The challenge to use graphQL for an evolution to microservice architecture.

Lukas Ramach, BOC-Group

Introduction

DI (FH) Lukas RAMACH

lukas.ramach@boc-group.com

https://at.linkedin.com/in/lukas-ramach-18219374

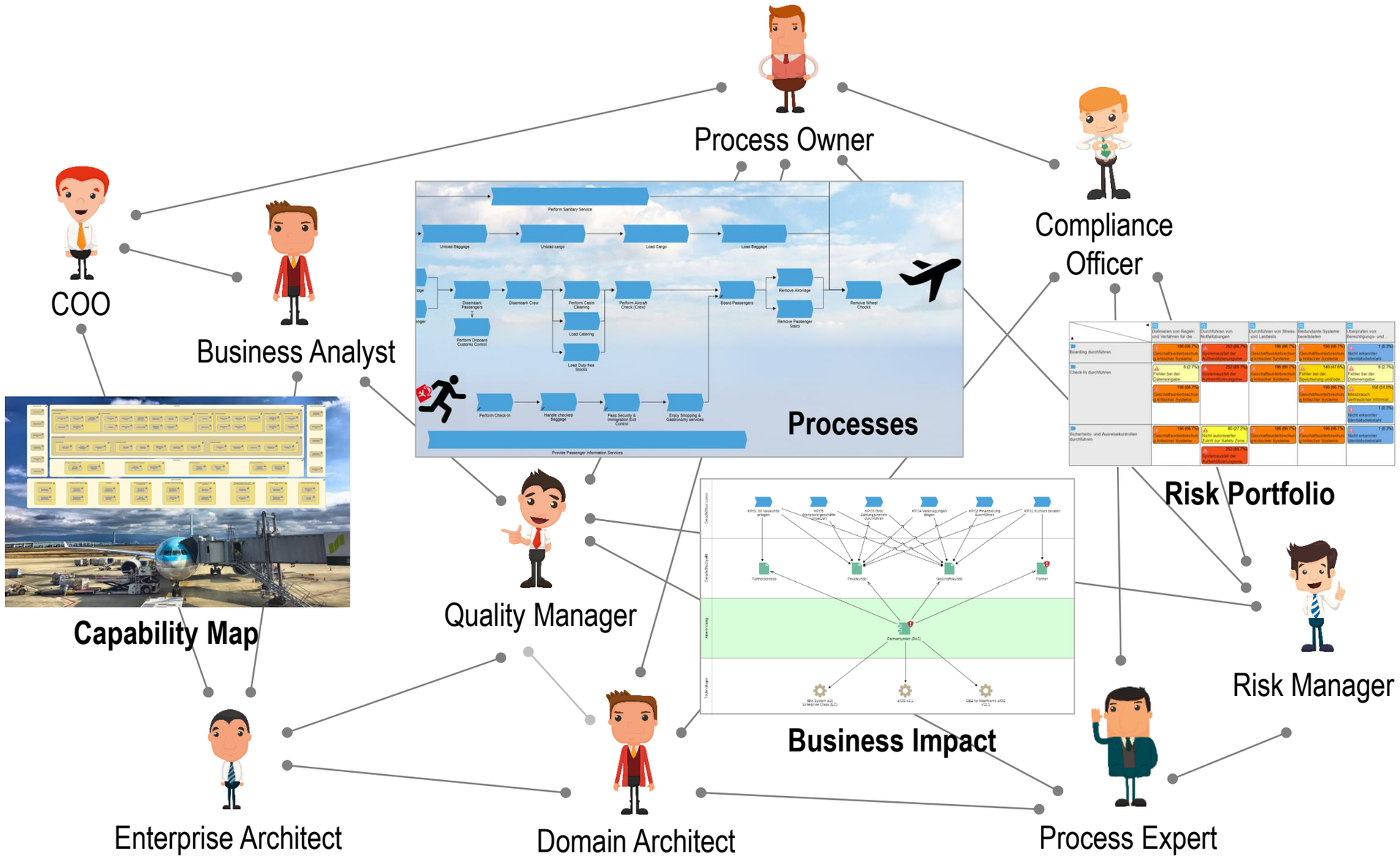
Roles	Fields of Interest
Managing Director	Agile Organization
	Development excellence
Head of Software Development	Clean Architecture
	DevOps
Cloud Services	Continuous Deployment
	Microservice Architecture
Group IT Services	Cyber Physical Systems

www.boc-group.com

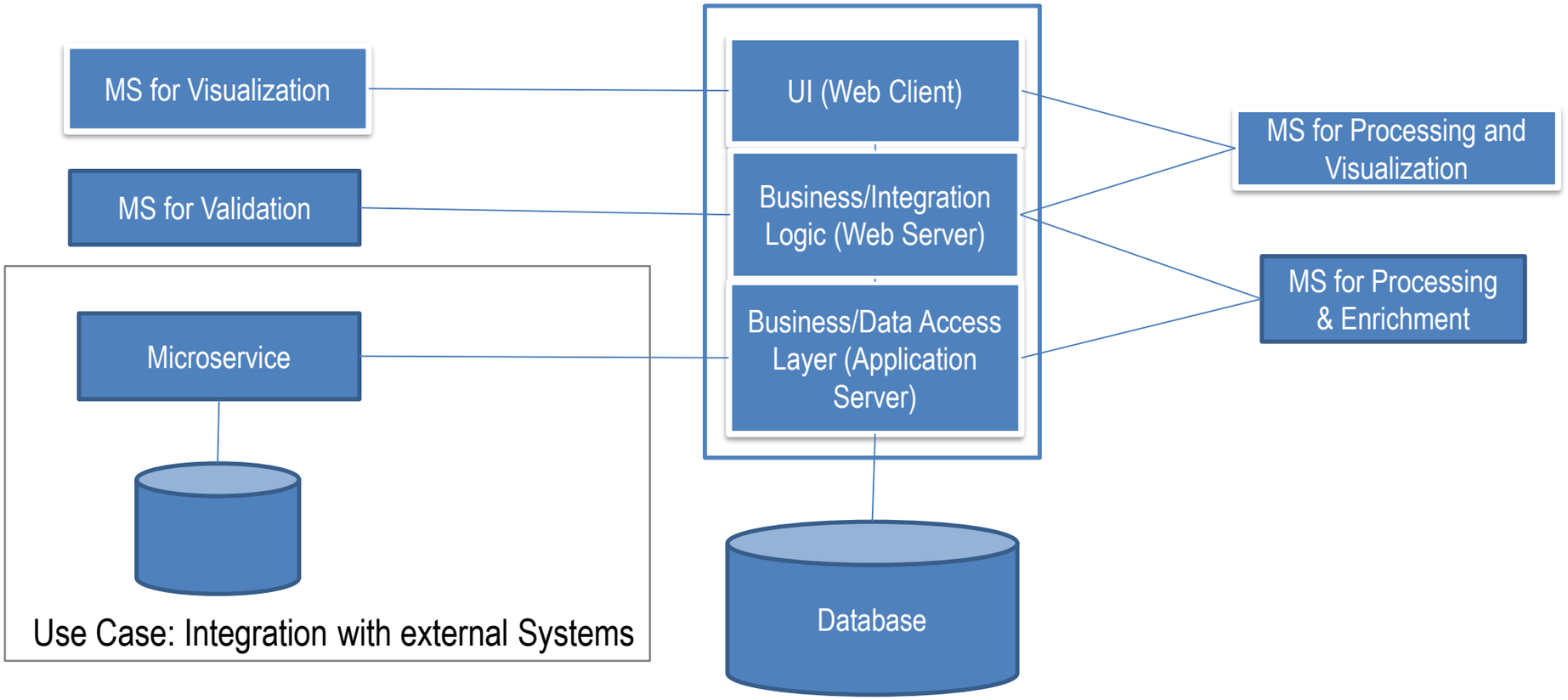
Agenda

- Application Domain
- Motivation to adopt MS & GraphQL
- Target Architecture
- Experience

Integrated Management Systems



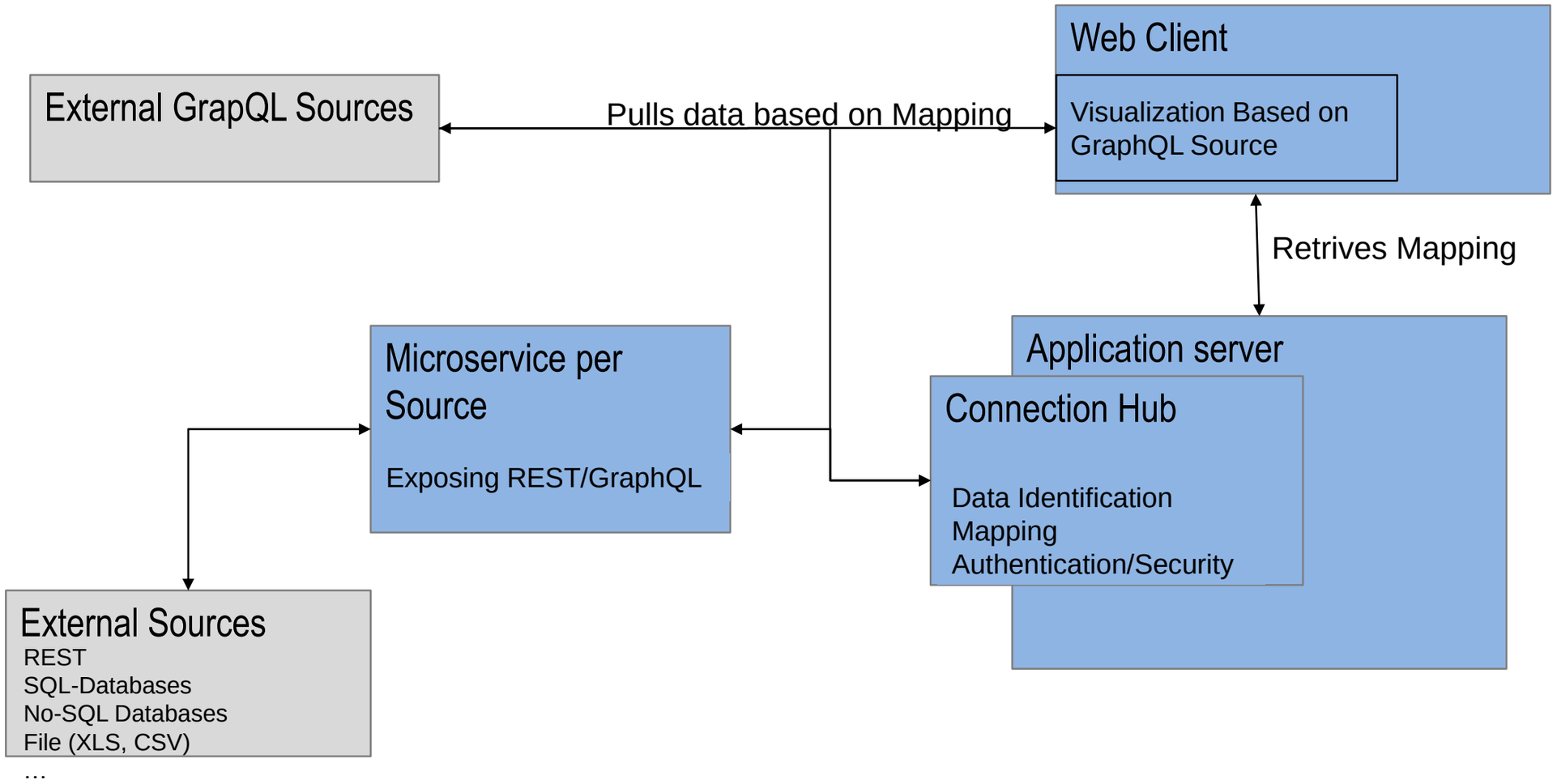
Architecture Evolution to Openness



The challenges to consider

- API Contract: The API/Implementation will change over time
- Transformation: The source system data needs to comply fully to the target data model.
- Data Duplication: Which system is the golden source for the information?
- Scaling: What happens if you integrate with one tool but from multiple variants/deployment?

Target Architecture for Integration



What we Learned

Benefits

- Discussing the relevant data is the GraphQL Schema notation
- The integration hub contract is stable
- Changing Details of the integration is configuration
- Versioning/Compatibility chaos resolved

Challenges

- The effort is in understanding the source systems structure
- The identification mapping is not always an automatic process
- Multiple integrations map to one element
- Building performant transformations
  - Non-Blocking & Request based Resource Caching

The next challenges

- Integrating multiple schemas into one view (schema stitching)
- Integrating on data intensive operations (e.g. search)
- Handling of mutation