tfz ICT
Cluster Insights

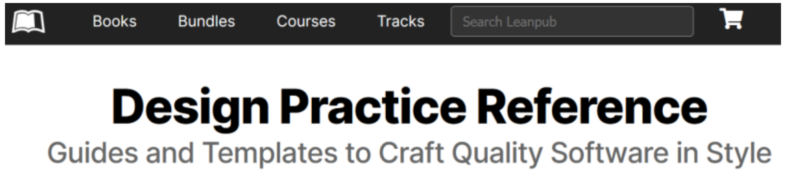
«Microservices API Patterns – wie ein „Musterschüler" entstehen kann»

Referent:

Prof. Dr. Olaf Zimmermann (ZIO)
Professor für Software-Architektur
OST – Ostschweizer Fachhochschule

Who is ZIO?

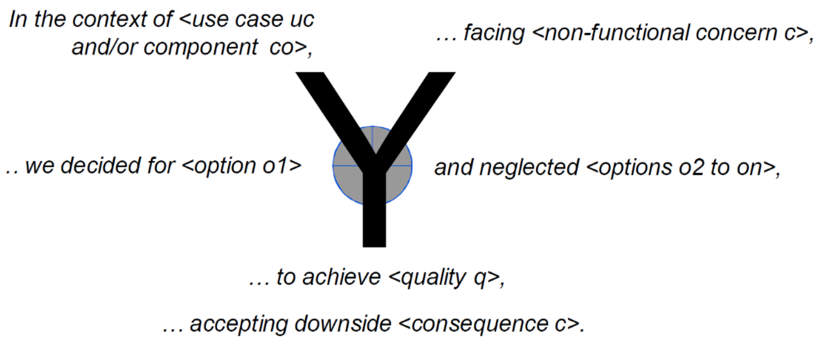
- Software Architect
- Lecturer, Author, Blogger
- Open Sourcer
  - Context Mapper (DSL for Domain-Driven Design)
  - Interface Refatoring Catalog
  - Y-Statements as a compact form of ADRs
  - Markdown ADRs, API description MDSL
  - Lakeside Mutual microservices (sample application)



Design Practice Reference

Guides and Templates to Craft Quality Software in Style

https://leanpub.com/dpr



In the context of <use case uc and/or component co>,

… facing <non-functional concern c>,

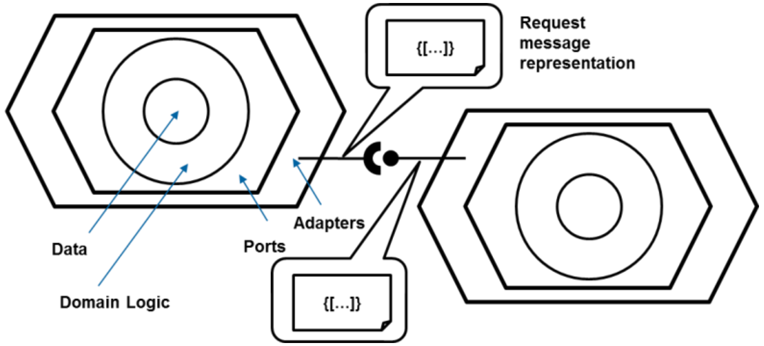
.. we decided for <option o1>

and neglected <options o2 to on>,

… to achieve <quality q>,

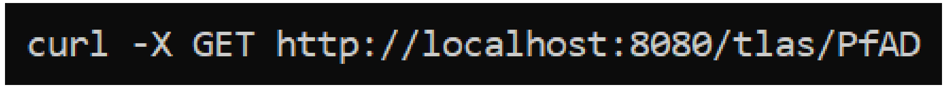
… accepting downside <consequence c>.

https://medium.com/olzzio



Motivating Example: TLA Resolver

- HTTP POST, GET, … to a Java Spring Boot application resolving acronyms
  - *In:* URI with path and query parameters



```
curl -X GET http://localhost:8080/tlas/PfAD
```

```
// MAP: RetrievalOperation
@GetMapping(value="/tlas/{theme}", produces="application/json")
@ResponseBody // MAP: Parameter Tree (nested)
public String resolveAll(@PathVariable String theme) { ... }
```

  - *Out:* plain text or JSON document (with embedded objects and arrays)

```
{ "acronyms": {
    "MAP": {
      "shortName": "MAP", "fullName": "Microservice API Patterns",
      "alternativeMeanings": [ "Must adopt patterns", "Messaging API Patterns" ],
      "link": "https://microservice-api-patterns.org/", "status": "common" }}}
```

API Design Myths

- **Myth 1:**
  As many program-internal features as possible should be exposed as APIs.
- **Myth 2:**
  All program-internal data structures should be passed through (as is).
- **Myth 3:**
  One size (of API operations, of request/response messages) fits all clients.
- **Myth 4:**
  HTTP is the only protocol needed, a de-facto standard for program integration.
- **Myth 5:**
  REST is a technology, and "REST API" is a correct and complete term.

API Design Mythbusting

- **Myth 1:** As many program features as possible should be exposed as APIs.
  Response: An API first approach is recommended, from client goals to API design.
- **Myth 2:** All program-internal data structures should be passed through (as is).
  Response: A wrap-and-map strategy hides implementation details and promotes evolvability. API design patterns explain how to wrap and map.
- **Myth 3:** One size (of API operations and their messages) fits all clients.
  Response: Clients are different, patterns exist to let them drive the response shapes.
- **Myth 4:** HTTP is the only protocol needed.
  Response: Queue-based messaging is recommended for backend integration.
- **Myth 5:** REST is a technology, and "REST API" is a correct and complete term.
  Response: It is an architectural style; say "HTTP resource API" if HATEOAS is not used

API Design is Simple?!

Complexity of systems

Project pressure, legacy constraints

Quality goals, security needs

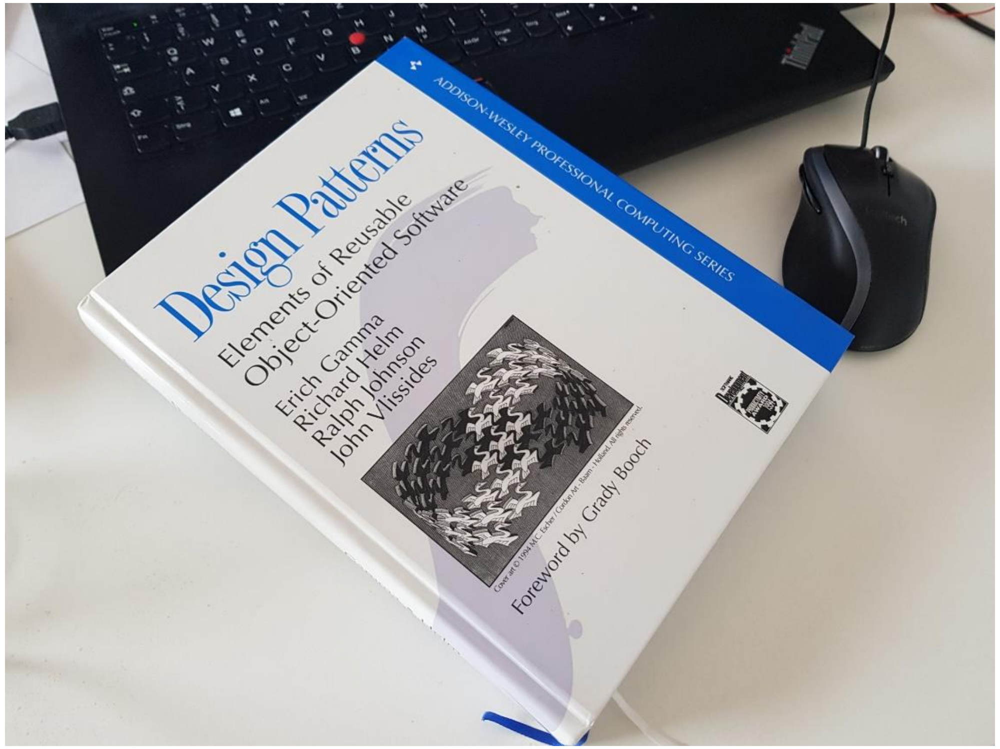
What are the urgent and important questions?

Architectural knowledge helps, even the experienced!
Patterns structure problem and solution domain
and can offer context-specific design guidance.

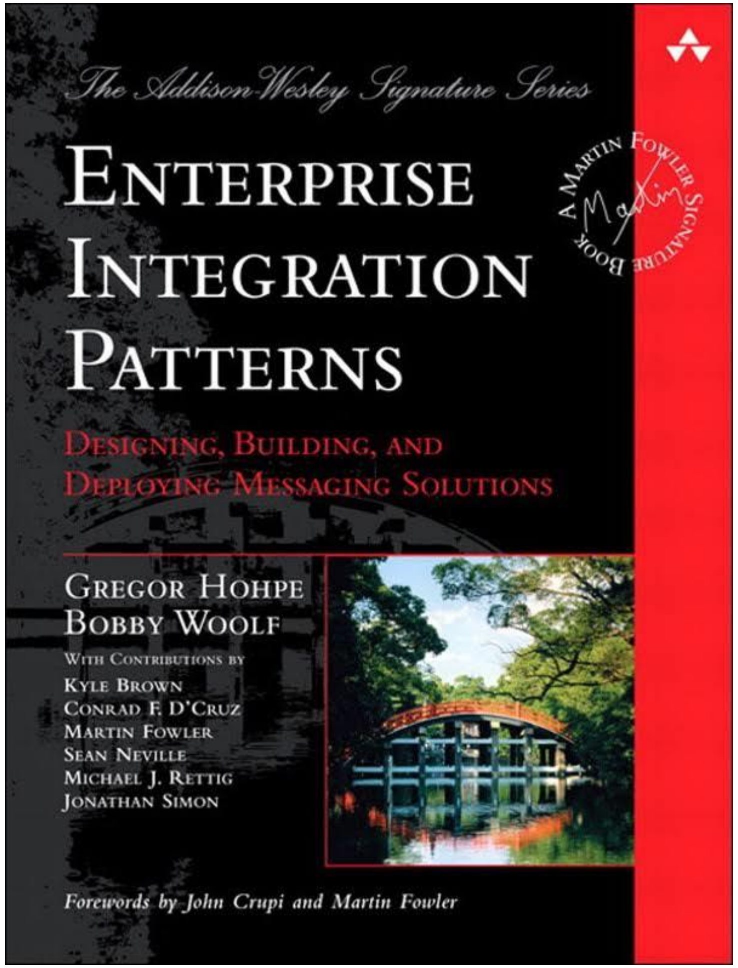
Why Patterns?

- Patterns collect and document experiences with proven solutions

- Many templates:
  - – Name, Icon
  - – Context: Intent, Motivation and Applicability
  - – Solution Structure and its Forces
  - – Consequences: Benefits and Liabilities
  - – Examples, Implementation Hints
  - – Pointers to other Patterns
  - – Known Uses


ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES
Design Patterns
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch

Patterns Relevant for API Design



The Addison-Wesley Signature Series

ENTERPRISE INTEGRATION PATTERNS

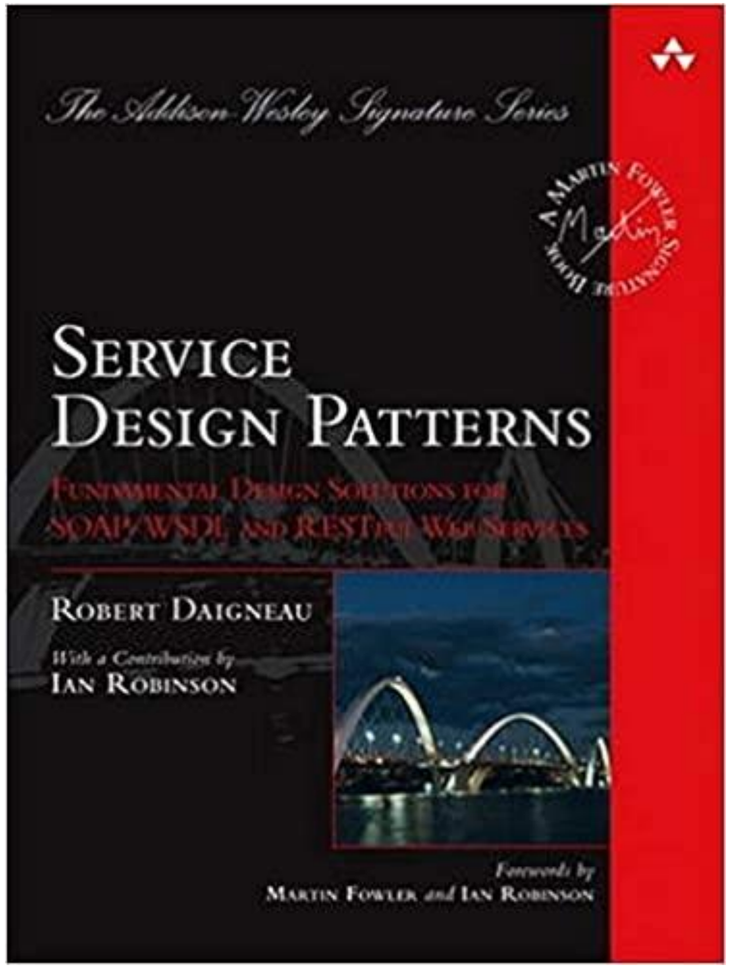
DESIGNING, BUILDING, AND DEPLOYING MESSAGING SOLUTIONS

A MARTIN FOWLER SIGNATURE BOOK

GREGOR HOHPE
BOBBY WOOLF

WITH CONTRIBUTIONS BY
KYLE BROWN
CONRAD F. D'CRUZ
MARTIN FOWLER
SEAN NEVILLE
MICHAEL J. RETTIG
JONATHAN SIMON

Forewords by John Crupi and Martin Fowler



The Addison-Wesley Signature Series

A MARTIN FOWLER SIGNATURE BOOK

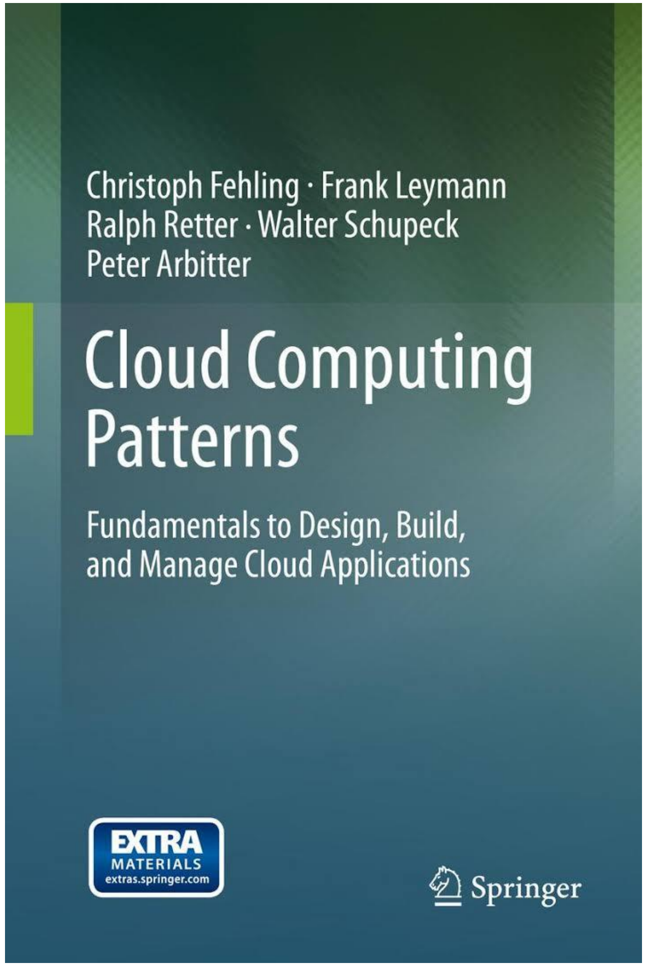
SERVICE DESIGN PATTERNS

FUNDAMENTAL DESIGN SOLUTIONS FOR SOAP/WSDL AND RESTFUL WEB SERVICES

ROBERT DAIGNEAU

With a Contribution by
IAN ROBINSON

Forewords by
MARTIN FOWLER *and* IAN ROBINSON



Christoph Fehling · Frank Leymann
Ralph Retter · Walter Schupeck
Peter Arbitter

Cloud Computing Patterns

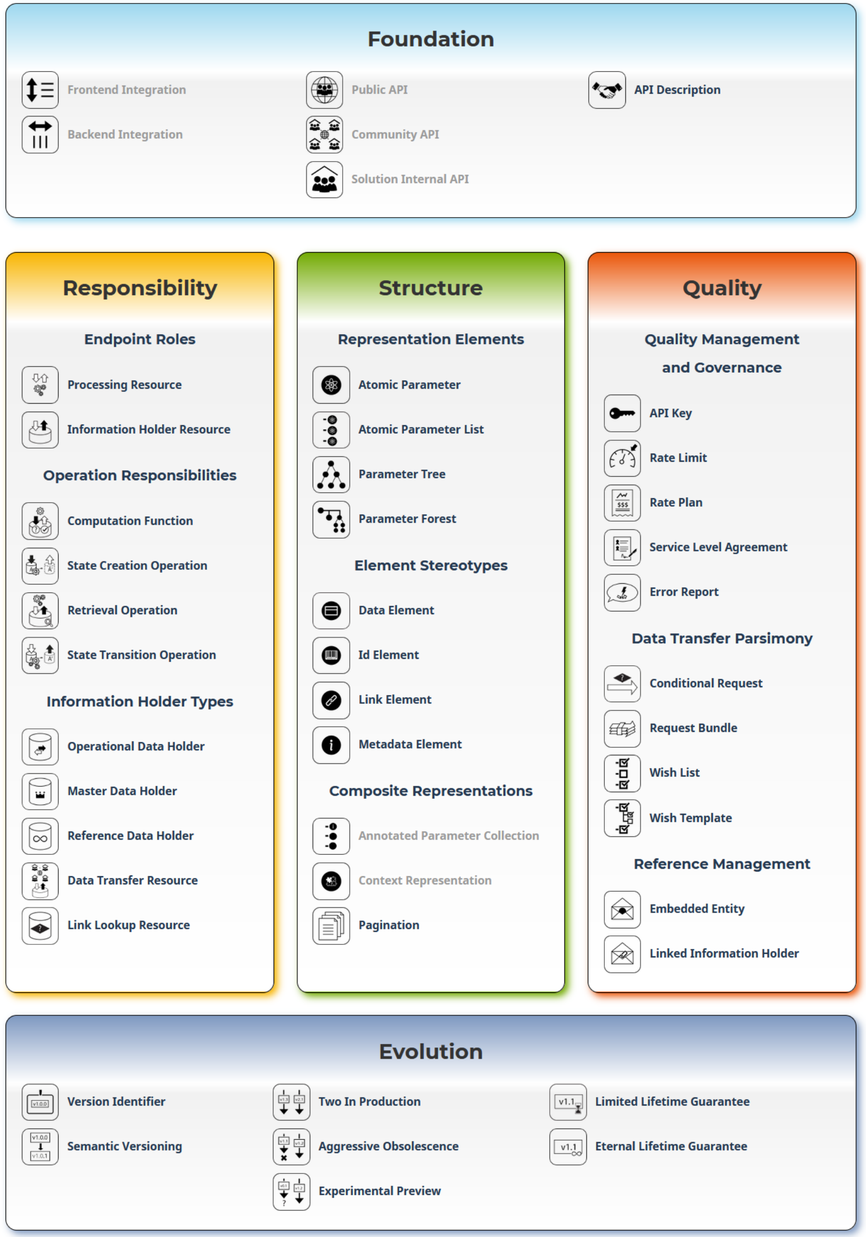
Fundamentals to Design, Build, and Manage Cloud Applications

EXTRA MATERIALS extras.springer.com

Springer

New (2016-2022): Microservice API Patterns

- Distilled solution patterns from projects
  - Not invented, but curated (5 authors)
  - 45 patterns in 5 categories (focus: architectural roles, data structures)
- API-related concepts independent of concrete technology:
  - RESTful HTTP, SOAP
  - GraphQL, gRPC, CORBA
  - Messaging systems, event streaming



Foundation

Frontend Integration
Backend Integration
Public API
Community API
Solution Internal API
API Description

Responsibility

Endpoint Roles
Processing Resource
Information Holder Resource

Operation Responsibilities
Computation Function
State Creation Operation
Retrieval Operation
State Transition Operation

Information Holder Types
Operational Data Holder
Master Data Holder
Reference Data Holder
Data Transfer Resource
Link Lookup Resource

Structure

Representation Elements
Atomic Parameter
Atomic Parameter List
Parameter Tree
Parameter Forest

Element Stereotypes
Data Element
Id Element
Link Element
Metadata Element

Composite Representations
Annotated Parameter Collection
Context Representation
Pagination

Quality

Quality Management and Governance
API Key
Rate Limit
Rate Plan
Service Level Agreement
Error Report

Data Transfer Parsimony
Conditional Request
Request Bundle
Wish List
Wish Template

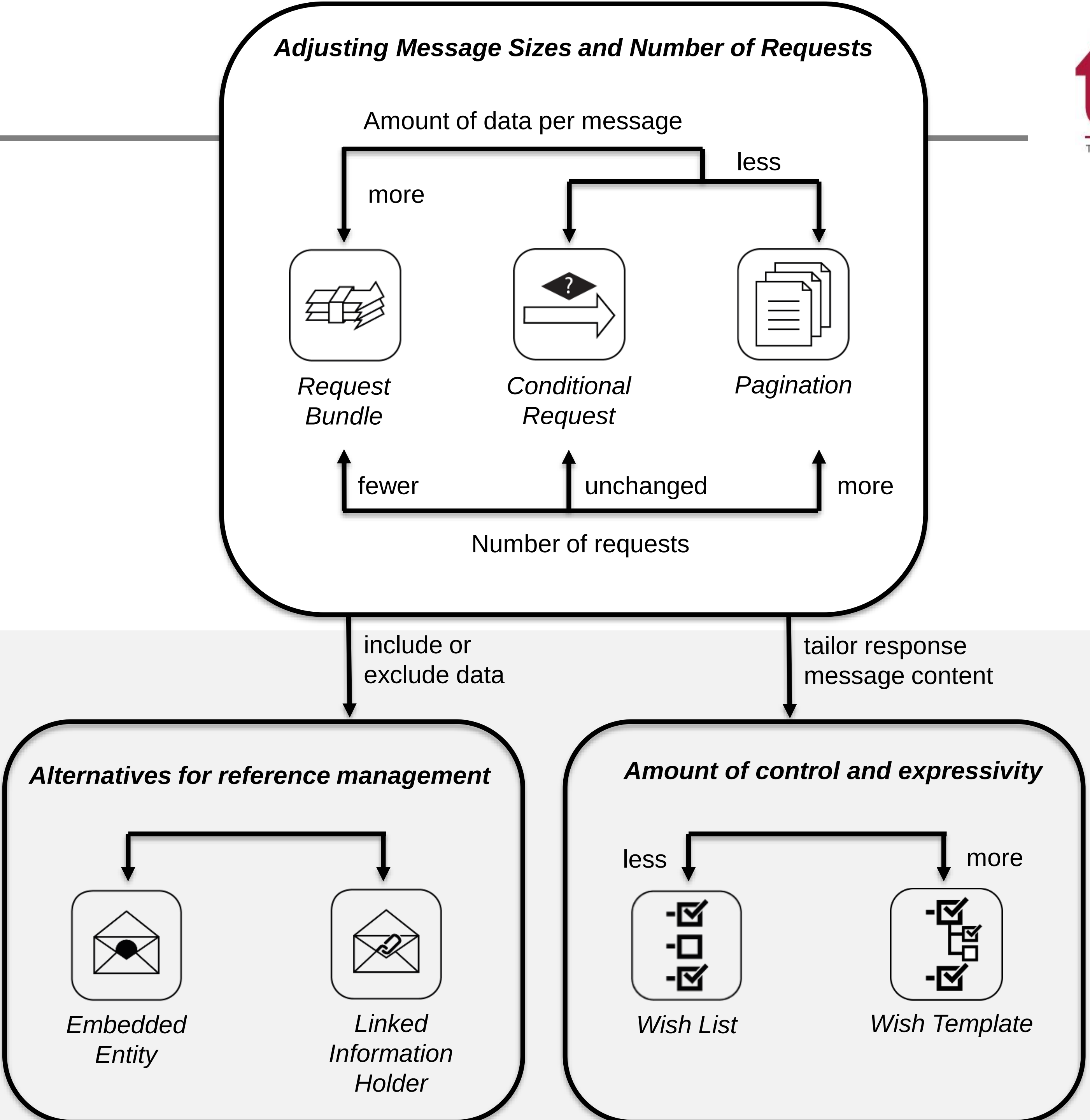
Reference Management
Embedded Entity
Linked Information Holder

Evolution

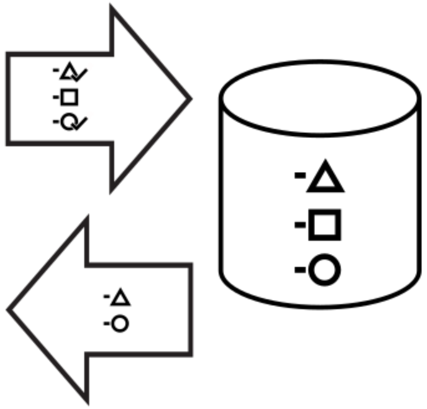
Version Identifier
Semantic Versioning
Two In Production
Aggressive Obsolescence
Experimental Preview
Limited Lifetime Guarantee
Eternal Lifetime Guarantee

www.microservice-api-patterns.org

How much data do API client and API provider exchange? And how often?


Adjusting Message Sizes and Number of Requests
Amount of data per message
more
less
Request Bundle
Conditional Request
Pagination
fewer
unchanged
more
Number of requests
include or exclude data
tailor response message content
Alternatives for reference management
Embedded Entity
Linked Information Holder
Amount of control and expressivity
less
more
Wish List
Wish Template

Sample Pattern: Wish List (1/2)



- **Problem:**
  How can an API client inform the API provider at runtime about the data it is interested in?
- **Forces:**
  Client diversity, message size vs. number of messages, endpoint complexity
- **Solution:**
  As an API client, provide a WISH LIST in the request that enumerates all desired data elements of the requested resource.

https://microservice-api-patterns.org/patterns/quality/dataTransferParsimony/WishList

© Olaf Zimmermann, 2022

Sample Pattern: Wish List (2/2)

- **Known uses:** Found in many Web and product APIs, e.g. Atlassian Jira
  - Variations: expansion, wild cards (*), query expression (GraphQL!)
  - Alternative: WISH TEMPLATE (structured, mock object rather than flat name list)

- **Example:** Lakeside Mutual Customer Core microservice

```
curl -X GET http://localhost:8080/customers/gktlipwhjr?fields=customerId,birthday,postalCode

{
  "customerId": "gktlipwhjr",
  "birthday": "1989-12-31T23:00:00.000+0000",
  "postalCode": "8640"
}
```

https://microservice-api-patterns.org/patterns/quality/dataTransferParsimony/WishList

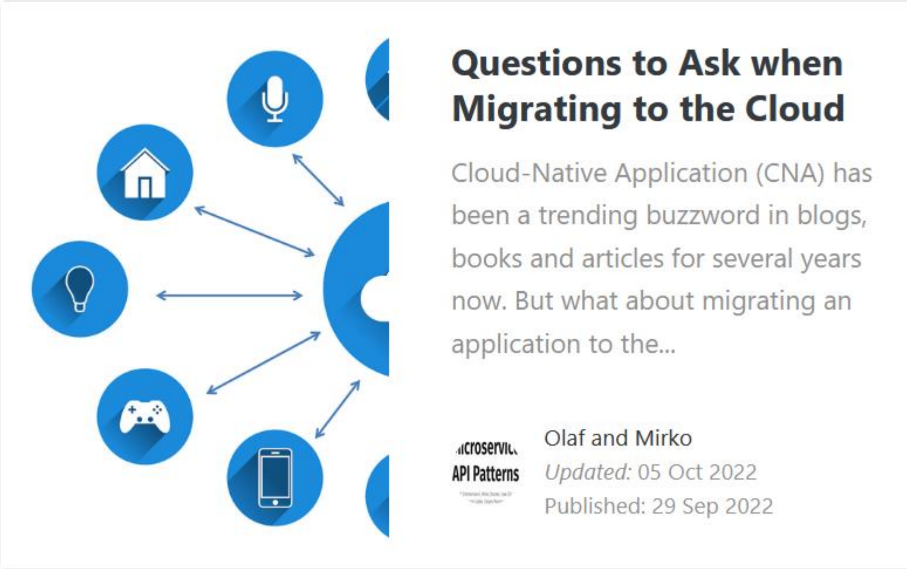
Discussion

- Do the patterns names work for you?
- Do the presented solutions make sense?
- Which patterns might be missing?

- Pros and cons of the options for API design/architecture decisions
- Sample scenarios

More Information: api-patterns.org and blogs

- https://ozimmer.ch/blog/ und (kürzer, weniger) https://docsoc.medium.com/



Questions to Ask when Migrating to the Cloud

Cloud-Native Application (CNA) has been a trending buzzword in blogs, books and articles for several years now. But what about migrating an application to the...

Olaf and Mirko
Updated: 05 Oct 2022
Published: 29 Sep 2022

https://microservice-api-patterns.org/publications

Introduction to Microservice API Patterns (MAP)

Olaf Zimmermann
University of Applied Sciences of Eastern Switzerland, Rapperswil, Switzerland
ozimmerm@hsr.ch

Mirko Stocker
University of Applied Sciences of Eastern Switzerland, Rapperswil, Switzerland
mirko.stocker@hsr.ch

Daniel Lübke
iQuest GmbH, Hanover, Germany
ich@daniel-luebke.de

Cesare Pautasso
Software Institute, Faculty of Informatics, USI Lugano, Switzerland
c.pautasso@ieee.org

Uwe Zdun
University of Vienna, Faculty of Computer Science, Software Architecture Resear
Vienna, Austria
uwe.zdun@univie.ac.at

Abstract

The Microservice API Patterns (MAP) language and supporting website premiered at Microservices 2019. MAP distills proven, platform- and technology-indepen recurring (micro-)service design and interface specification problems such as fi service granularities, rightsizing message representations, and managing the evolu their implementations. In this paper, we motivate the need for such a pattern l the language organization and present two exemplary patterns describing altern representing nested data. We also identify future research and development direc

2012 ACM Subject Classification Software and its engineering → Patterns; Softw eering → Designing software

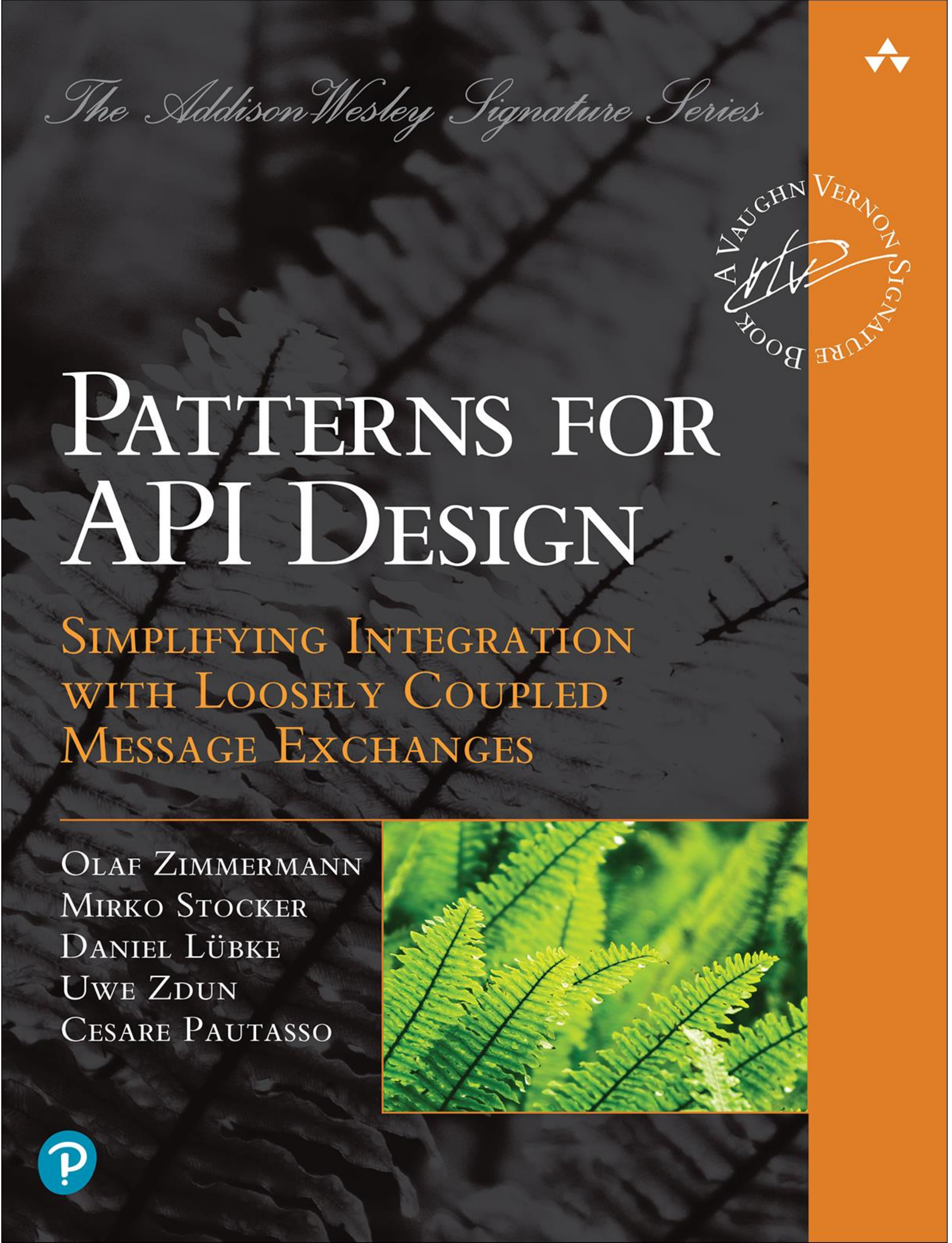
Keywords and phrases application programming interfaces, distributed systems, enterprise application integration, service-oriented computing, software architecture

Digital Object Identifier 10.4230/OASIcs.Microservices.2017-2019.4

Microservice API Patterns: A Language for API Design

1.1K views • 8 months ago

https://www.youtube.com/watch?v=cNp7ys0g0Bs


The Addison-Wesley Signature Series
A Vaughn Vernon Signature Book
Patterns for API Design
Simplifying Integration with Loosely Coupled Message Exchanges
Olaf Zimmermann
Mirko Stocker
Daniel Lübke
Uwe Zdun
Cesare Pautasso

Available December 2022

Pre-order & Save 35%* on eBook at informit.com/api-patterns

- Use code **API-PATTERNS** during checkout
- Offer only good at informit.com
- eBook – DRM-Free PDF & EPUB

Print books available**
Please check your local or online store where you purchase technical related books. Note – print books should be available in Europe in January 2023.

*Discount code API-PATTERNS is only good at informit.com and cannot be used on the already discounted book + eBook bundle or combined with any other offer. Discount offer is subject to change.

**If your order print books from InformIT, your order is subject to import duties and taxes, which are levied once the package reaches the destination country.