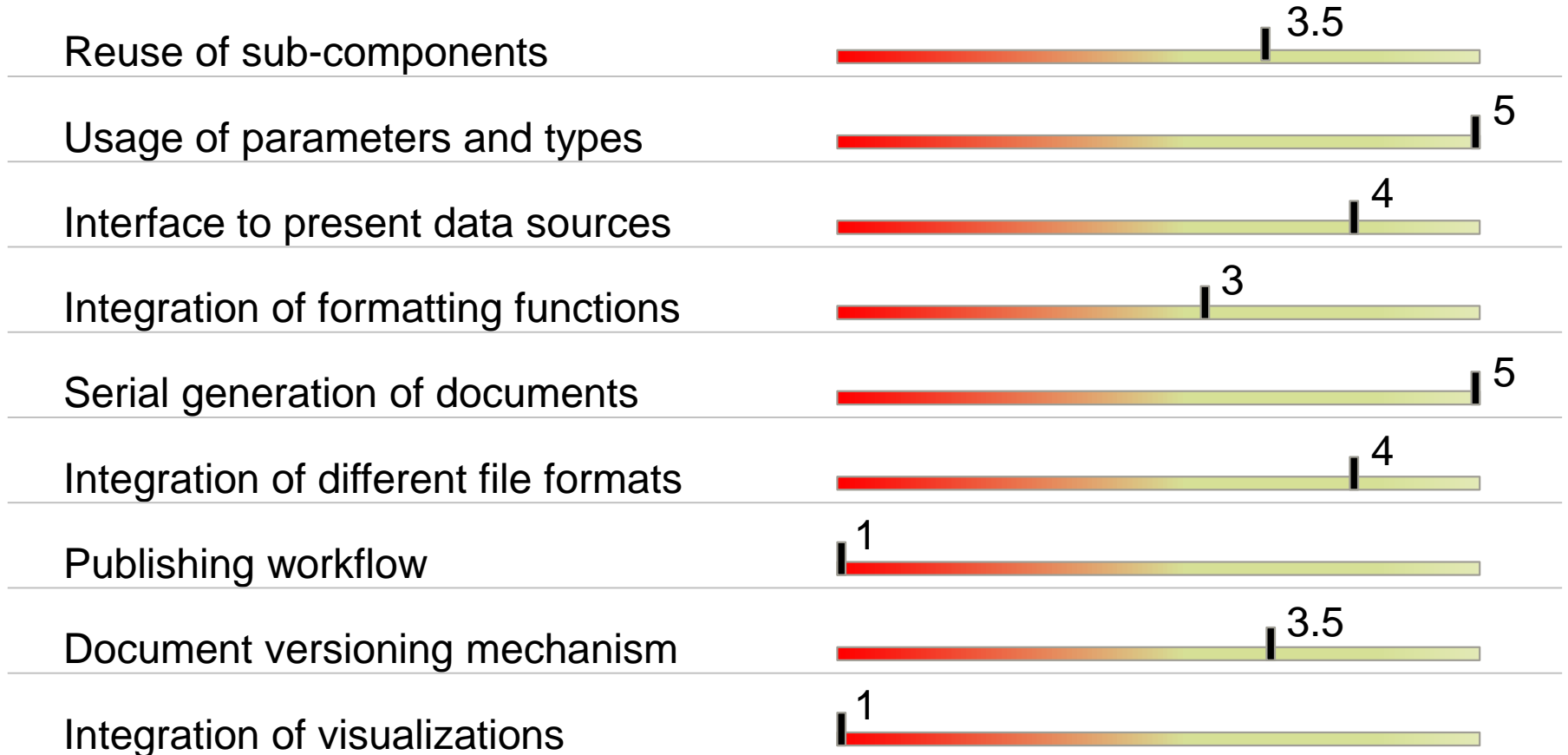| Scenario 1	Scenario 2
Creation of a management report by highlighting crucial applications and their relationships with each other	*Creation of technical documentations for each available application in the system*

Evaluation by demonstrating the scenarios to 2 interested companies

Assessment of each requirement by discussion on the basis of a questionnaire

Additional quantified assessment (scala of 1 to 5)

Reuse of sub-components	3.5
Usage of parameters and types	5
Interface to present data sources	4
Integration of formatting functions	3
Serial generation of documents	5
Integration of different file formats	4
Publishing workflow	1
Document versioning mechanism	3.5
Integration of visualizations	1

- Enhancement of versioning-mechanism
    - Version-numbers of documents
    - Usage of SocioCortex versioning-mechanism
- Support for additional file-formats (especially Excel)
- Support of visualizations
- Creation of more sophisticated templates
- Attachment of created documents to concrete entities
- Implementation of a publishing workflow

Still different aspects are missing for productive application

Overall the prototype and the approach was considered as very well and expedient

sebis

- Fulfillment of all requirements
  - Visualizations
  - Publishing workflow
- More elaborated integration within SocioCortex
  - Append templates to entityTypes ("Stylesheet")
  - Distribute the documents to the entities
    (User will always access up-to-date version)
  - Usage of SC-Visualizer

The prototype laid the foundation for further developments to prevent the problems of present EA Documentation

Integration within SocioCortex would enhance the powerfulness of the system and increase the usability of the end-user

Thank you for your attention!

Do you have any questions or comments?