



Digital Solution
Architecture

API
CONFERENCE
HYBRID EDITION

Microservice API Patterns

Step by Step

Daniel Lübke

daniel.luebke@digital-solution-architecture.com



About me

Dr.-Ing. Daniel Lübke

Expert for Software Architecture and
Business Process Execution

API as a Success Factor

Often you want to offer a service to be integrated by others

- e.g., SaaS: messaging, invoicing, ...
- e.g., internal systems that together support a business process

**Digital Transformation requires automation
between systems**

Good APIs can lead to better software integration

- Easy-to-use
- Better to maintain
- Cost-efficient

Help, I need to design an API!

May be helpful

Captured Experience

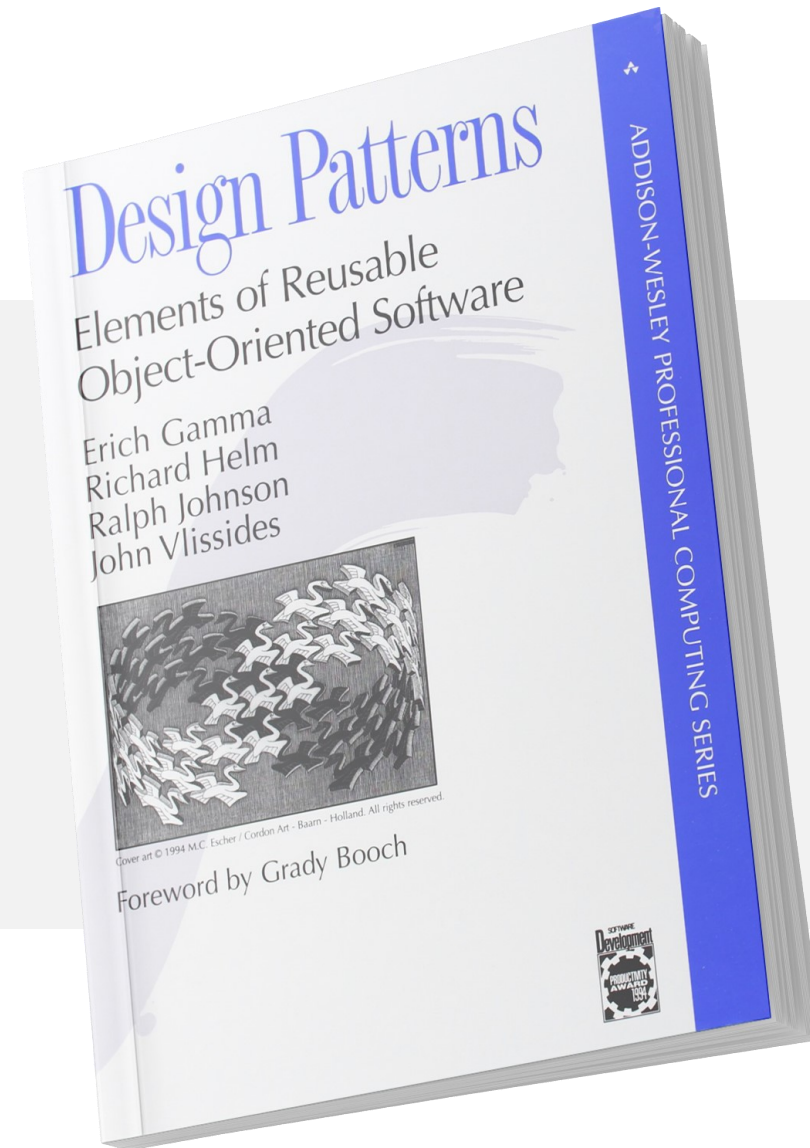
Pattern

Frequently Used

Context Dependent

Contents of **Patterns**

- ✓ **Patterns collect and document experiences of proven solutions**
- ✓ **Name** ✓ Applicability ✓ Structure
- ✓ **Consequences** (Benefits & Liabilities; Forces)
- ✓ Intent ✓ Known Uses ✓ **Motivation**
- ✓ Implementation ✓ Sample

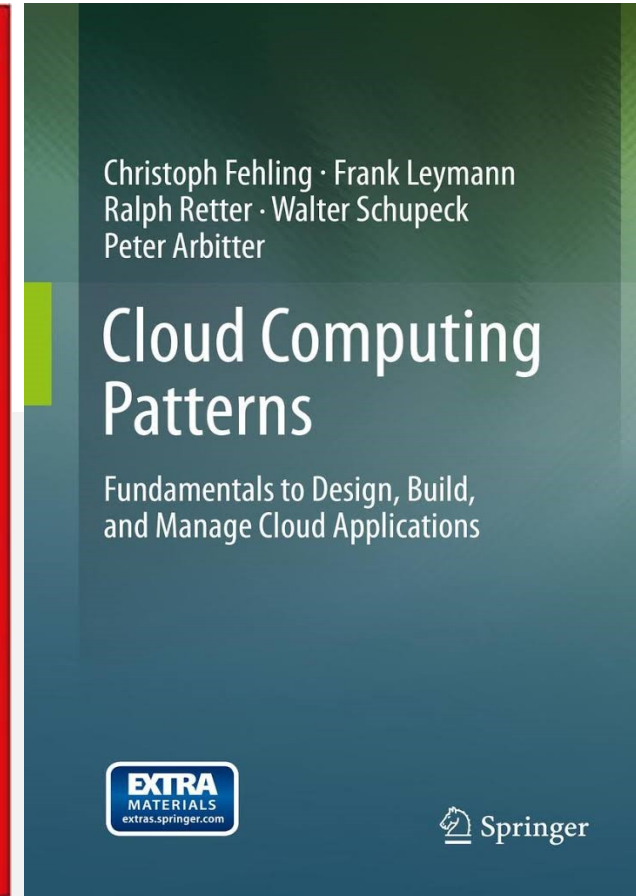
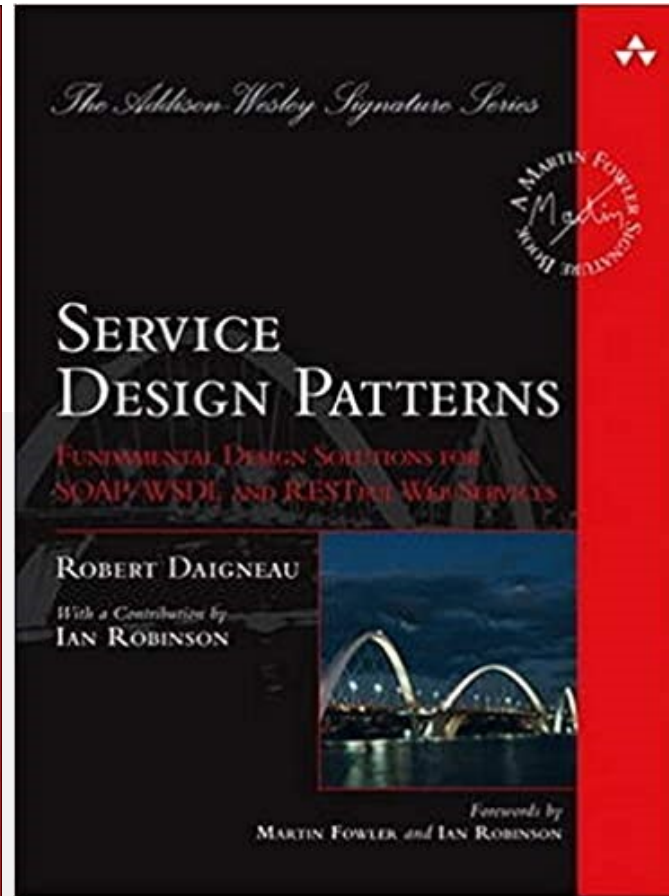
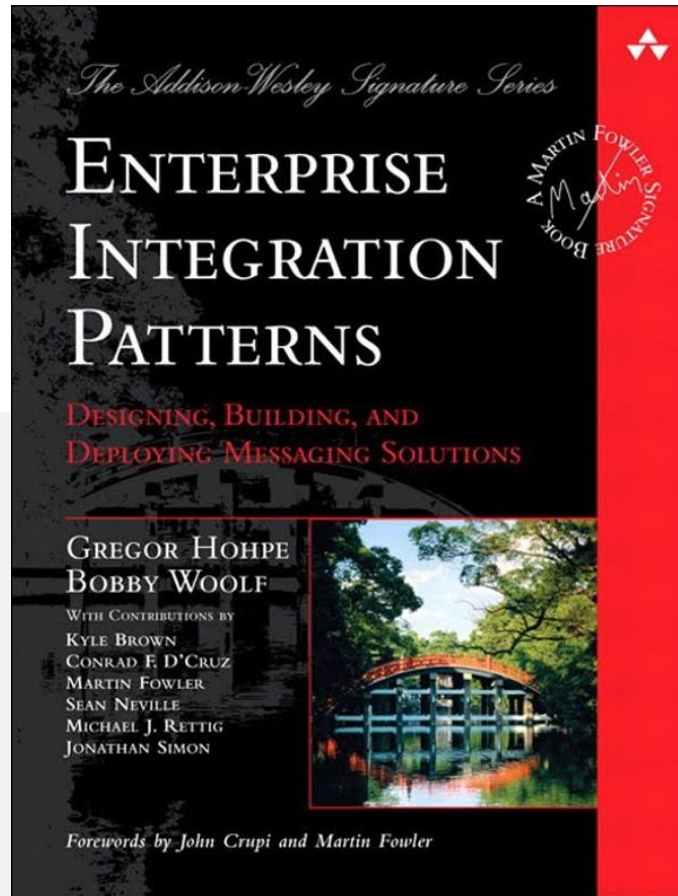


Our Example

A Query Service for Land Register Data

- Fetch Parcel data with associated data from a land registry system
 - Ownership
 - Rights
 - Mortgages
 - ...

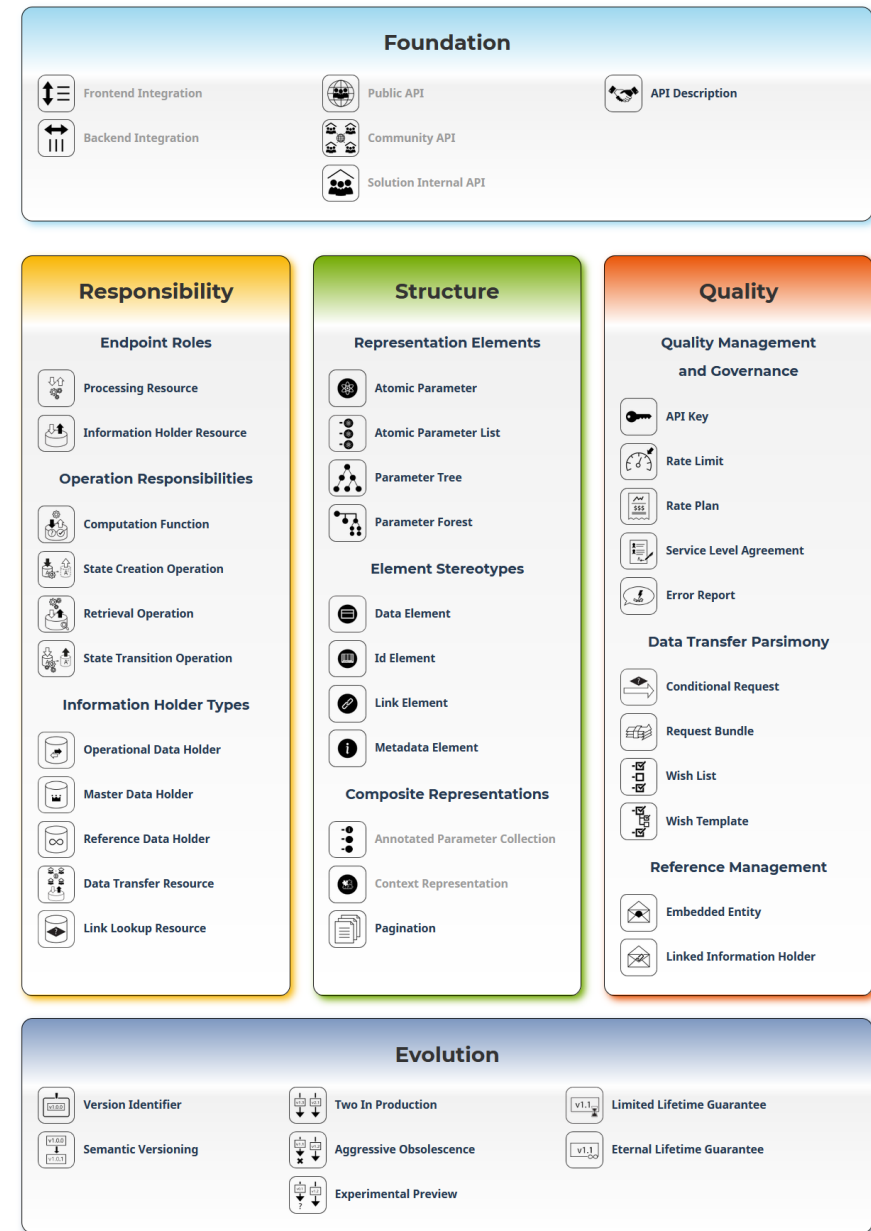
Related Patterns



Microservice API Patterns (MAP)

- New but not new
- Distilled solution patterns from many projects
 - Not invented by us but curated by us
- API-related concepts independent of concrete technology
 - CORBA, SOAP, REST, HTTP/JSON, GraphQL, gRPC, ..., MQ, Events, ...
- Currently 46 patterns in 5 categories

www.microservice-api-patterns.org



The Team



Olaf Zimmermann



Daniel Lübke



Mirko Stocker



Uwe Zdun



Cesare Pautasso

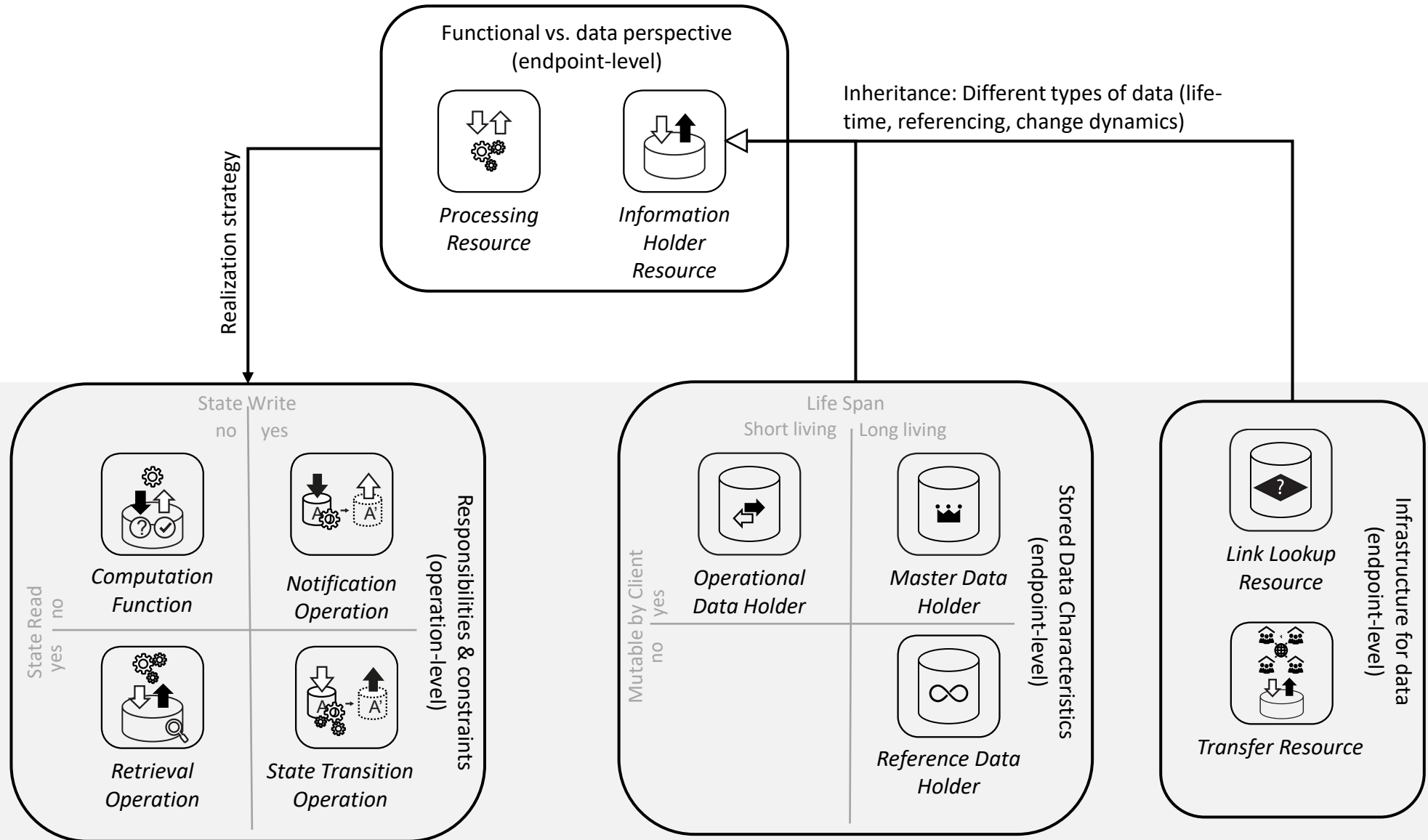
History of MAP



Walking through Your **Decisions**

- **Many Decisions** to be made
- A **Checklist** is always good
- Different **Dimensions**

What is my API (operation) supposed to do?

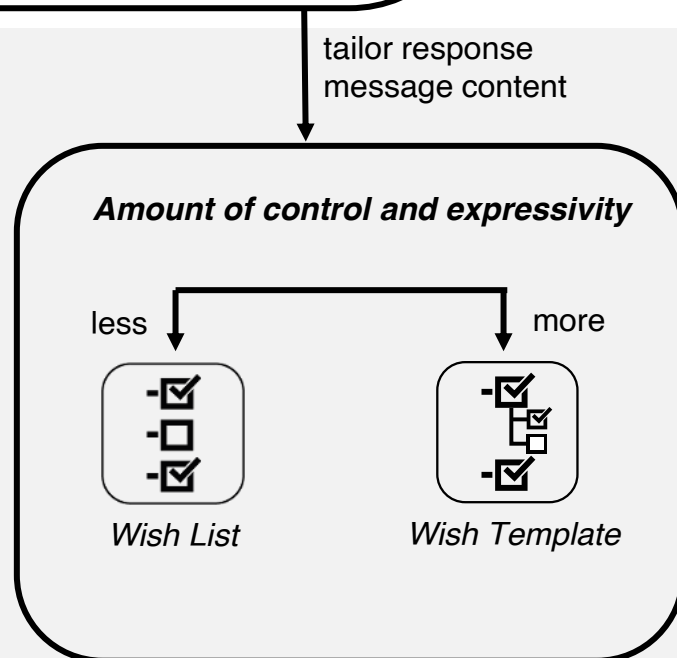
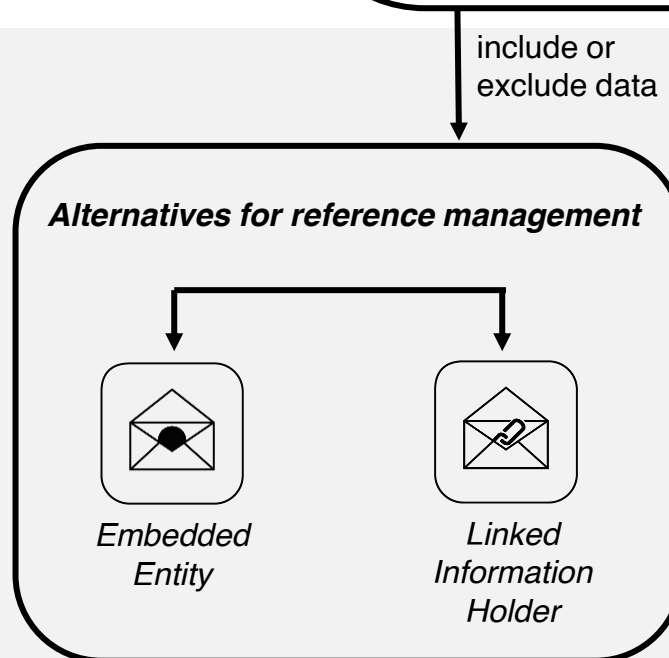
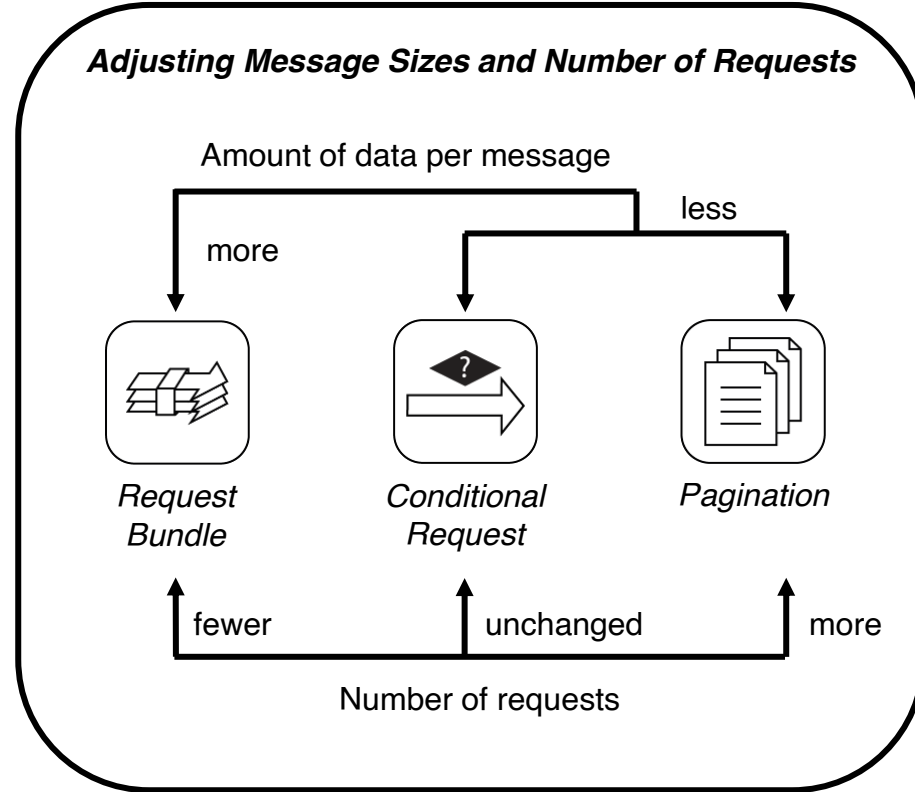


Our Example

A Query Service for Land Register Data

- Read Only (**Retrieval Operation**)
- **Master Data Holder** (quite static information)
 - Allows for Optimizations
- No further indirection or infrastructure

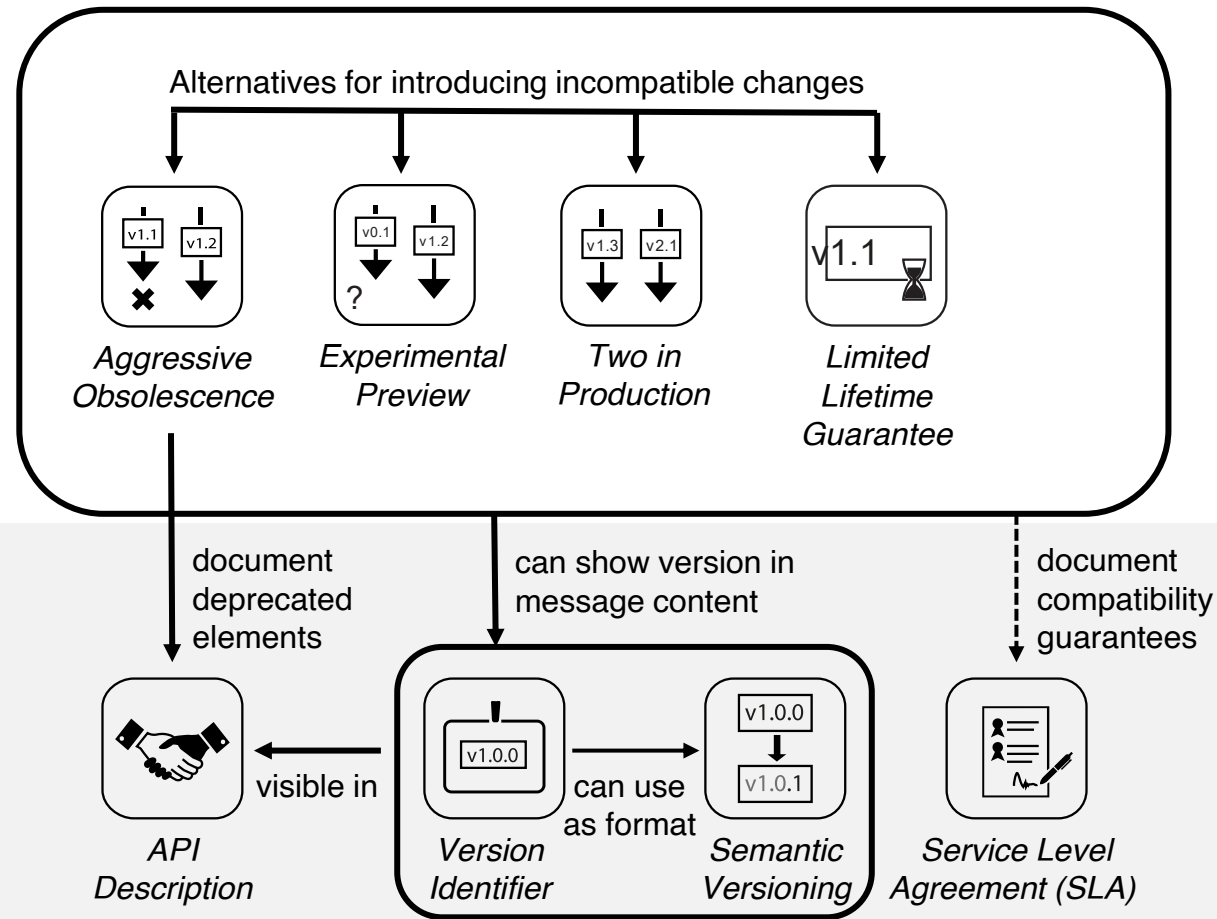
How **much** **data** do I transfer?



Our Example

- **Conditional Request**
 - Master Data Holder
- **Embedded Entities**
 - All data is transfered in one request
 - Less Requests
- **Wish List**
 - Pre-defined sets of attributes that are returned
 - History data
 - Details in structure

How do I **evolve** my API?



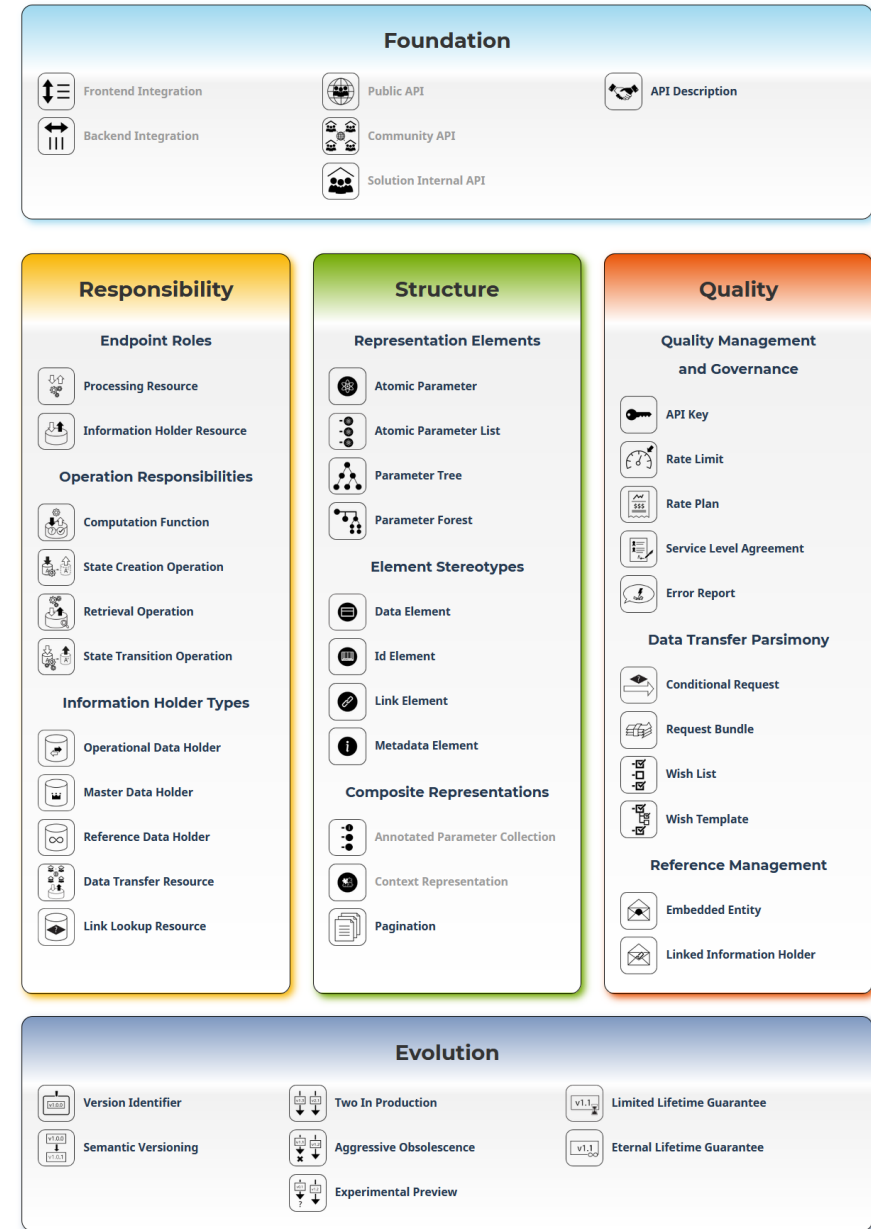
Our Example

- **Two in Production**
 - Less Agile Setting
 - No synchronized deployments
 - Long leap times
- **Version Identifier**
 - Informational only
 - Debugging and diagnostic information
- **Semantic Versioning**
 - Signals compatibility and migration effort

Where to **look**?

- Microservice API Patterns are
 - the foundation to a shared language concerning API design
 - a resource for necessary and possible design decisions
 - a library for implementation hints
- Visit <https://www.microservice-api-patterns.org>
- Happily, use and comment on our content!
- **Let's establish a common language in API design!**

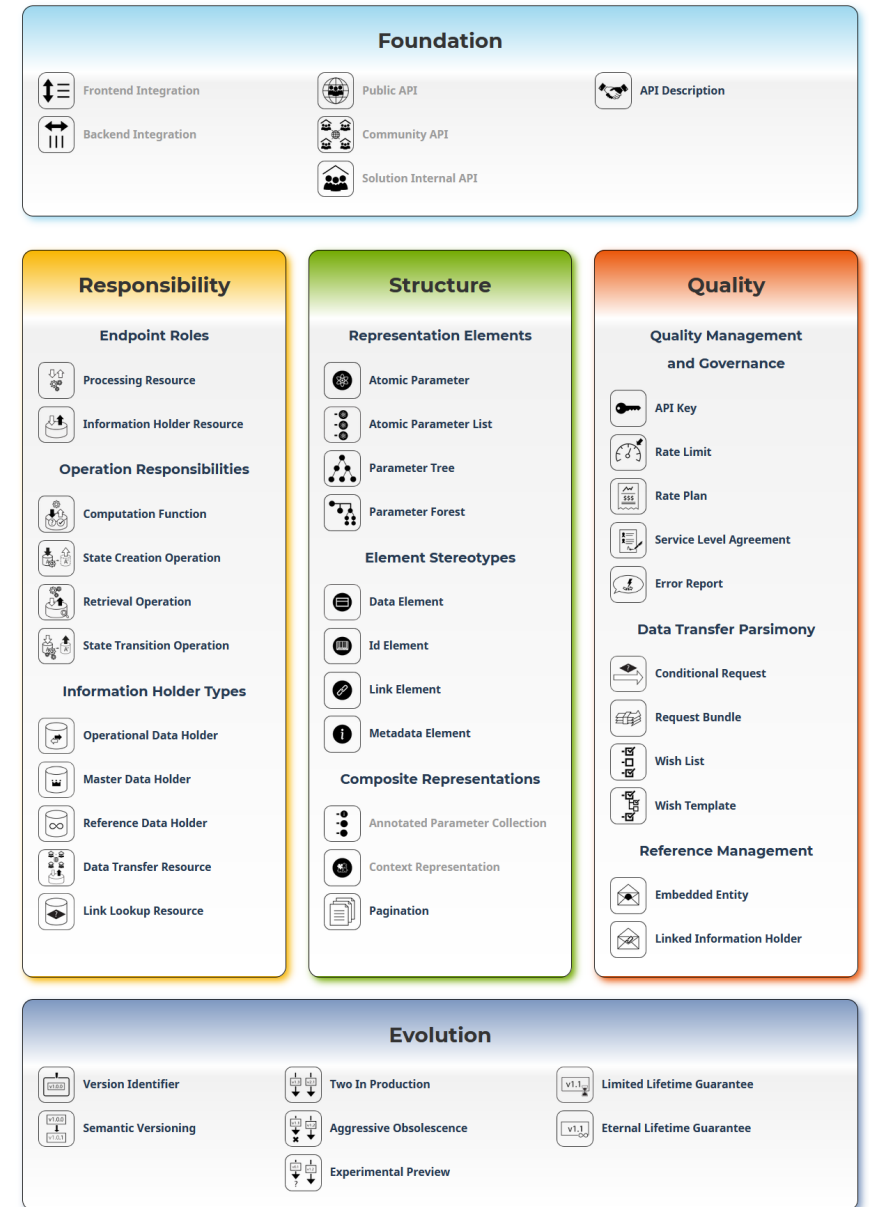
Thank you for your attention! **Any Questions?**
daniel.luebke@digital-solution-architecture.com



Outlook

- There is more to come
- However, currently **not yet official** 😞
- Register to get **notified**
- We try to work out some **advantages** for you

<https://www.digital-solution-architecture.com/map/>



Thank you for your attention! **Any Questions?**

daniel.luebke@digital-solution-architecture.com