Platform-Independent UI Models: Extraction from UI Prototypes and rendering as W3C Web Components

Marvin Aulenbacher, 19.06.2017, Munich

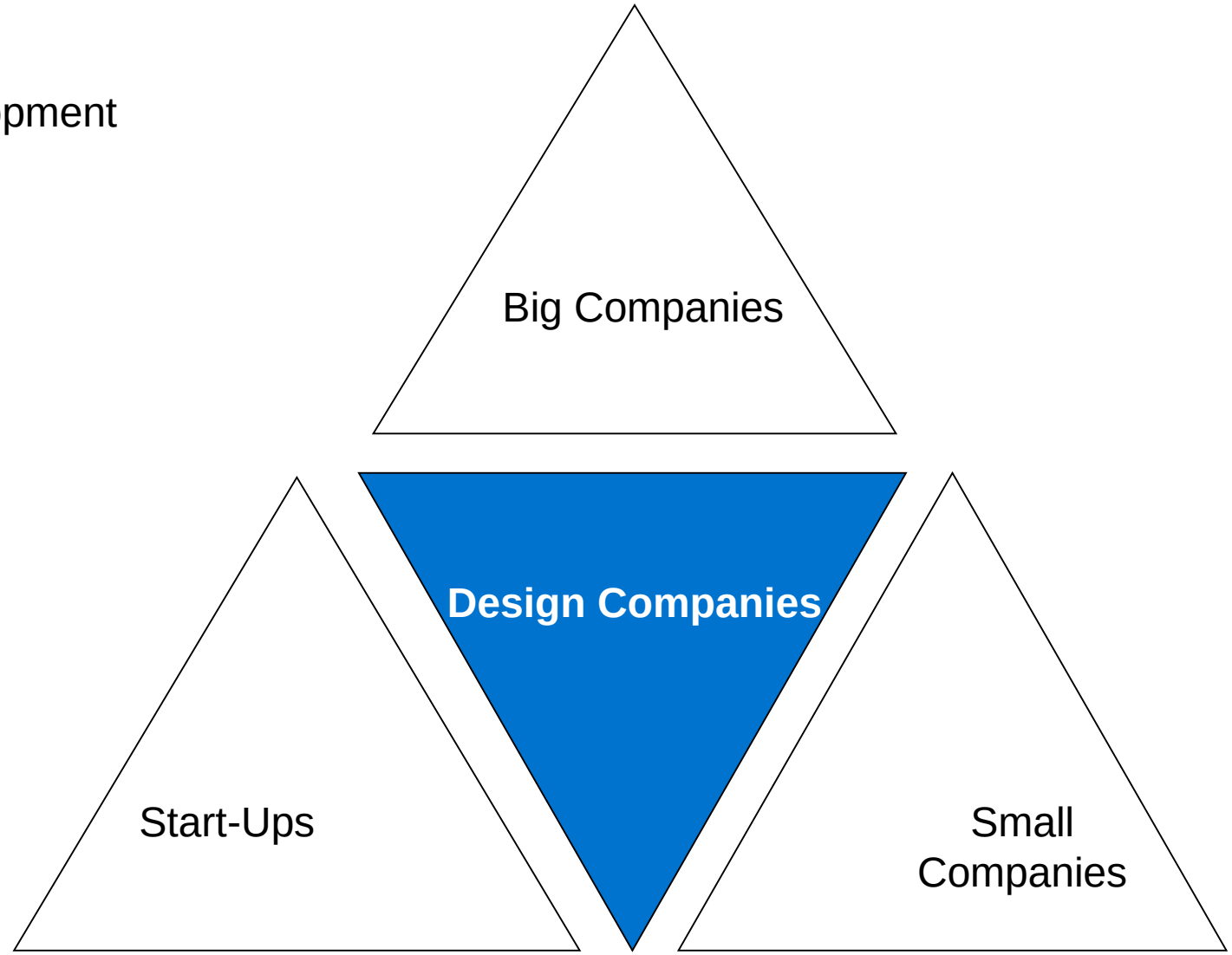
Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

Outline

1. Introduction
2. Idea
3. Proposed Process
4. Simple Example
5. Timeline

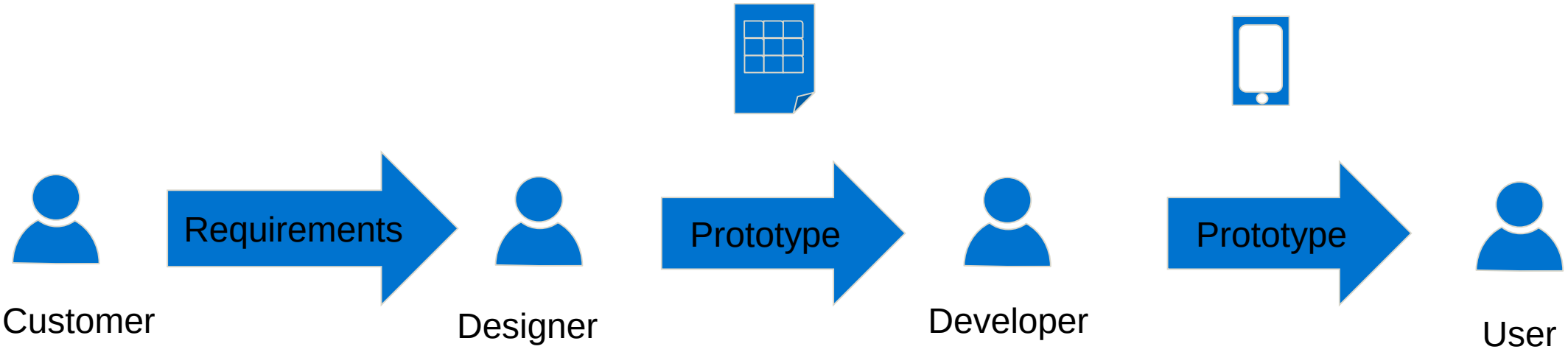
Context

Web Application Development



Domain Description

Prototype driven development in web application development



How should the UI look like?

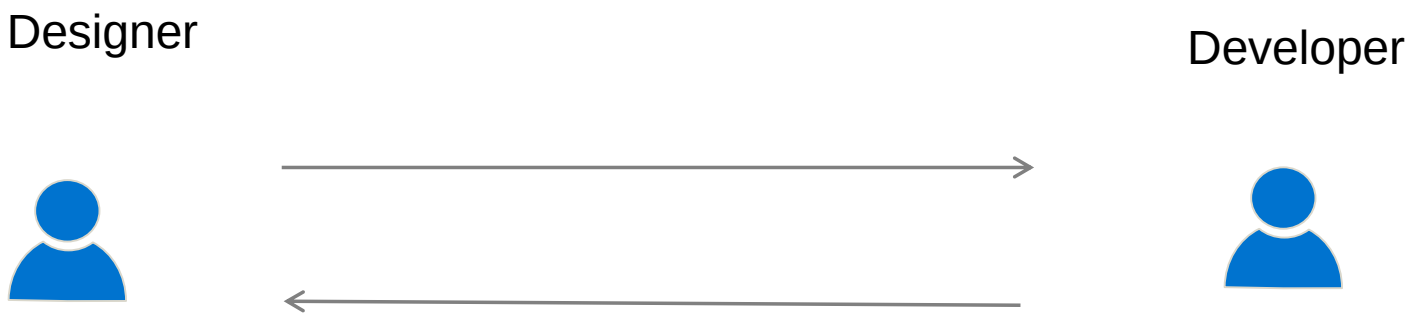
Creates low-fidelity Prototype

Creates high-fidelity Prototype

Refactoring?

Problem

Communication Issue between Designer and Developer



Do not share common domain specific language

The Idea

- Support the developer by generating web components from an UI prototype

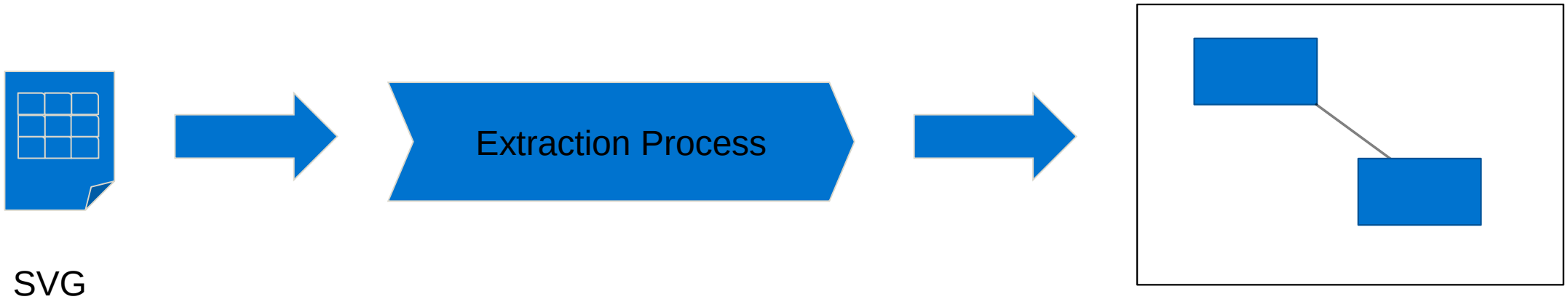
- Standardize in- and output file formats(SVG, JSON)

- Use view models as domain specific language

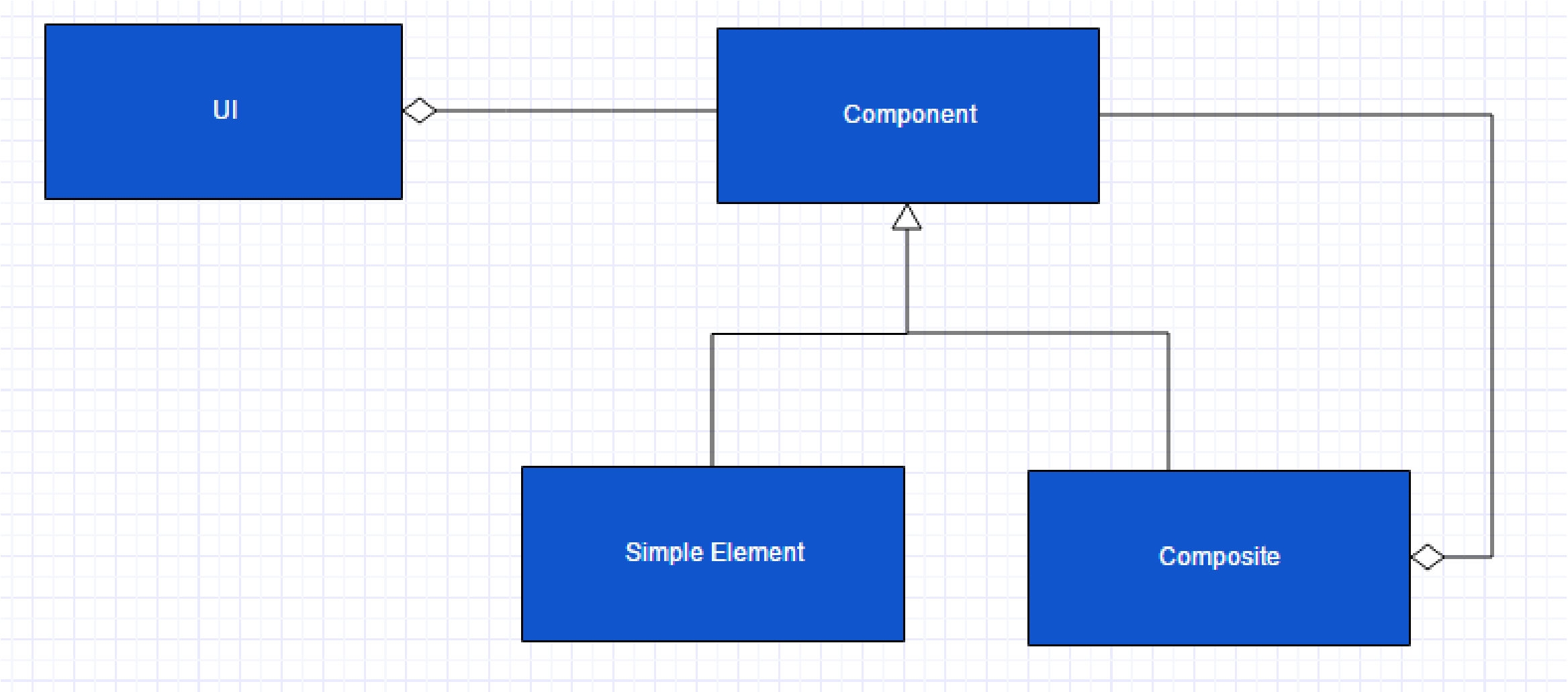
Research Questions

- Which design tools do designers use and why? (2 Tools)
- How should the architecture of a supporting process for the generation of web components be structured?
- How useful is the automation process based on three basic use cases for prototype driven development?

View Model Generation



UI View Model



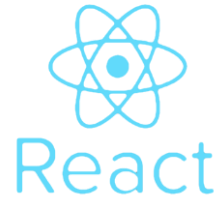
UI-Design Tools

Features:

- Easy prototype design
- Pitch an idea
- Power of visual communication
- Proof of concept
- Instant feedback
- Support prototype driven development

Component Based Frameworks

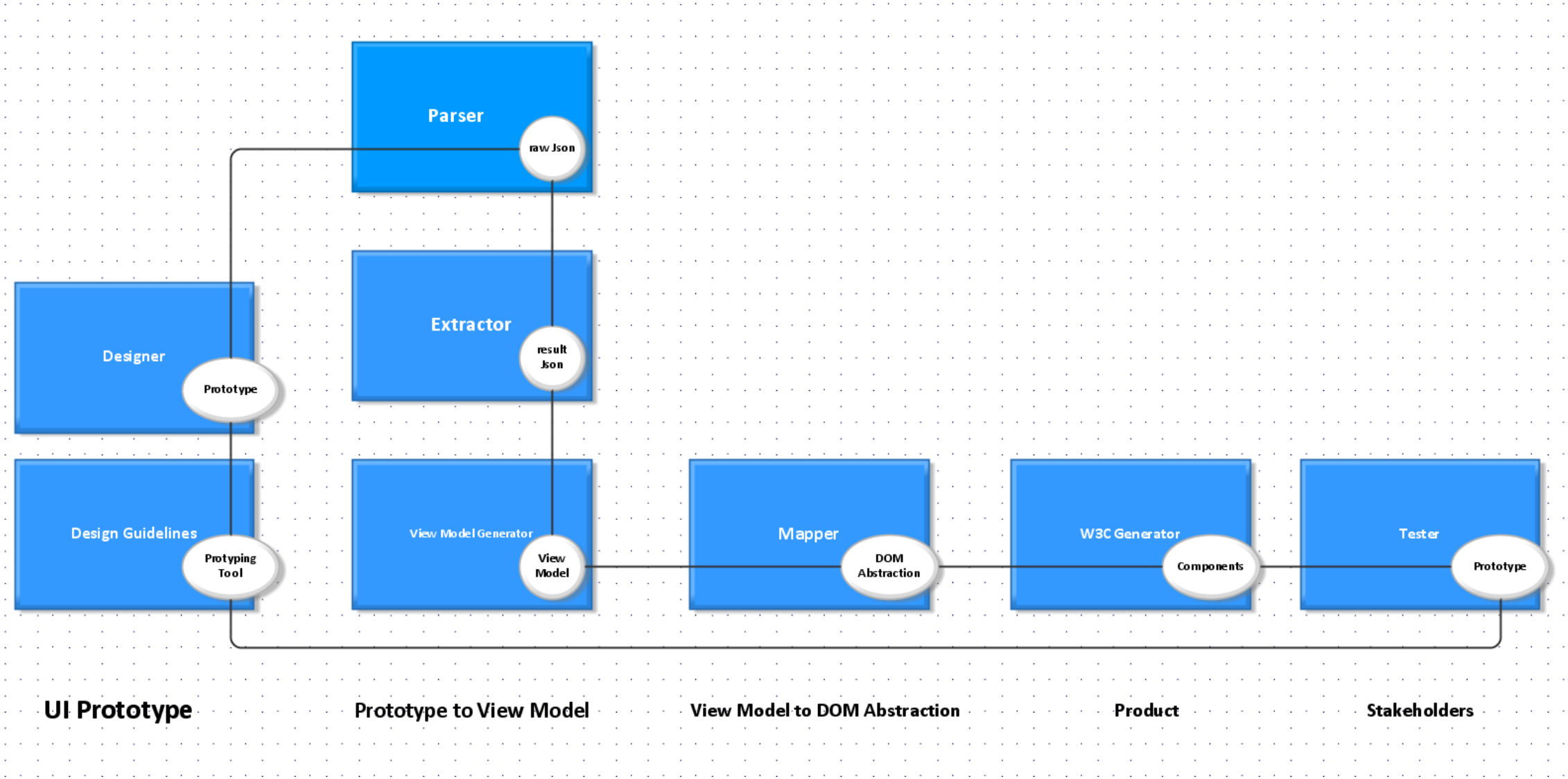
- Allow high reusability
- can be extended and modified
- split responsibilites vertically



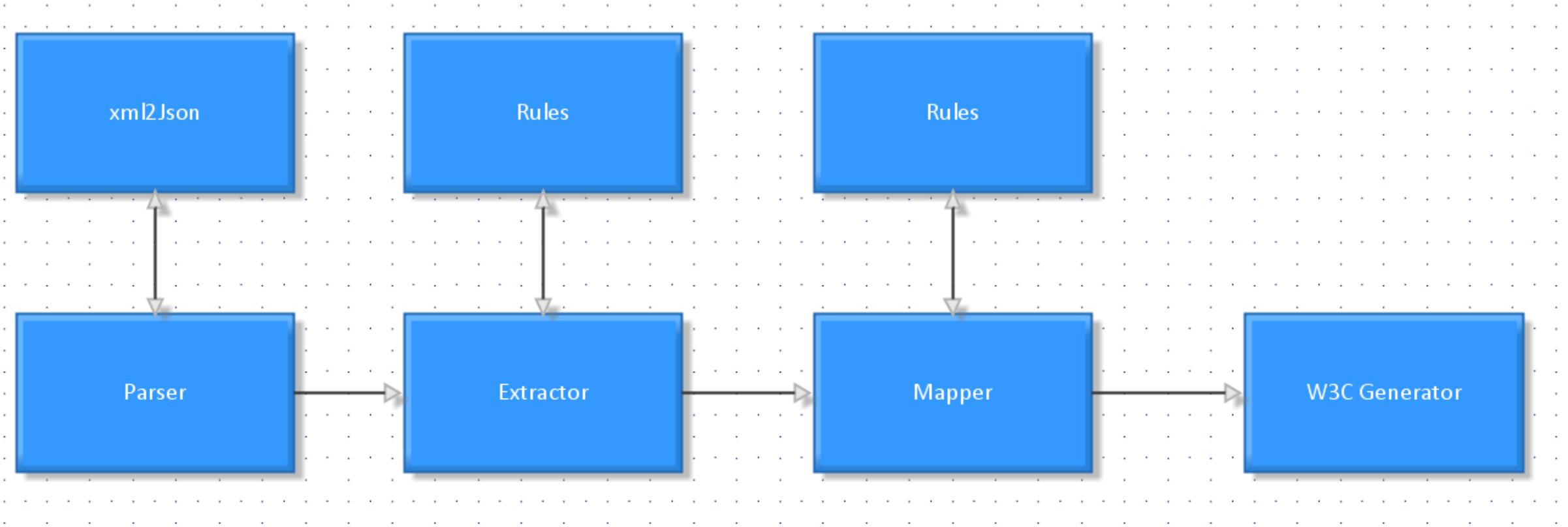
W3C - Web Components

- Custom Elements
- HTML Templates
- Shadow DOM

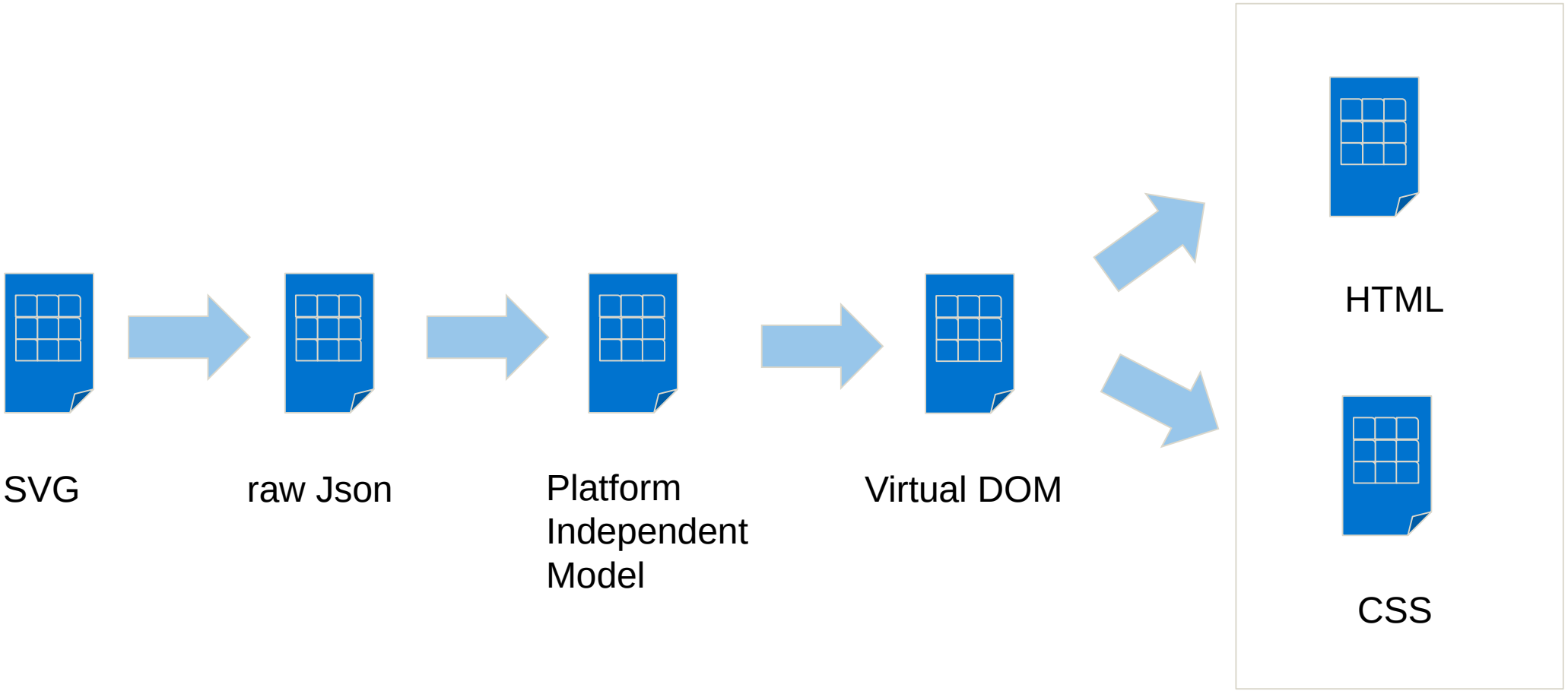
Proposed Process



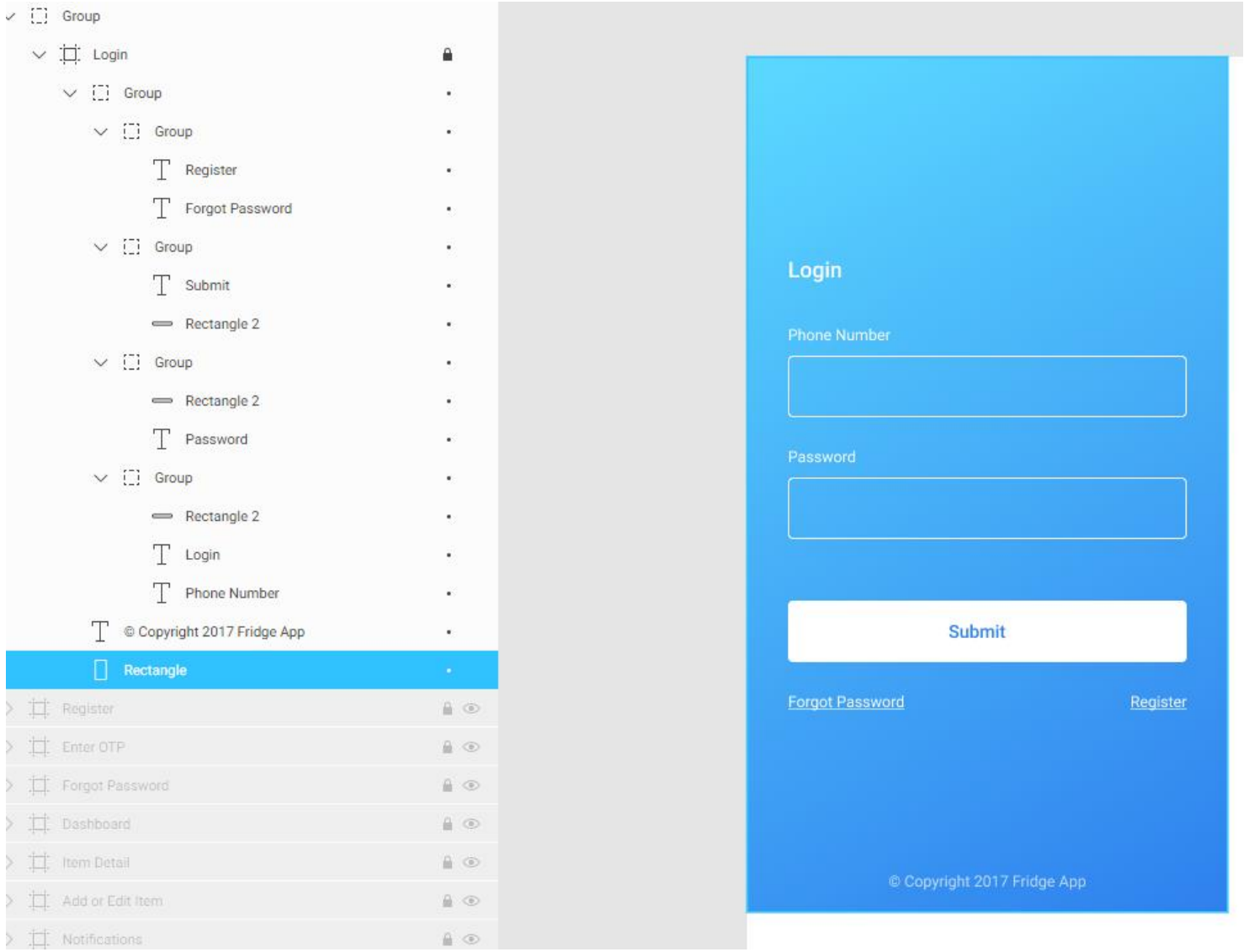
Workflow (JS Modules)



Data


HTML
CSS
SVG
raw Json
Platform Independent Model
Virtual DOM
W3C Web Components

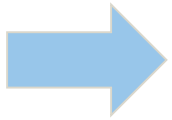
Simple Example



Simple Example

```
{
  "id": "Group",
  "g": [
    {
      "id": "Phone Number",
      "use": {
        "xlink:href": "#path2_fill",
        "transform": "translate(-156 -124)",
        "fill": "#FFFFFF"
      }
    },
    {
      "id": "Rectangle 2",
      "g": {
        "mask": "url(#mask0_outline_ins)",
        "use": {
          "xlink:href": "#path5_stroke_2x",
          "transform": "translate(-156 -101)",
          "fill": "#FFFFFF"
        }
      }
    }
  ],
```

Parsed SVG

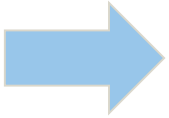


```
{
  "div": {
    "id": "Group",
    "label": {
      "id": "Phone Number",
      "style": {
        "fill": "#FFFFFF",
        "transform": "translate(-156 -124)"
      }
    },
    "form": {
      "id": "Rectangle-2",
      "style": {
        "fill": "#FFFFFF",
        "transform": "translate(-156 -101)"
      }
    }
  }
}
```

View Model

Simple Example

```
{
  "div": {
    "id": "Group",
    "label": {
      "id": "Phone Number",
      "style": {
        "fill": "#FFFFFF",
        "transform": "translate(-156 -124)"
      }
    },
    "form": {
      "id": "Rectangle-2",
      "style": {
        "fill": "#FFFFFF",
        "transform": "translate(-156 -101)"
      }
    }
  }
}
```



```
<style>
#Phone-Number{
  fill:#FFFFFF;
  transform:translate(-156px, -124px);
  /*
  .....
  */
}
#Rectangle-2{
  fill:#FFFFFF;
  transform:translate(-156px, -101px);
  /*
  .....
  */
}
</style>
.....
<div id=Group>
  <label id=Phone-Number>Phone Number</label>
  <form id=Rectangle-2>
    <input type="text" name="Phone Number"><br>
  </form>
</div>
.....
```

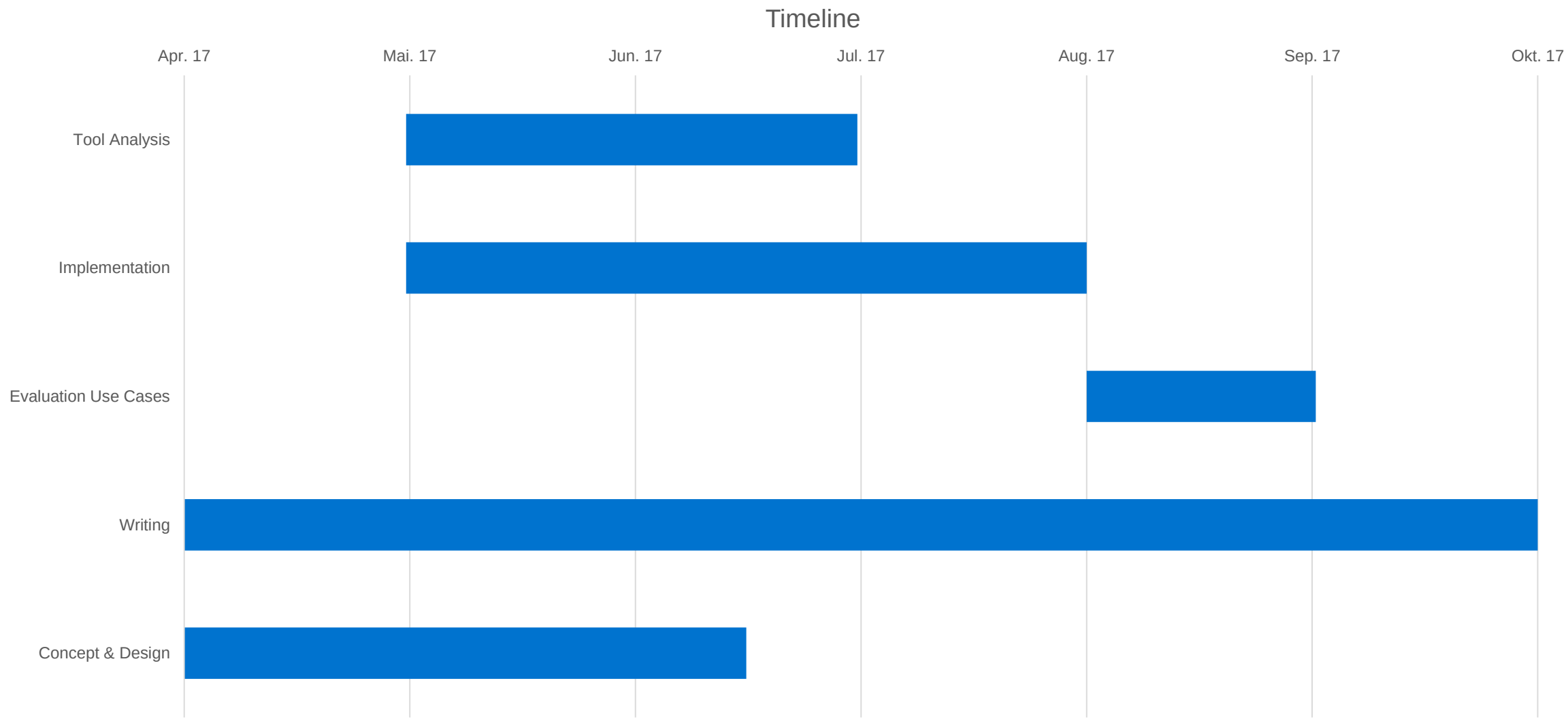
View Model

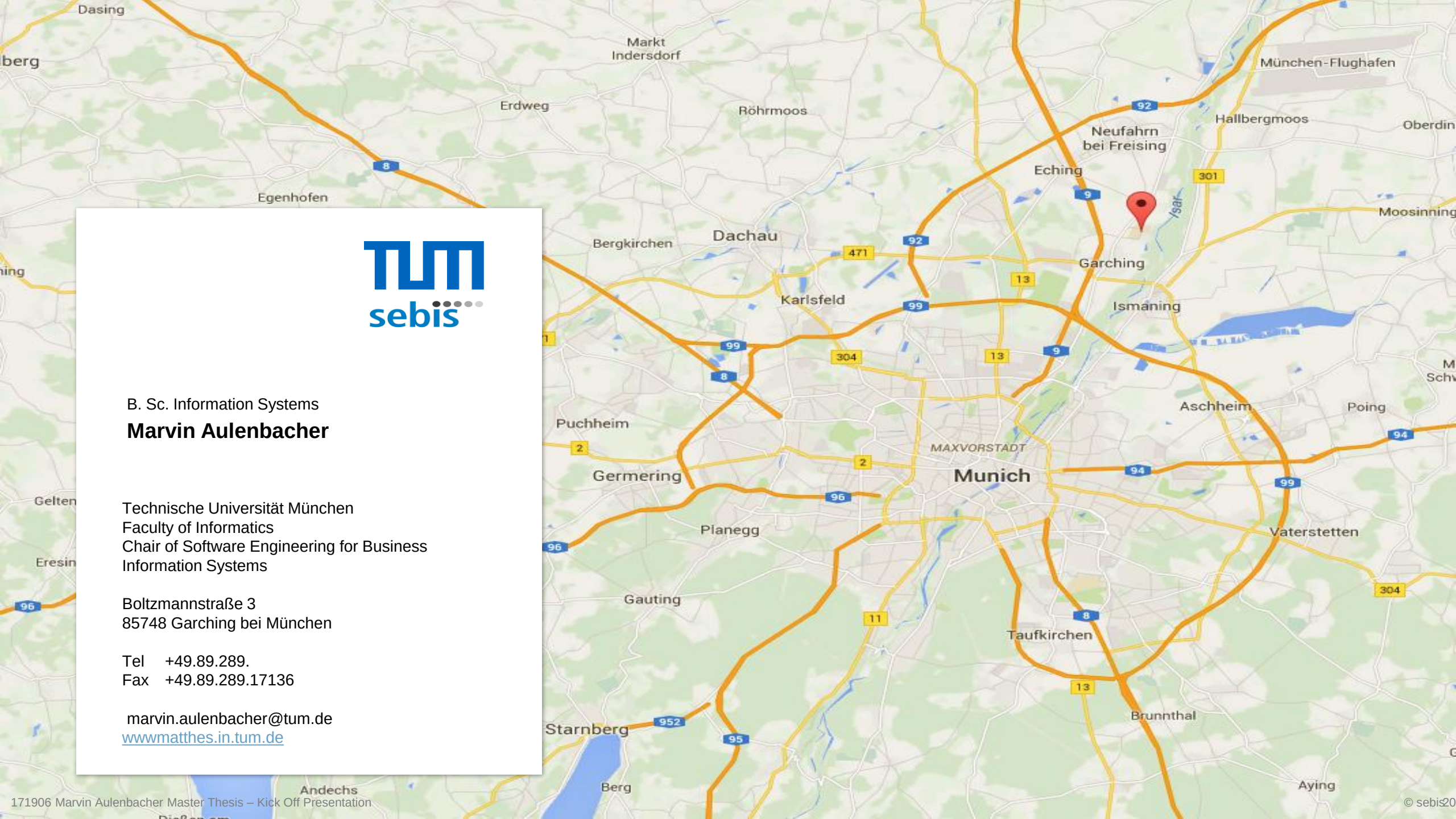
Web Component

Next Steps

- Render Components in Shadow DOM
- Create new rules and modify existing ones
- Complete proposed process
- Work on use cases

Timeline





B. Sc. Information Systems

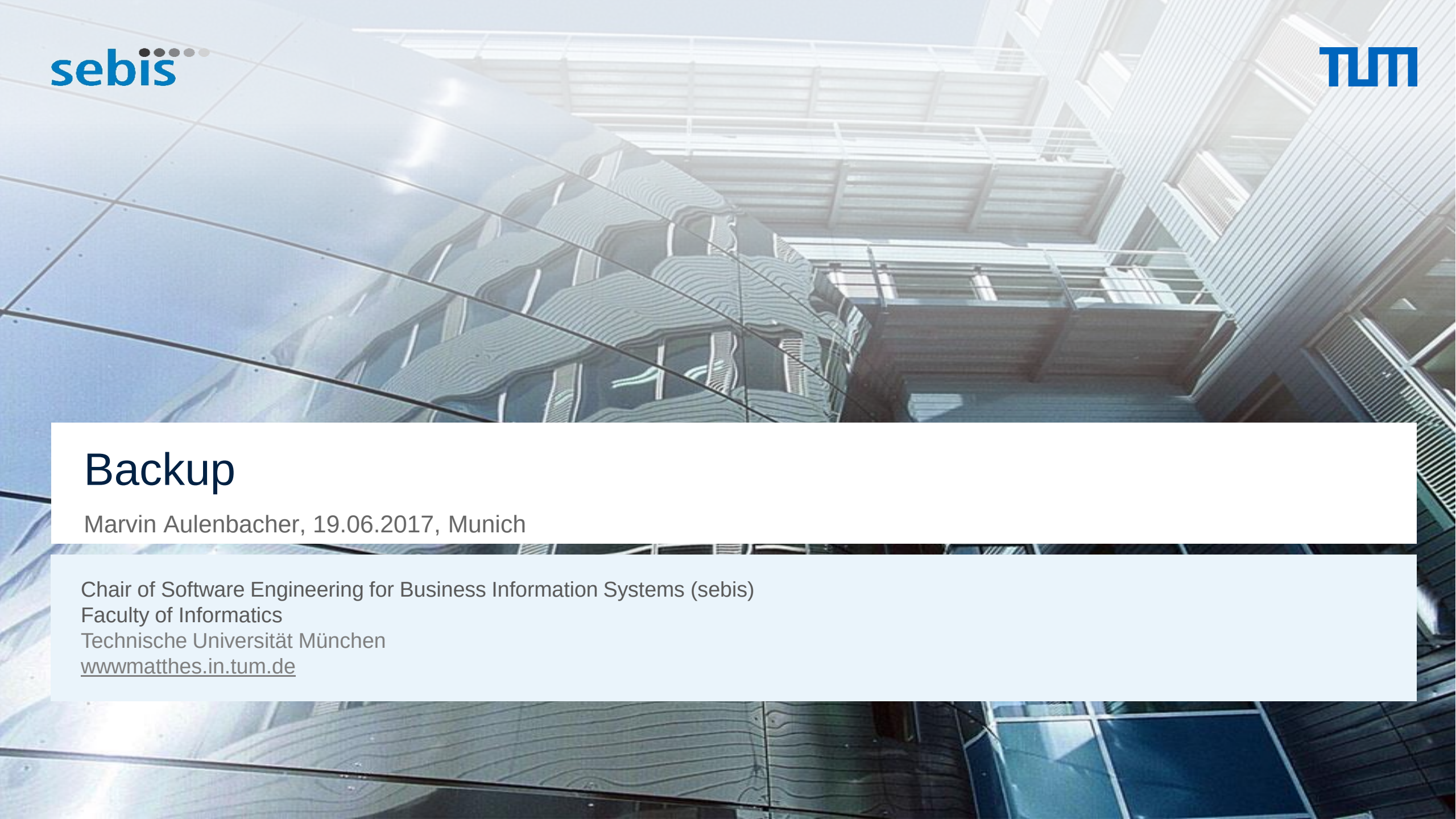
Marvin Aulenbacher

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

marvin.aulenbacher@tum.de
wwwmatthes.in.tum.de

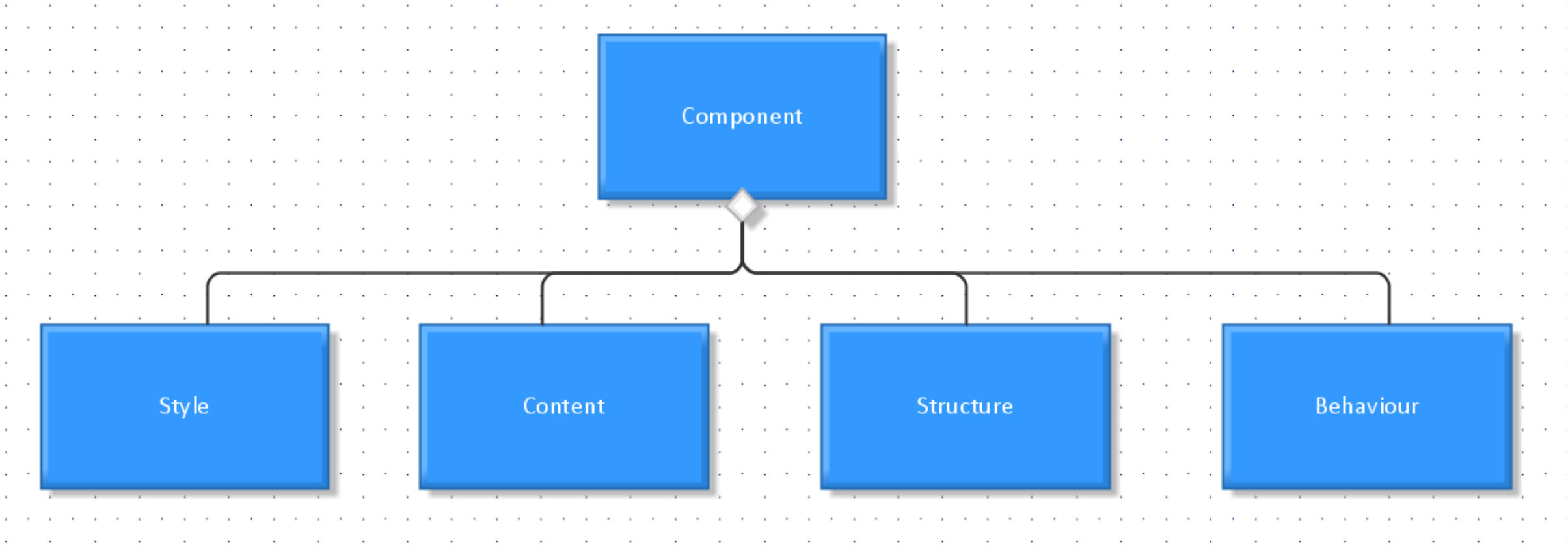


Backup

Marvin Aulenbacher, 19.06.2017, Munich

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

Component UML View Model



W3C Components

- Specification for platform independent web components

- Consist of several seperate technologies

- Reusable user interface widgets

- Part of browser environment(no external libraries needed)

Custom Elements

- create new HTML tags (customElement.define(name,component, options:{}))
- Modify existing HTML tags
- Extend HTML tags (e.g. Button)
- life cycle behaviours(connected, disconnected)
- Can have own scripted behaviour and CSS styling

HTML Templates

- client-side content

- Is not rendered when page is loaded, has to be activated

- Parser only validates content of template

- High reusability

Shadow DOM

Two differences to normal DOM

1. How it is created and used
2. How it behaves in relation to the rest of the page

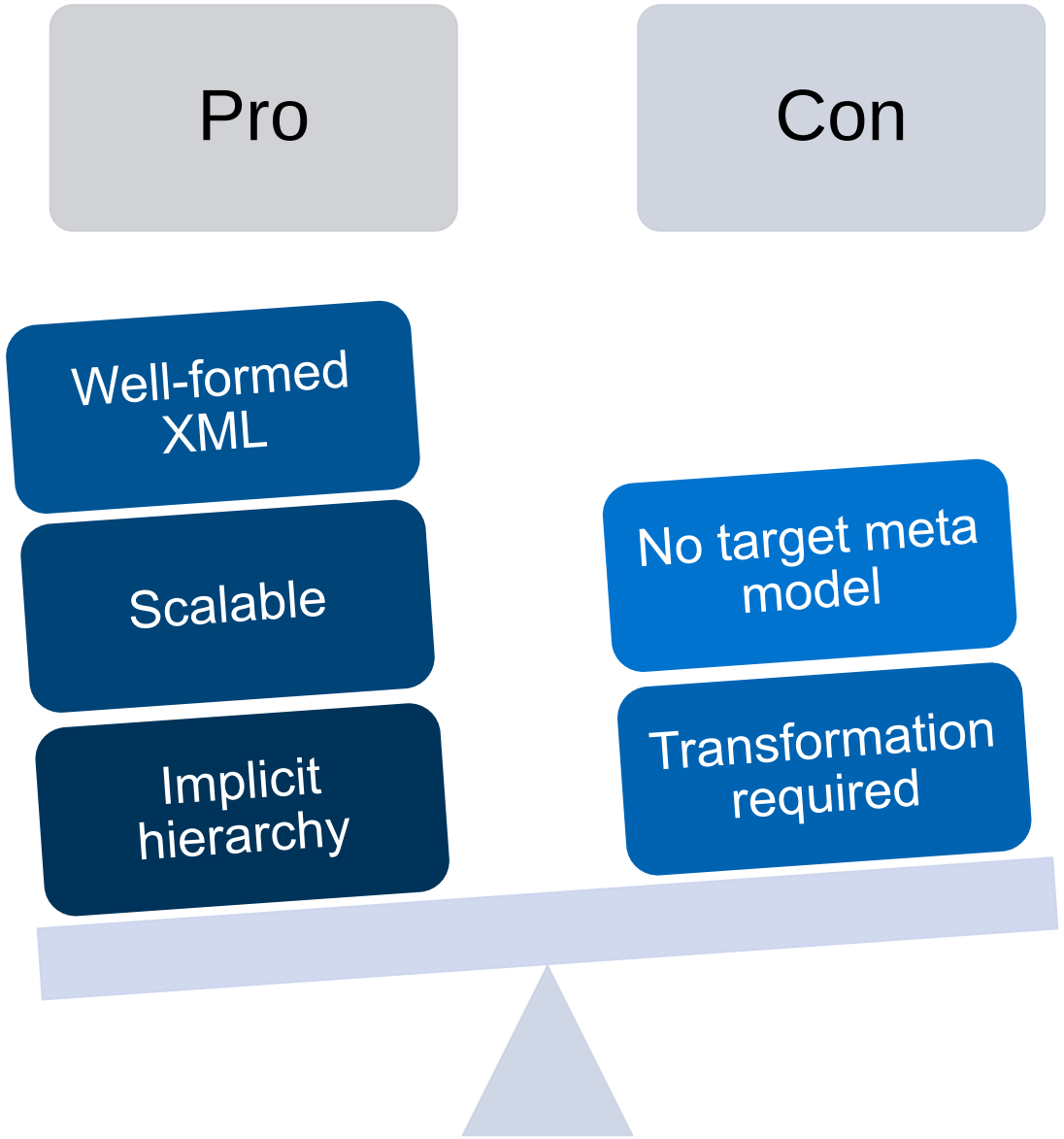
This enables:

- Tree-scoping
- Details of the implementation are hidden.
- Shadow root contains implementation details and are rendered into a single tree
- Shadow dom must be attached to an existing element

Browser Support

- Backward compability via polyfills
- Custom Elements supported by modern browsers
- Early version support of Shadom Dom and Custom Elements in Chrome, Opera, Firefox
- Edge has not started to implement Shadow Dom nor Custom Elements

Why SVG?



Why JSON?