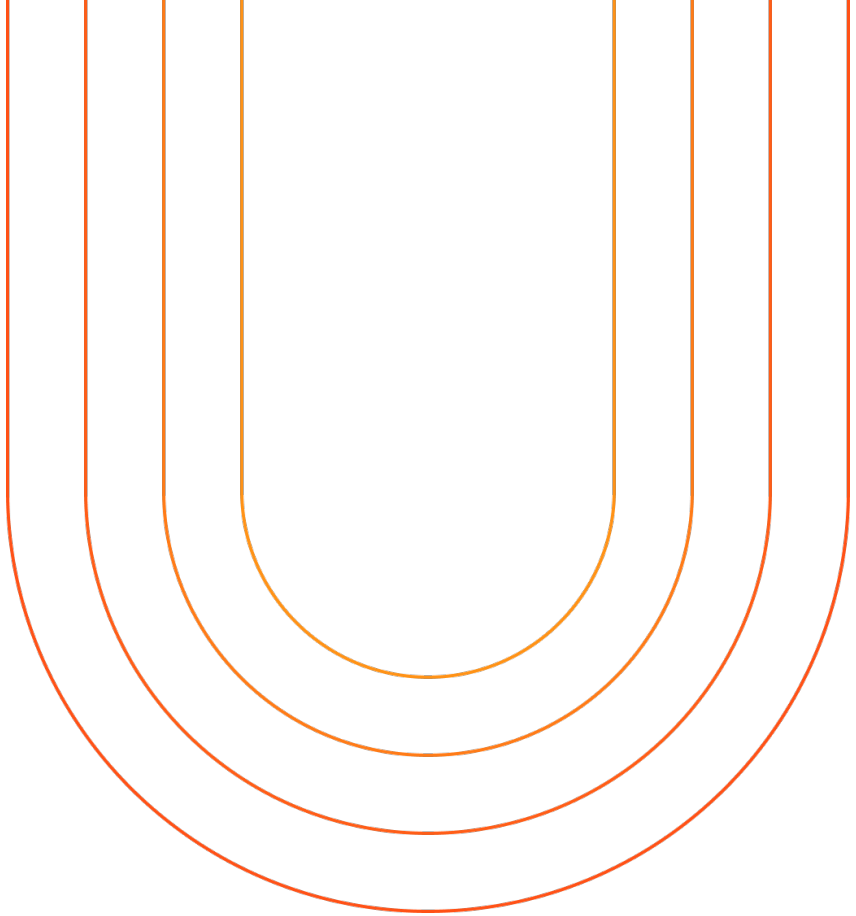
Component Reuse and “Micro-Frontend" Concepts in a SPA

A Report

Andreas Pflugrad, eXXcellent solutions

July 7th/8th, 2022



UI5con HYBRID

Component Reuse and “Micro-Frontend" Concepts in a SPA

A Report

Andreas Pflugrad, eXXcellent solutions

July 7th/8th, 2022

UI5con HYBRID

Introduction

- What is this talk about?

The standard use case

- Embedding a component

Different operating modes

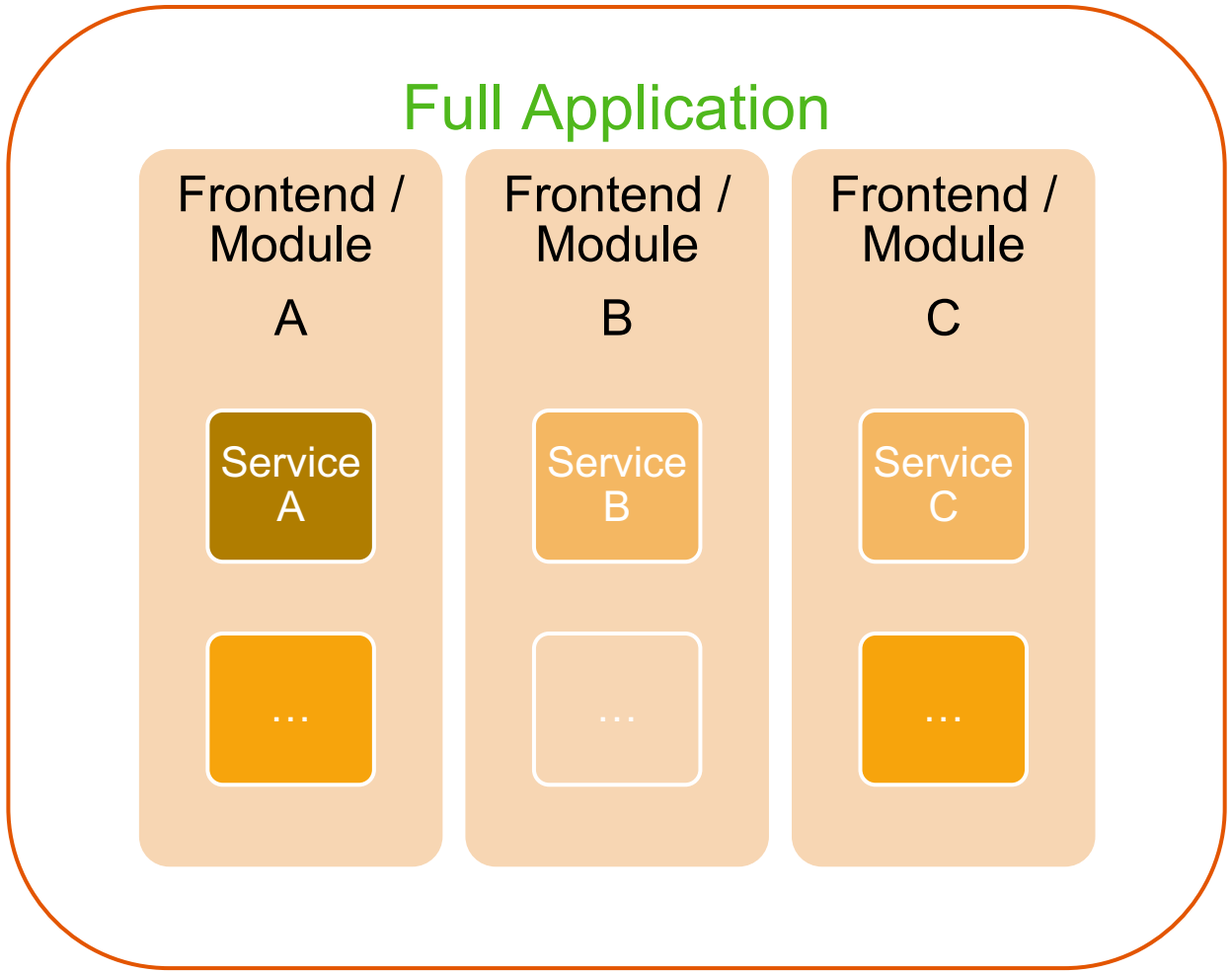
- Same component, different data

Different output formats

- Same component, different views

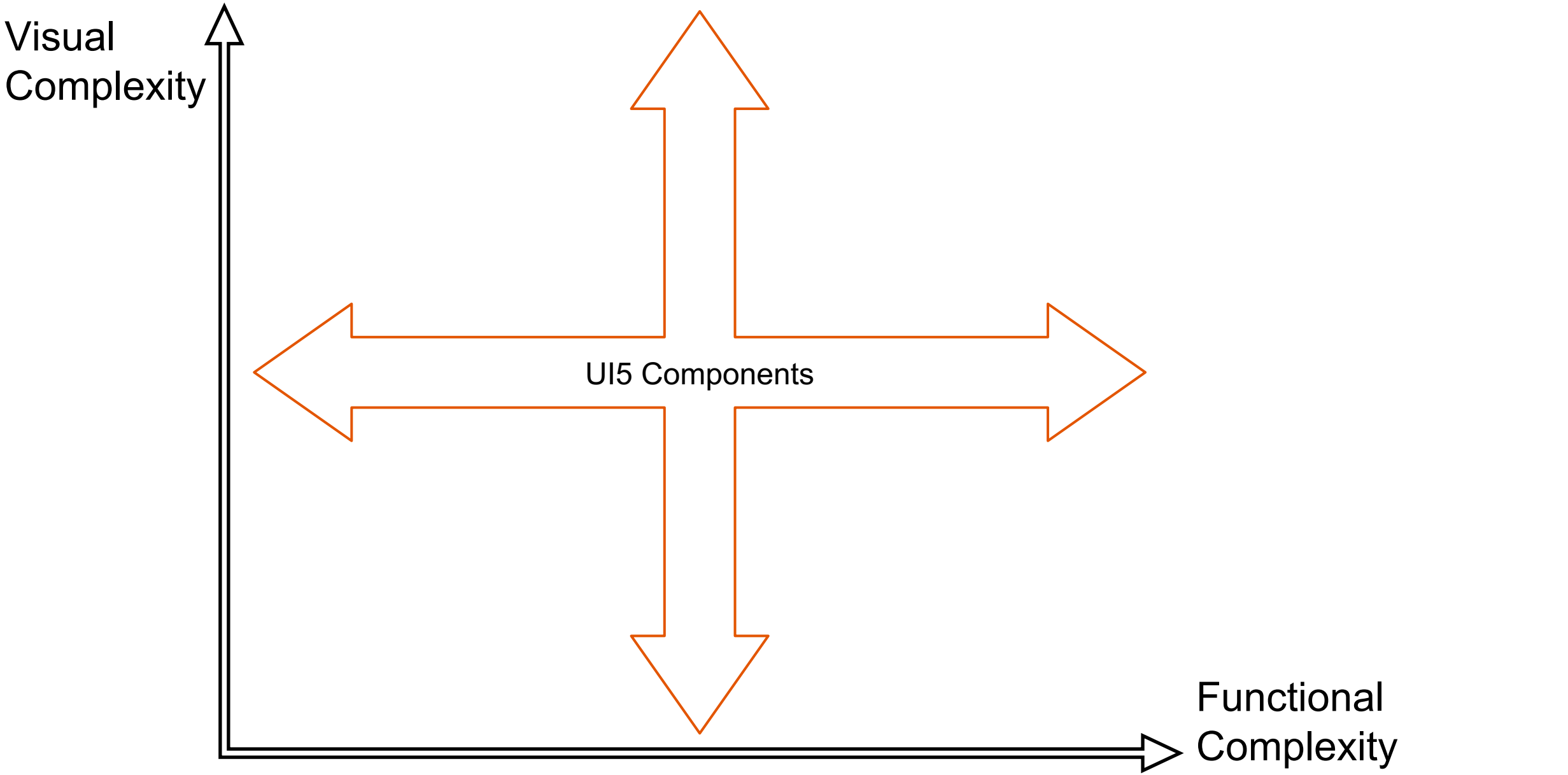
“Micro-Frontend”-Concept

(but just the part of it that matters for this talk)

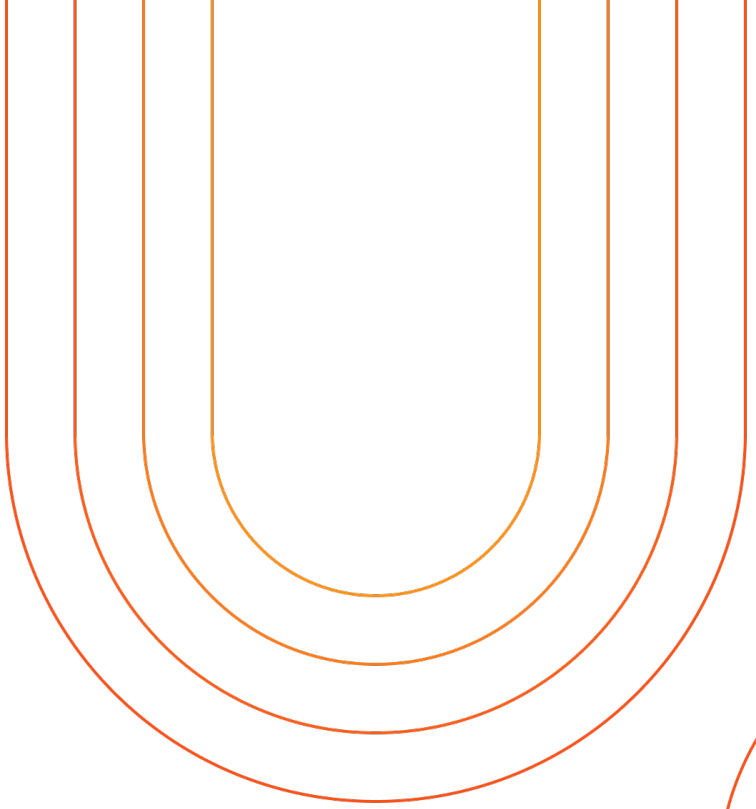


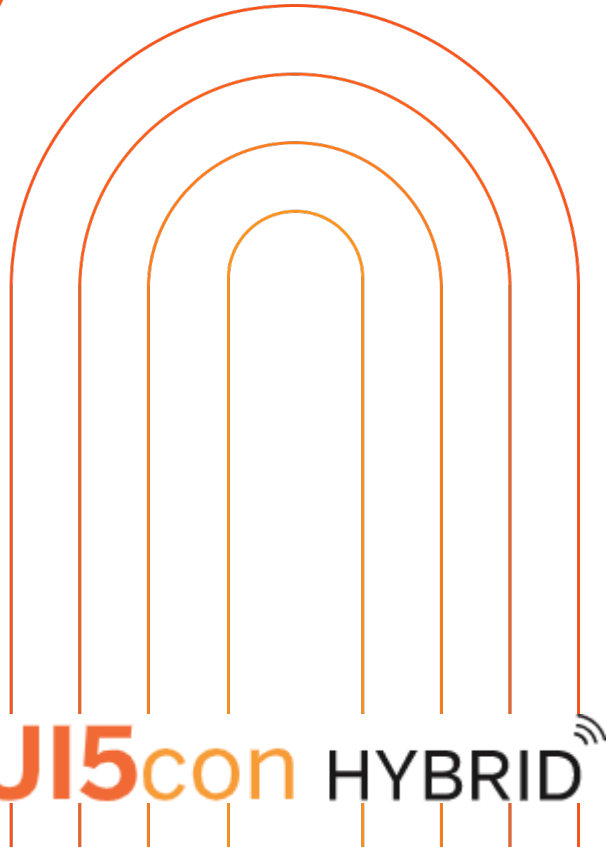
Having small frontend applications that comprise a bigger application

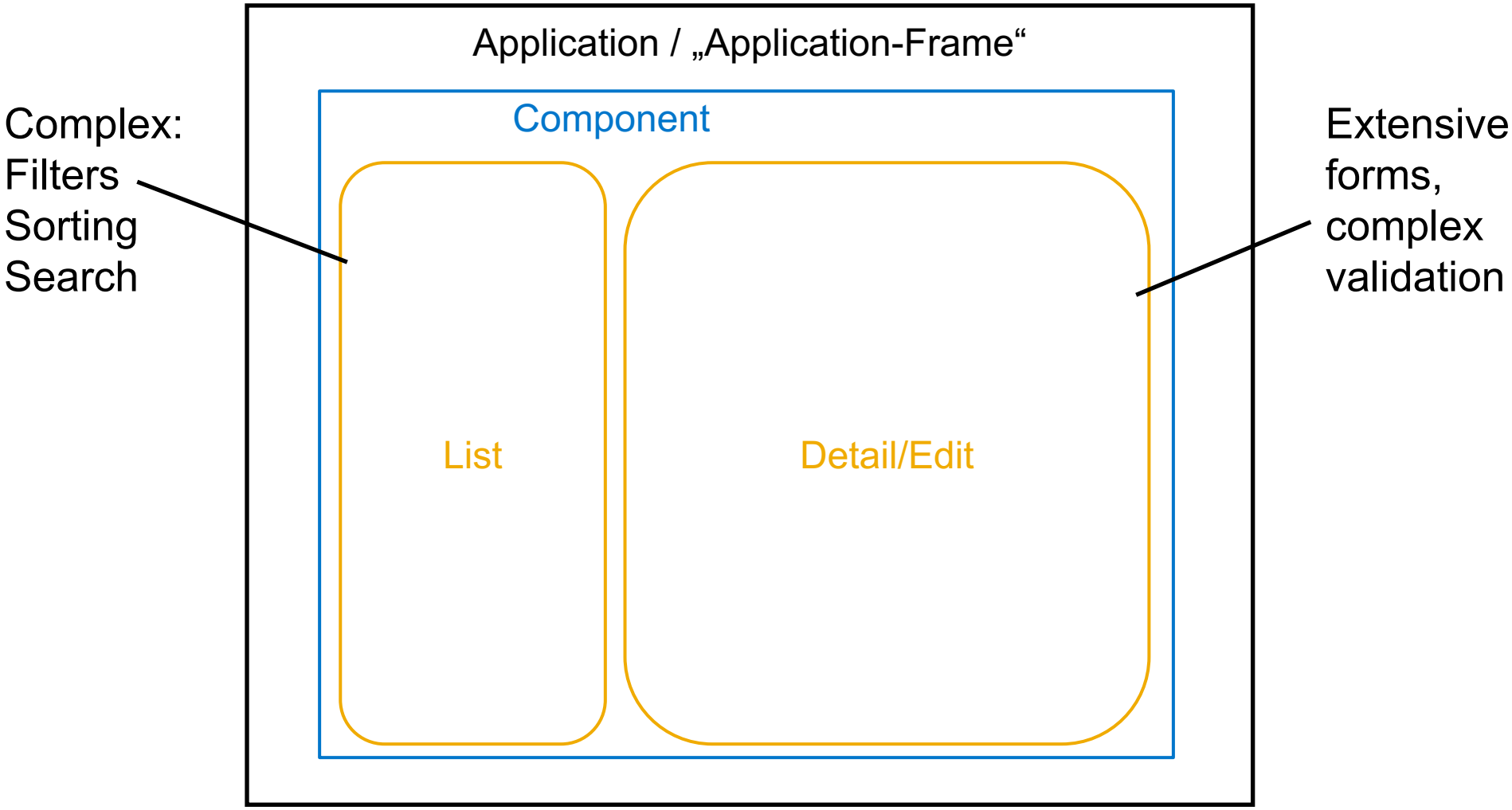
… i.e. UI components that handle their own data

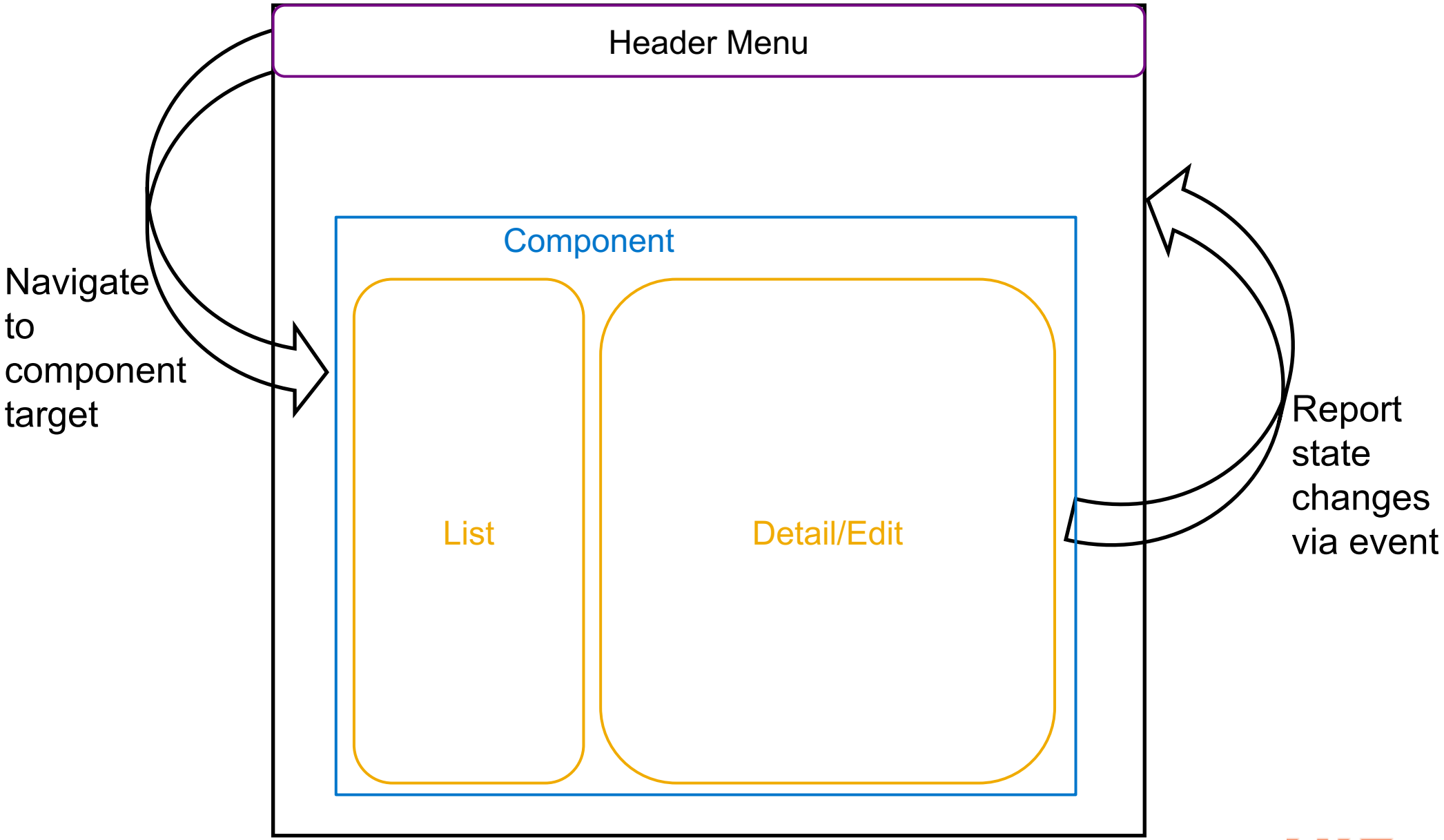

Visual
Complexity
UI5 Components
Functional
Complexity

The standard use case



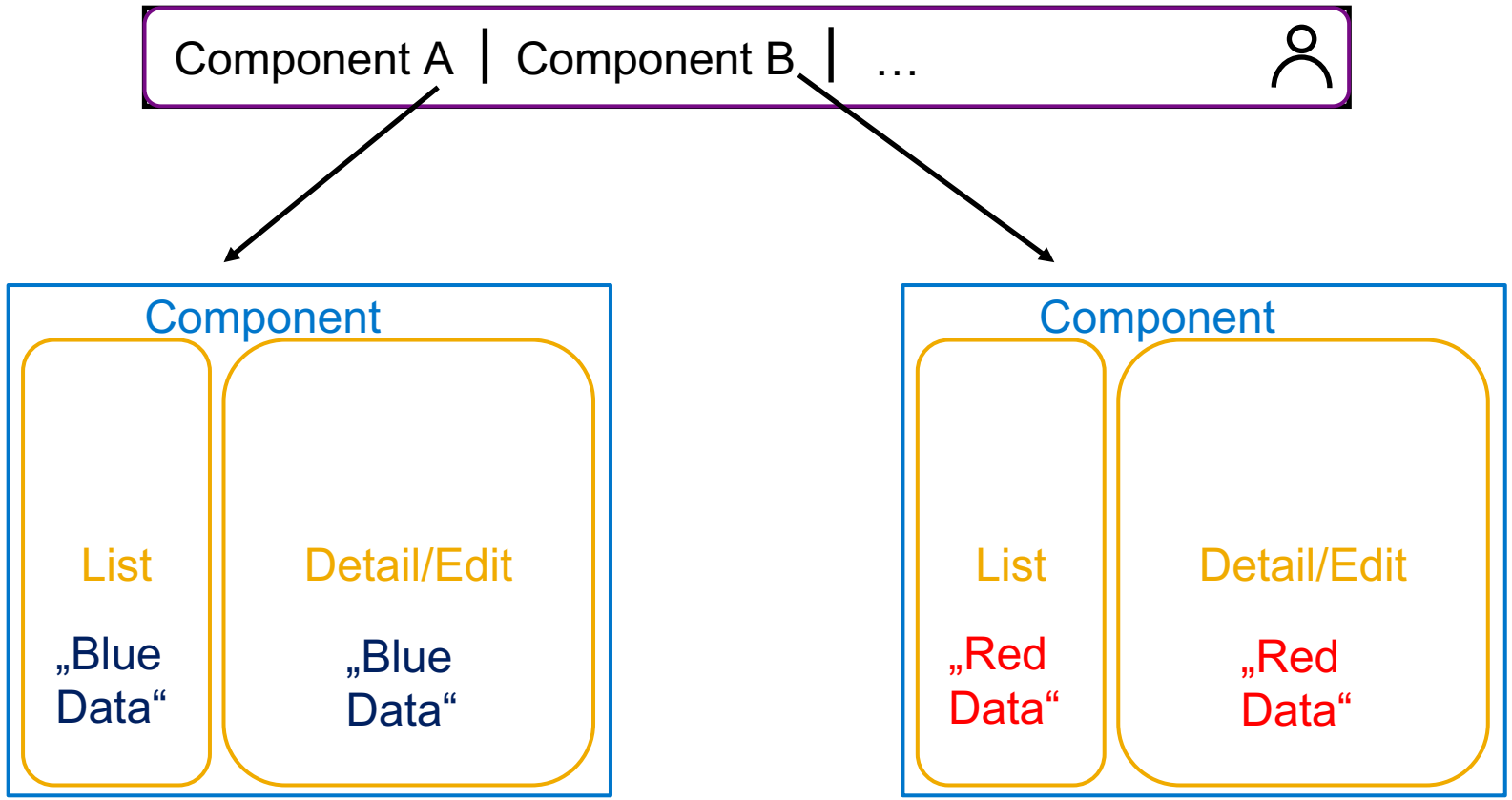



Application / „Application-Frame“
Component
List
Detail/Edit
Complex:
Filters
Sorting
Search
Extensive
forms,
complex
validation

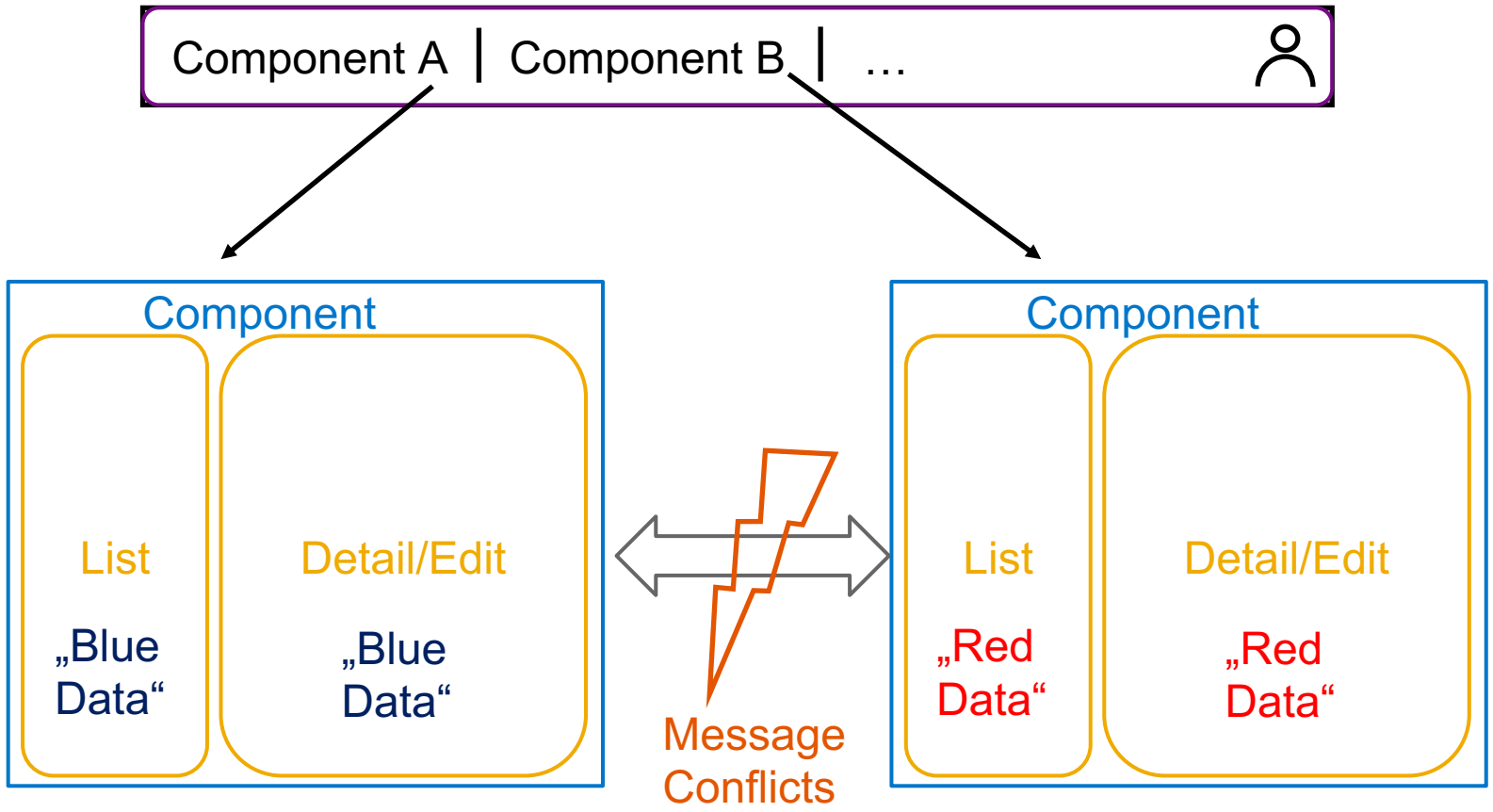

Header Menu
Component
List
Detail/Edit
Navigate to component target
Report state changes via event

Different operating modes

Same component, different mode



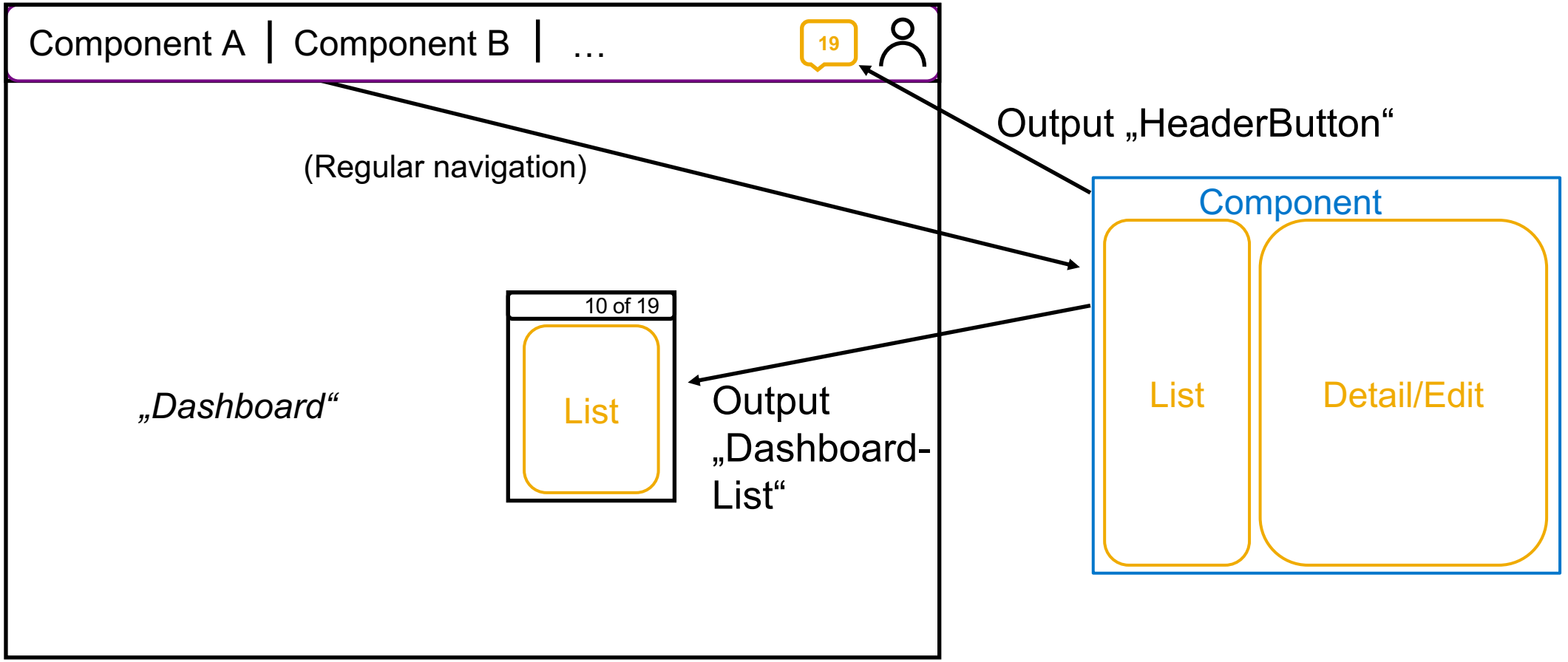
Same component, different mode



Solution: don‘t share the MessageManager!

Different output formats

Same component, different output



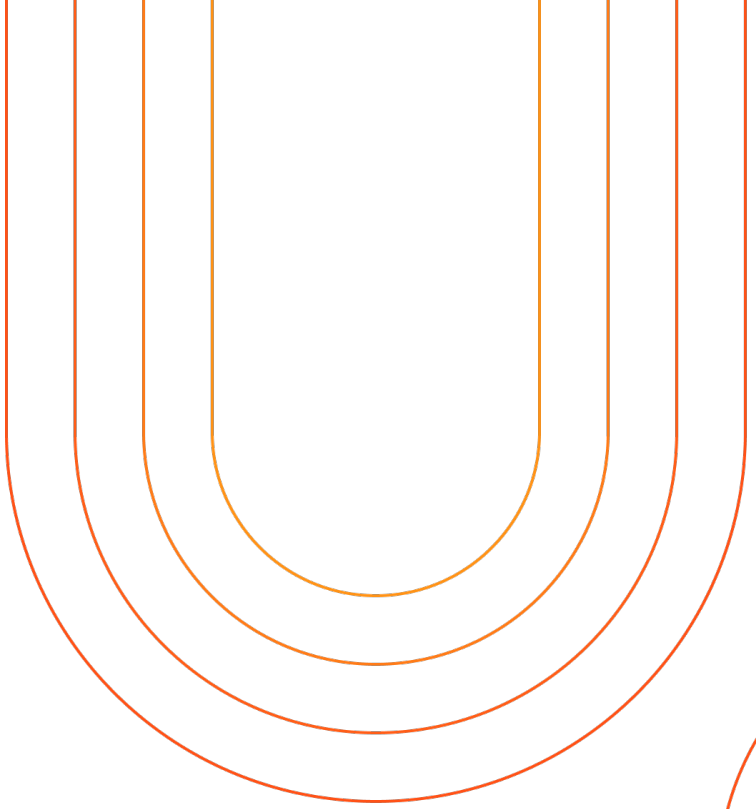
Have a closer look at the sample code at:

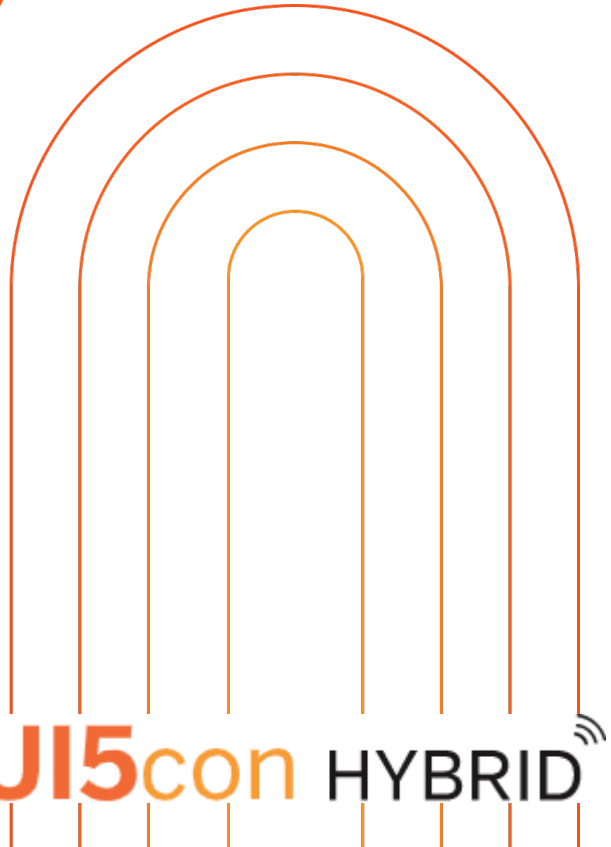
https://github.com/exxcellent/UI5con2022-Demo

Thank you!

Andreas Pflugrad
Business Architect
eXXcellent solutions GmbH

andreas.pflugrad@exxcellent.de





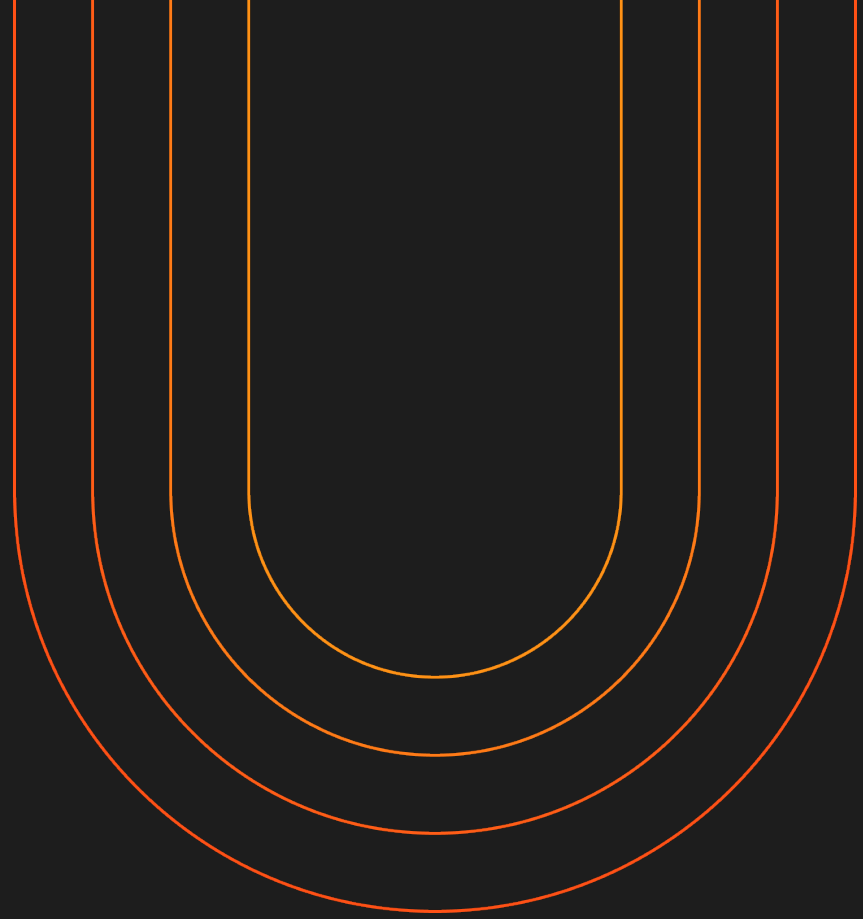
UI5con HYBRID

Component Reuse and “Micro-Frontend" Concepts in a SPA

A Report

Andreas Pflugrad, eXXcellent solutions

July 7th/8th, 2022



UI5con HYBRID

Component Reuse and “Micro-Frontend" Concepts in a SPA

A Report

Andreas Pflugrad, eXXcellent solutions

July 7th/8th, 2022

UI5con HYBRID

Introduction

- What is this talk about?

The standard use case

- Embedding a component

Different operating modes

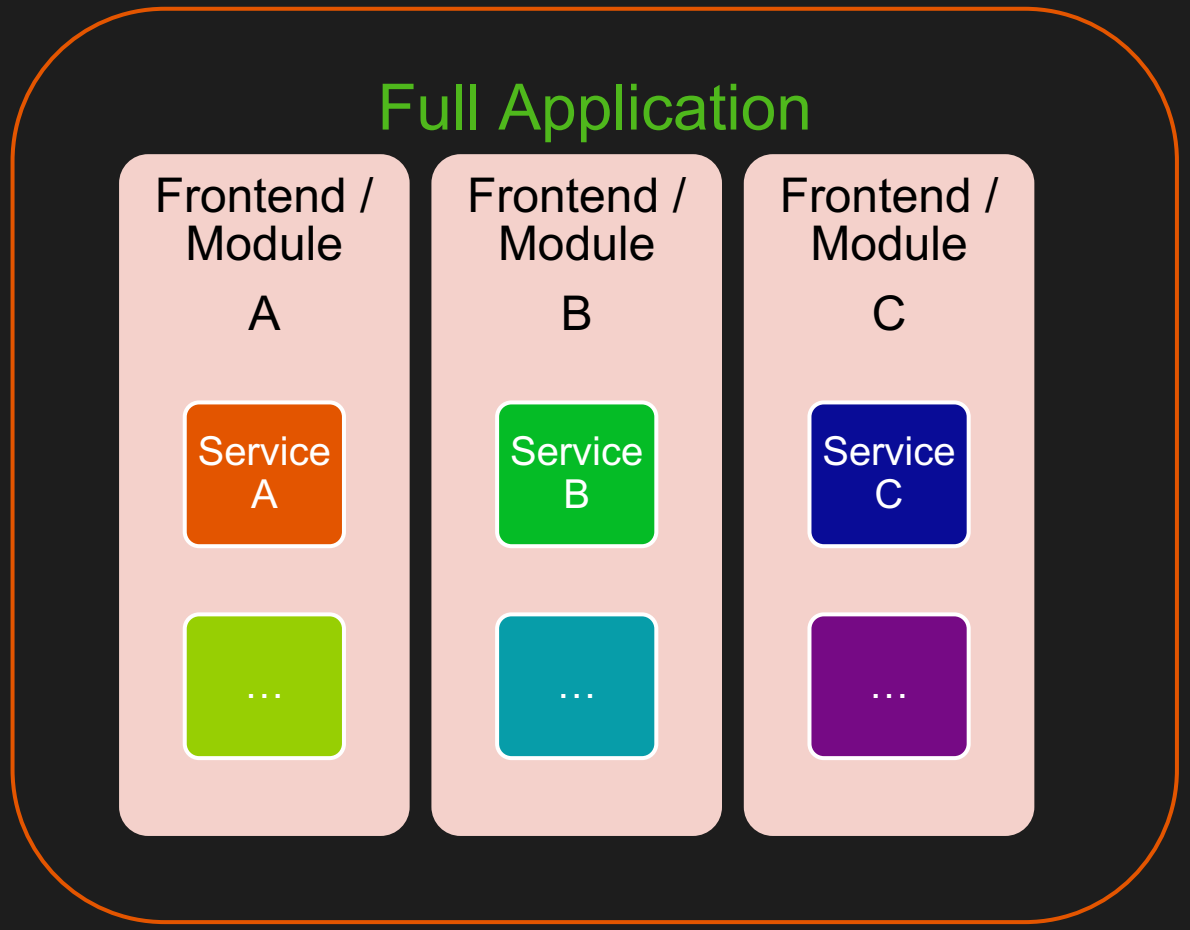
- Same component, different data

Different output formats

- Same component, different views

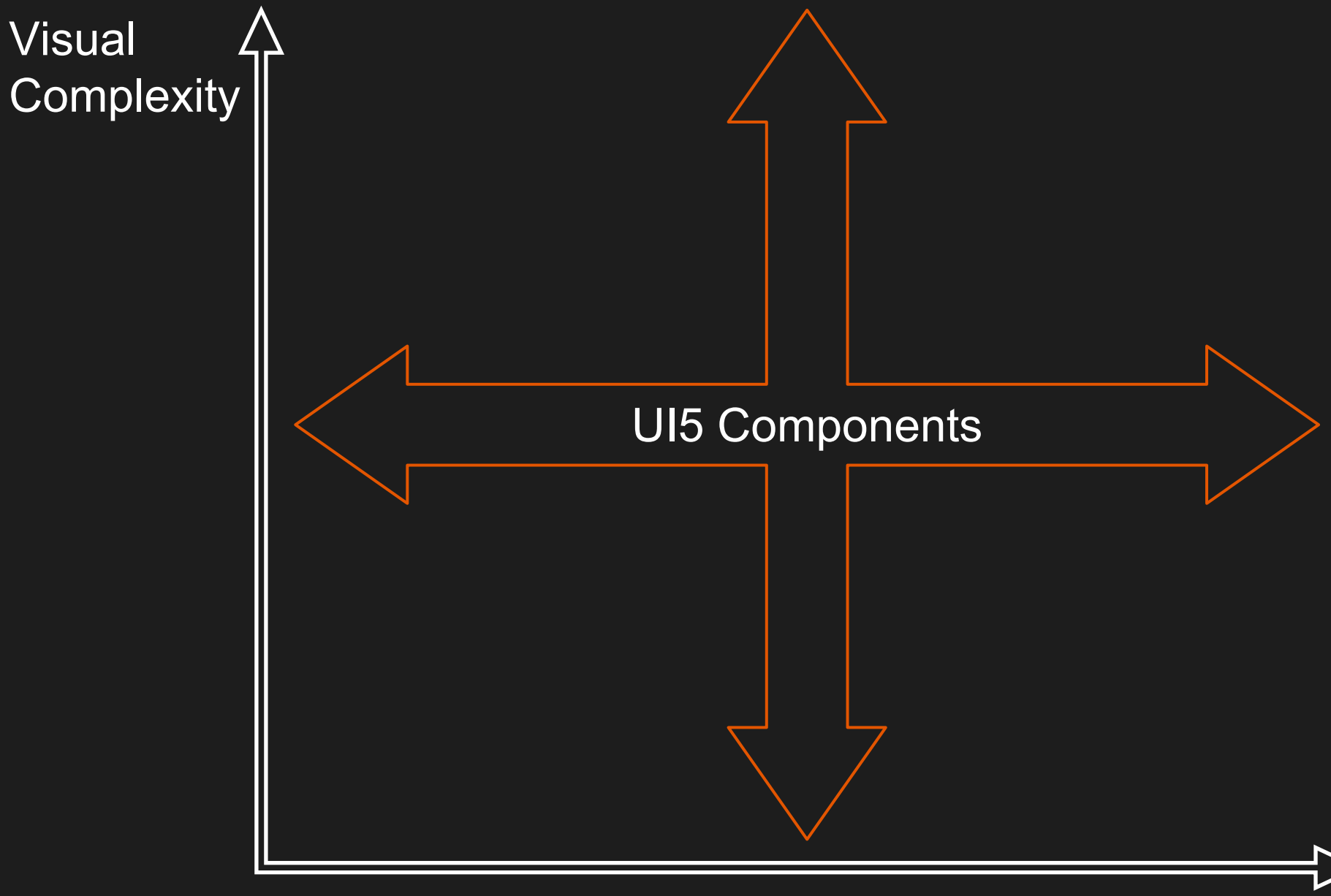
“Micro-Frontend”-Concept

(but just the part of it that matters for this talk)

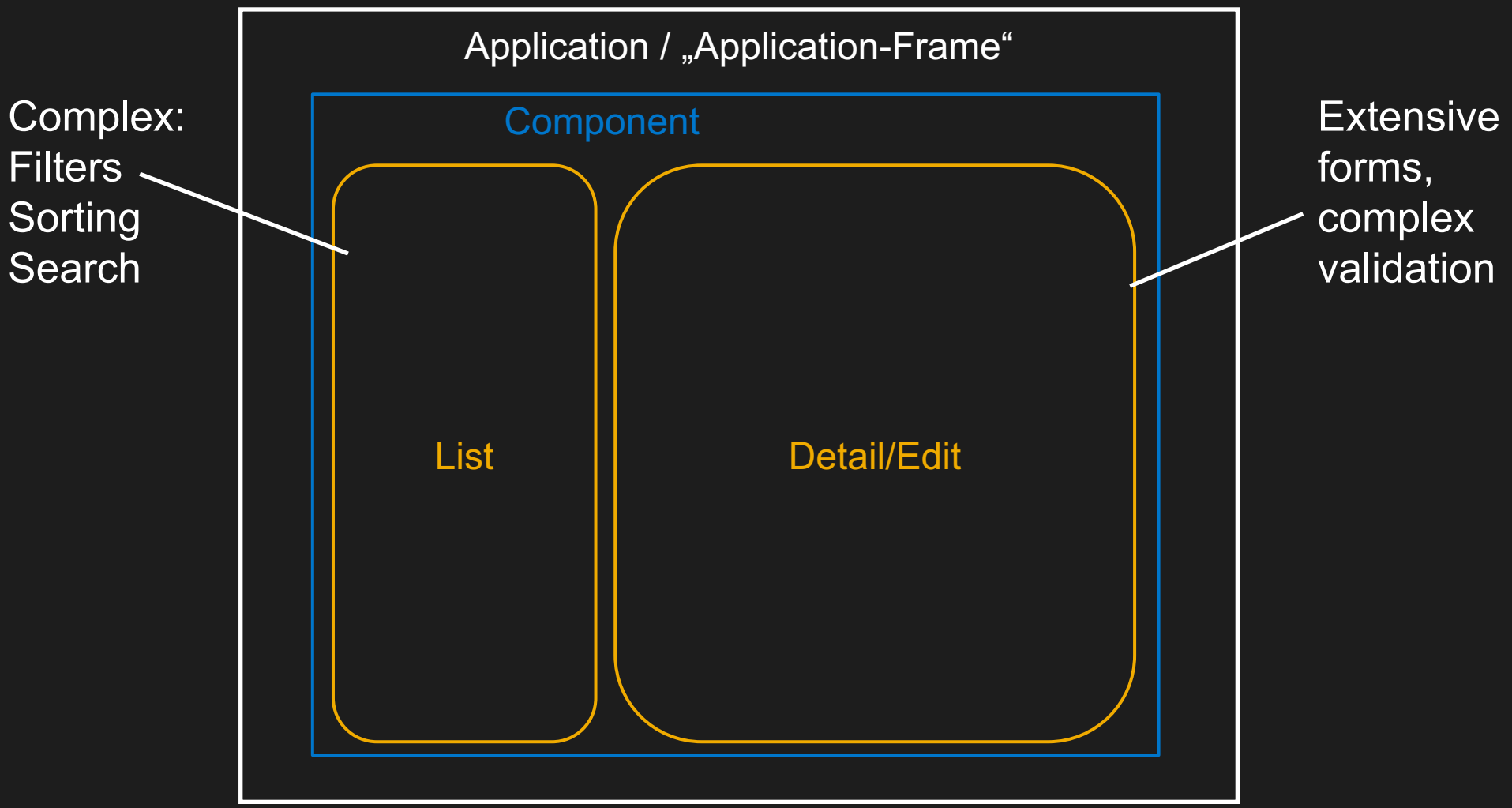


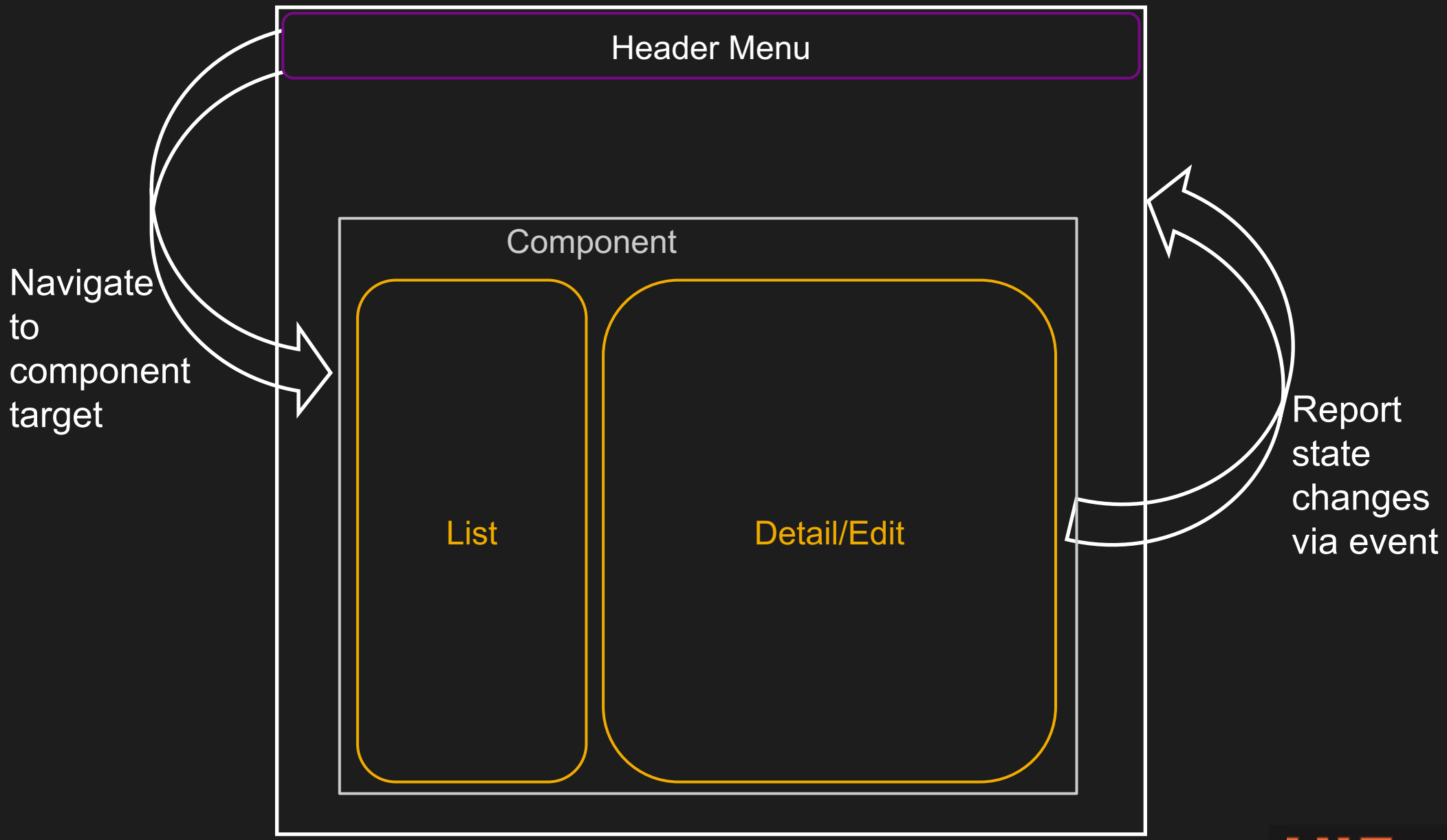
Having small frontend applications that comprise a bigger application

… i.e. UI components that handle their own data


Visual
Complexity
UI5 Components
Functional
Complexity

The standard use case

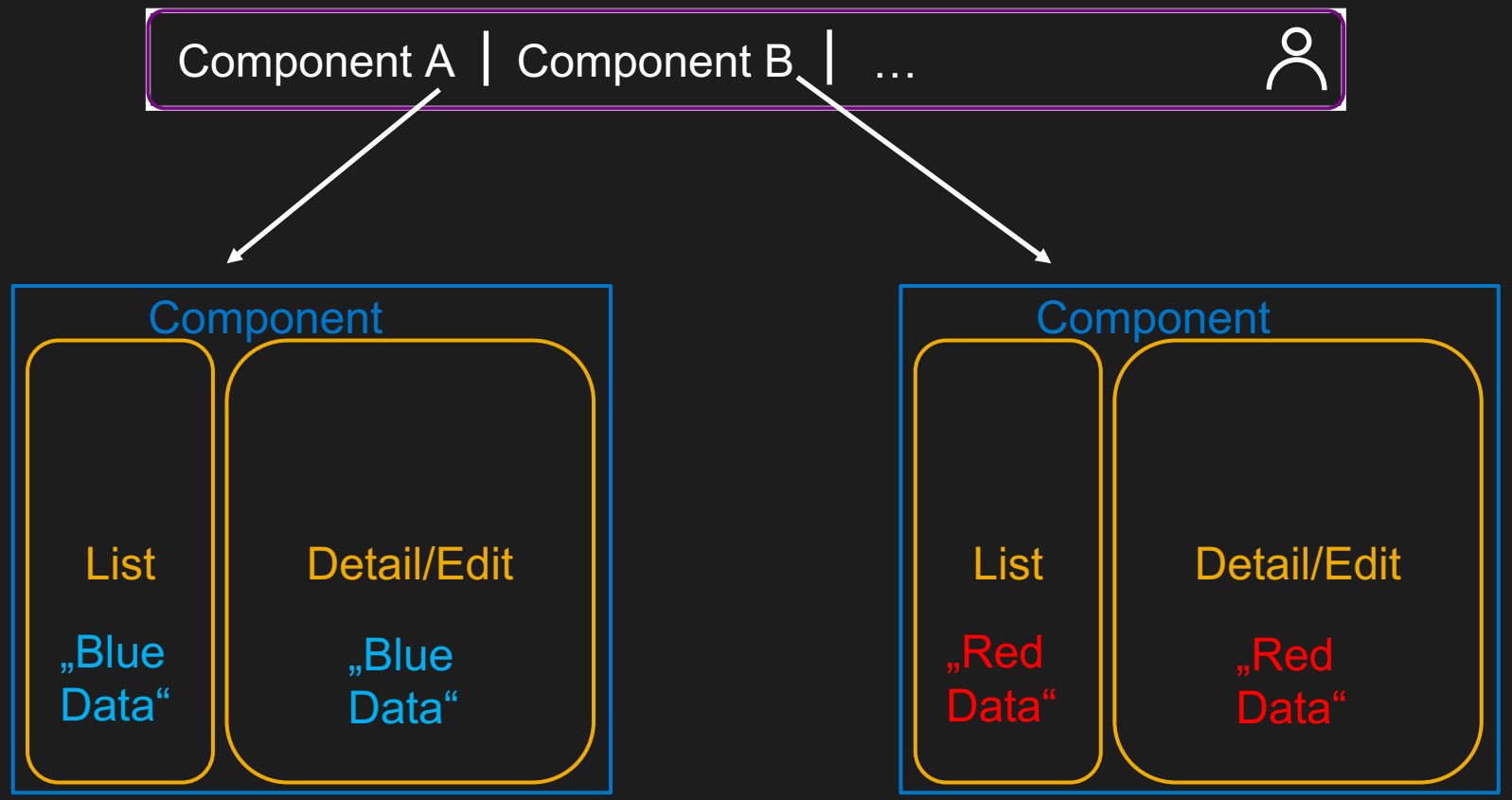

Application / „Application-Frame“
Component
List
Detail/Edit
Complex:
Filters
Sorting
Search
Extensive
forms,
complex
validation



UI5con HYBRID

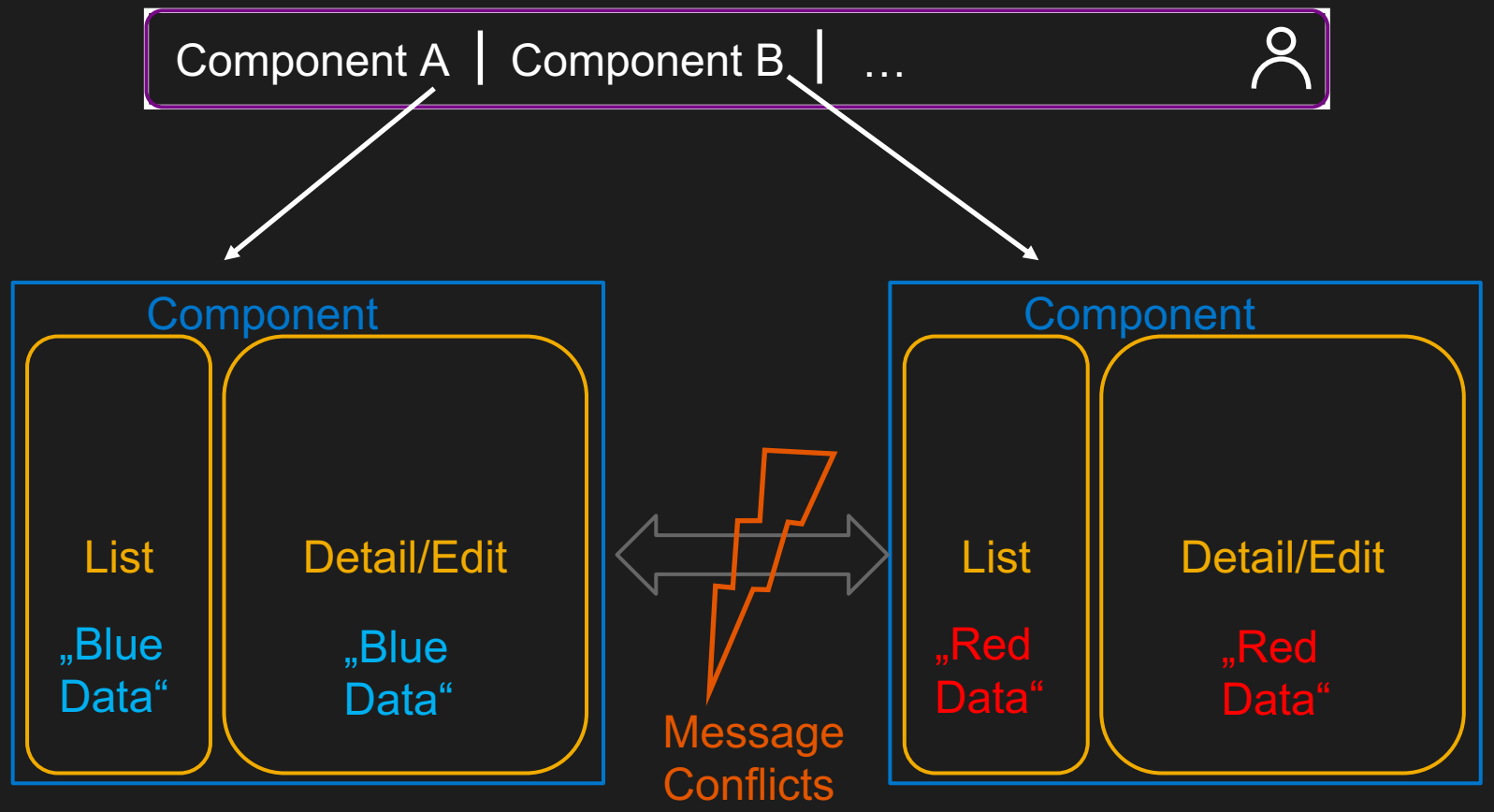
Different operating modes

Same component, different mode



UI5con HYBRID

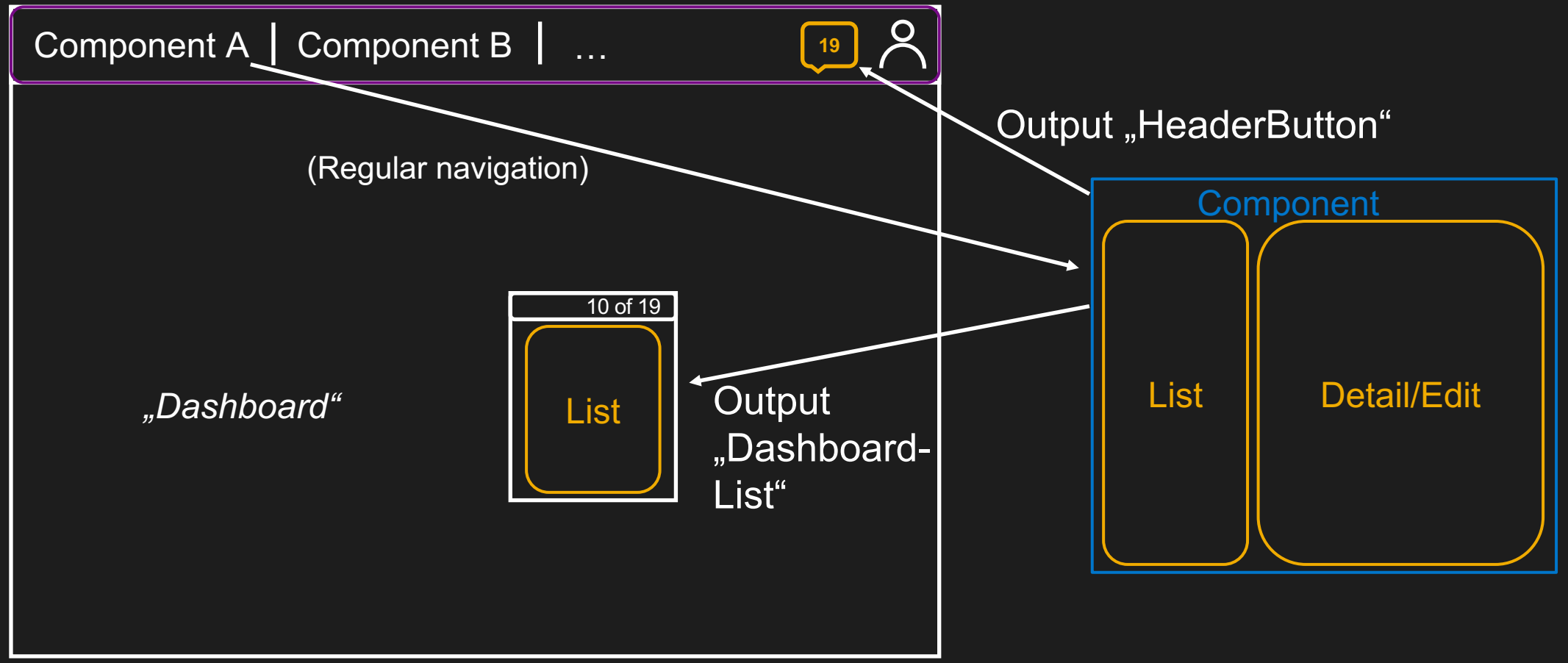
Same component, different mode



Solution: don‘t share the MessageManager!

Different output formats

Same component, different output



Have a closer look at the sample code at:

https://github.com/exxcellent/UI5con2022-Demo

Thank you!

Andreas Pflugrad
Business Architect
eXXcellent solutions GmbH

andreas.pflugrad@exxcellent.de