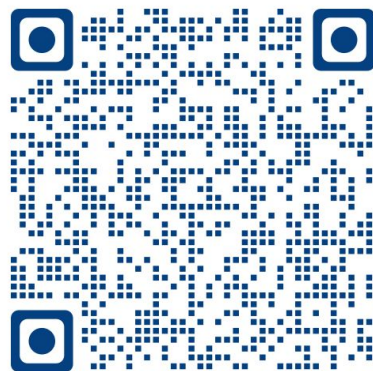




@codecentric

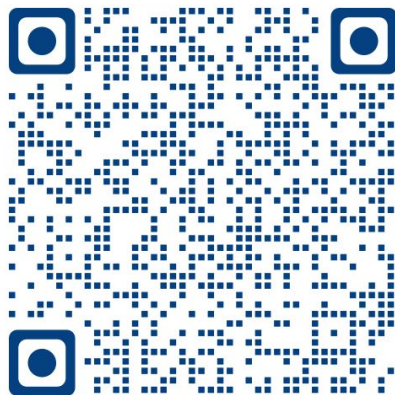
WAVESTONE

# Crafting Great APIs with Domain-Driven Design



**Architect at Wavestone**

**interested in API Design, Event Design,  
Microservices and how to make a developer's life  
easier**



Architect at codecentric

Interested in Microservices, DDD, API Design,  
Event Design and how to get faster from  
specification to code

# Your API is lying to you

```
1 # payload: {id: 1}
2 let url = new URL('https://slack.com/api/conversations?channel=...')
3 let headers = {
4   'Content-Type': 'application/json',
5   'Authorization': 'Bearer ...',
6 }
7 let body = {
8   'channel': '...',
9   'text': '...',
10 }
11 let options = {
12   method: 'POST',
13   headers,
14   body: JSON.stringify(body),
15 }
16 let response = await fetch(url, options)
17 let json = await response.json()
18 console.log(json)
19 // {
20 //   "ok": true,
21 //   "channel": "...",
22 //   "text": "..."
23 // }
```

```
1 // ...
2 // ...
3 // ...
4 // ...
5 // ...
6 // ...
7 // ...
8 // ...
9 // ...
10 // ...
11 // ...
12 // ...
13 // ...
14 // ...
15 // ...
16 // ...
17 // ...
18 // ...
19 // ...
20 // ...
21 // ...
22 // ...
23 // ...
24 // ...
25 // ...
26 // ...
27 // ...
28 // ...
29 // ...
30 // ...
31 // ...
32 // ...
33 // ...
34 // ...
35 // ...
36 // ...
37 // ...
38 // ...
39 // ...
40 // ...
41 // ...
42 // ...
43 // ...
44 // ...
45 // ...
46 // ...
47 // ...
48 // ...
49 // ...
50 // ...
51 // ...
52 // ...
53 // ...
54 // ...
55 // ...
56 // ...
57 // ...
58 // ...
59 // ...
60 // ...
61 // ...
62 // ...
63 // ...
64 // ...
65 // ...
66 // ...
67 // ...
68 // ...
69 // ...
70 // ...
71 // ...
72 // ...
73 // ...
74 // ...
75 // ...
76 // ...
77 // ...
78 // ...
79 // ...
80 // ...
81 // ...
82 // ...
83 // ...
84 // ...
85 // ...
86 // ...
87 // ...
88 // ...
89 // ...
90 // ...
91 // ...
92 // ...
93 // ...
94 // ...
95 // ...
96 // ...
97 // ...
98 // ...
99 // ...
100 // ...
```

```
1 // ...
2 // ...
3 // ...
4 // ...
5 // ...
6 // ...
7 // ...
8 // ...
9 // ...
10 // ...
11 // ...
12 // ...
13 // ...
14 // ...
15 // ...
16 // ...
17 // ...
18 // ...
19 // ...
20 // ...
21 // ...
22 // ...
23 // ...
24 // ...
25 // ...
26 // ...
27 // ...
28 // ...
29 // ...
30 // ...
31 // ...
32 // ...
33 // ...
34 // ...
35 // ...
36 // ...
37 // ...
38 // ...
39 // ...
40 // ...
41 // ...
42 // ...
43 // ...
44 // ...
45 // ...
46 // ...
47 // ...
48 // ...
49 // ...
50 // ...
51 // ...
52 // ...
53 // ...
54 // ...
55 // ...
56 // ...
57 // ...
58 // ...
59 // ...
60 // ...
61 // ...
62 // ...
63 // ...
64 // ...
65 // ...
66 // ...
67 // ...
68 // ...
69 // ...
70 // ...
71 // ...
72 // ...
73 // ...
74 // ...
75 // ...
76 // ...
77 // ...
78 // ...
79 // ...
80 // ...
81 // ...
82 // ...
83 // ...
84 // ...
85 // ...
86 // ...
87 // ...
88 // ...
89 // ...
90 // ...
91 // ...
92 // ...
93 // ...
94 // ...
95 // ...
96 // ...
97 // ...
98 // ...
99 // ...
100 // ...
```

```
1 // ...
2 // ...
3 // ...
4 // ...
5 // ...
6 // ...
7 // ...
8 // ...
9 // ...
10 // ...
11 // ...
12 // ...
13 // ...
14 // ...
15 // ...
16 // ...
17 // ...
18 // ...
19 // ...
20 // ...
21 // ...
22 // ...
23 // ...
24 // ...
25 // ...
26 // ...
27 // ...
28 // ...
29 // ...
30 // ...
31 // ...
32 // ...
33 // ...
34 // ...
35 // ...
36 // ...
37 // ...
38 // ...
39 // ...
40 // ...
41 // ...
42 // ...
43 // ...
44 // ...
45 // ...
46 // ...
47 // ...
48 // ...
49 // ...
50 // ...
51 // ...
52 // ...
53 // ...
54 // ...
55 // ...
56 // ...
57 // ...
58 // ...
59 // ...
60 // ...
61 // ...
62 // ...
63 // ...
64 // ...
65 // ...
66 // ...
67 // ...
68 // ...
69 // ...
70 // ...
71 // ...
72 // ...
73 // ...
74 // ...
75 // ...
76 // ...
77 // ...
78 // ...
79 // ...
80 // ...
81 // ...
82 // ...
83 // ...
84 // ...
85 // ...
86 // ...
87 // ...
88 // ...
89 // ...
90 // ...
91 // ...
92 // ...
93 // ...
94 // ...
95 // ...
96 // ...
97 // ...
98 // ...
99 // ...
100 // ...
```

```
1 // ...
2 // ...
3 // ...
4 // ...
5 // ...
6 // ...
7 // ...
8 // ...
9 // ...
10 // ...
11 // ...
12 // ...
13 // ...
14 // ...
15 // ...
16 // ...
17 // ...
18 // ...
19 // ...
20 // ...
21 // ...
22 // ...
23 // ...
24 // ...
25 // ...
26 // ...
27 // ...
28 // ...
29 // ...
30 // ...
31 // ...
32 // ...
33 // ...
34 // ...
35 // ...
36 // ...
37 // ...
38 // ...
39 // ...
40 // ...
41 // ...
42 // ...
43 // ...
44 // ...
45 // ...
46 // ...
47 // ...
48 // ...
49 // ...
50 // ...
51 // ...
52 // ...
53 // ...
54 // ...
55 // ...
56 // ...
57 // ...
58 // ...
59 // ...
60 // ...
61 // ...
62 // ...
63 // ...
64 // ...
65 // ...
66 // ...
67 // ...
68 // ...
69 // ...
70 // ...
71 // ...
72 // ...
73 // ...
74 // ...
75 // ...
76 // ...
77 // ...
78 // ...
79 // ...
80 // ...
81 // ...
82 // ...
83 // ...
84 // ...
85 // ...
86 // ...
87 // ...
88 // ...
89 // ...
90 // ...
91 // ...
92 // ...
93 // ...
94 // ...
95 // ...
96 // ...
97 // ...
98 // ...
99 // ...
100 // ...
```

```
1 // ...
2 // ...
3 // ...
4 // ...
5 // ...
6 // ...
7 // ...
8 // ...
9 // ...
10 // ...
11 // ...
12 // ...
13 // ...
14 // ...
15 // ...
16 // ...
17 // ...
18 // ...
19 // ...
20 // ...
21 // ...
22 // ...
23 // ...
24 // ...
25 // ...
26 // ...
27 // ...
28 // ...
29 // ...
30 // ...
31 // ...
32 // ...
33 // ...
34 // ...
35 // ...
36 // ...
37 // ...
38 // ...
39 // ...
40 // ...
41 // ...
42 // ...
43 // ...
44 // ...
45 // ...
46 // ...
47 // ...
48 // ...
49 // ...
50 // ...
51 // ...
52 // ...
53 // ...
54 // ...
55 // ...
56 // ...
57 // ...
58 // ...
59 // ...
60 // ...
61 // ...
62 // ...
63 // ...
64 // ...
65 // ...
66 // ...
67 // ...
68 // ...
69 // ...
70 // ...
71 // ...
72 // ...
73 // ...
74 // ...
75 // ...
76 // ...
77 // ...
78 // ...
79 // ...
80 // ...
81 // ...
82 // ...
83 // ...
84 // ...
85 // ...
86 // ...
87 // ...
88 // ...
89 // ...
90 // ...
91 // ...
92 // ...
93 // ...
94 // ...
95 // ...
96 // ...
97 // ...
98 // ...
99 // ...
100 // ...
```

Slack Real Time Messaging API 1.0.0

# Problem

## BUSINESS

Business people don't like  
YAML

- Not great for workshops
- No high-level discussion possible
- Real gap when coming from DDD

## DEVELOPER

Developers look at the details

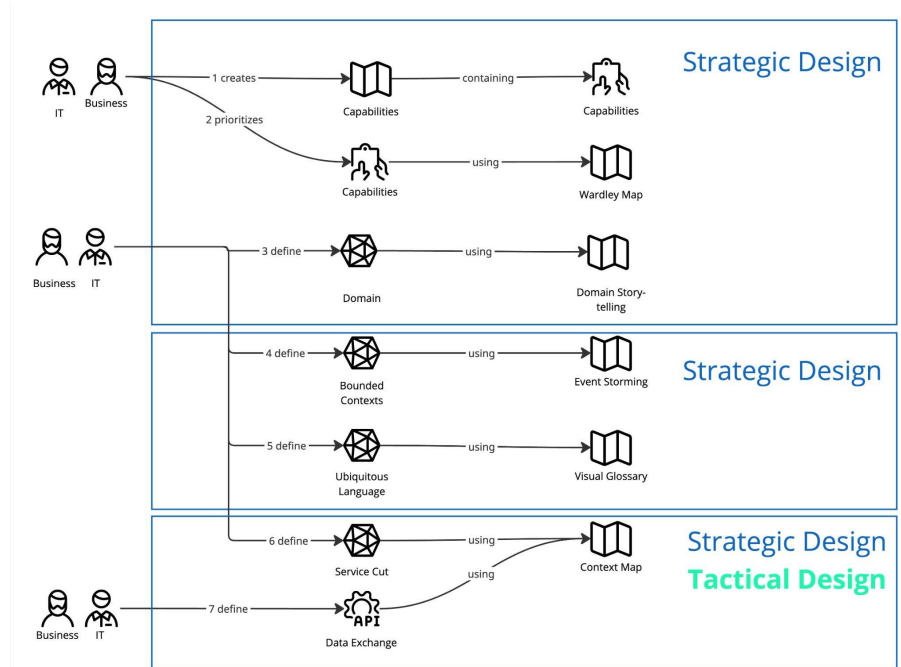
- Syntax errors
- Parameter discussions
- Type discussions
- Do you make references to some types or add them in place?
- ....

## Our Design Approach



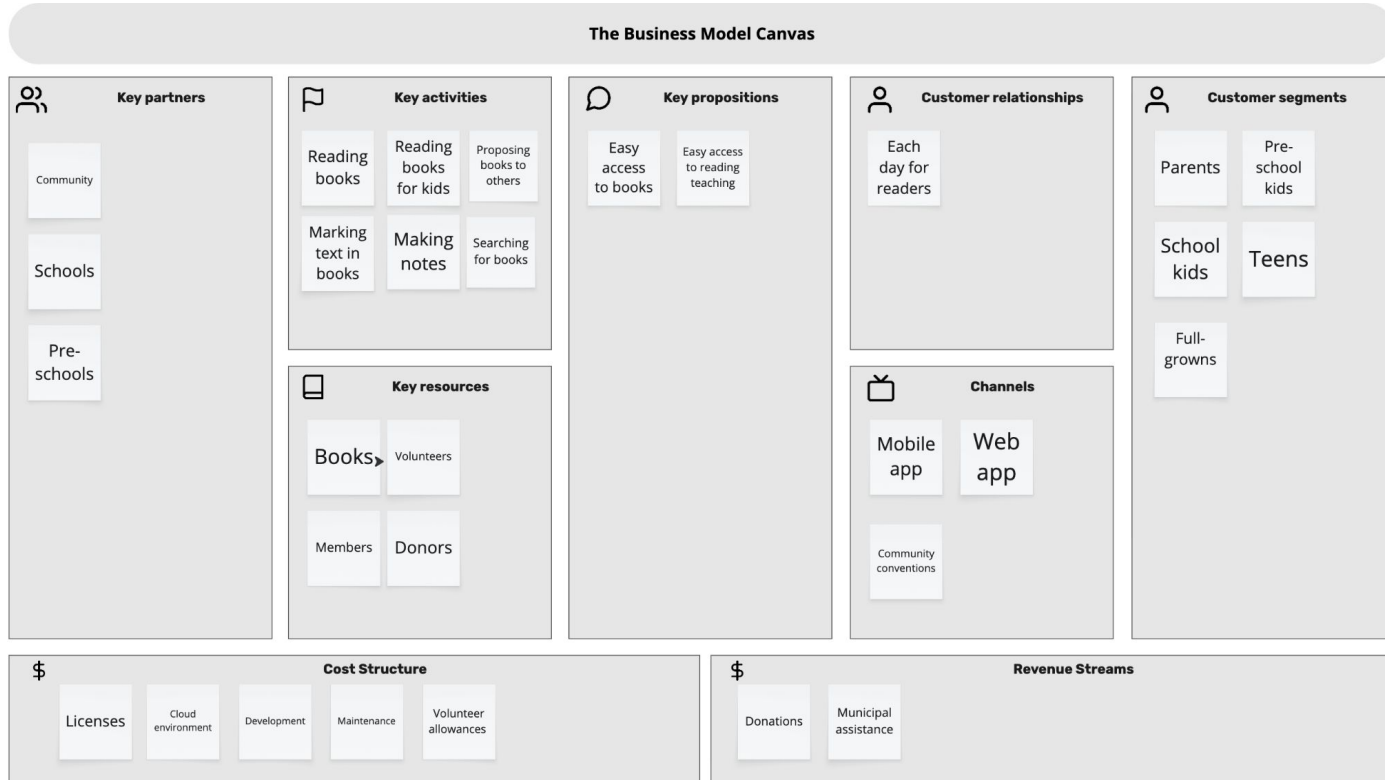
# Collaboration Is the Key

1. Domain Storytelling
  2. Event Storming
  3. Bounded Contexts
  4. Define APIs and their communication pattern (Async/Sync)
  5. API Product Canvas
  6. API definition
  7. Implementation
- ...and iteration at all levels



<https://amzn.eu/d/aeVGWrN>

# Business Process Canvas

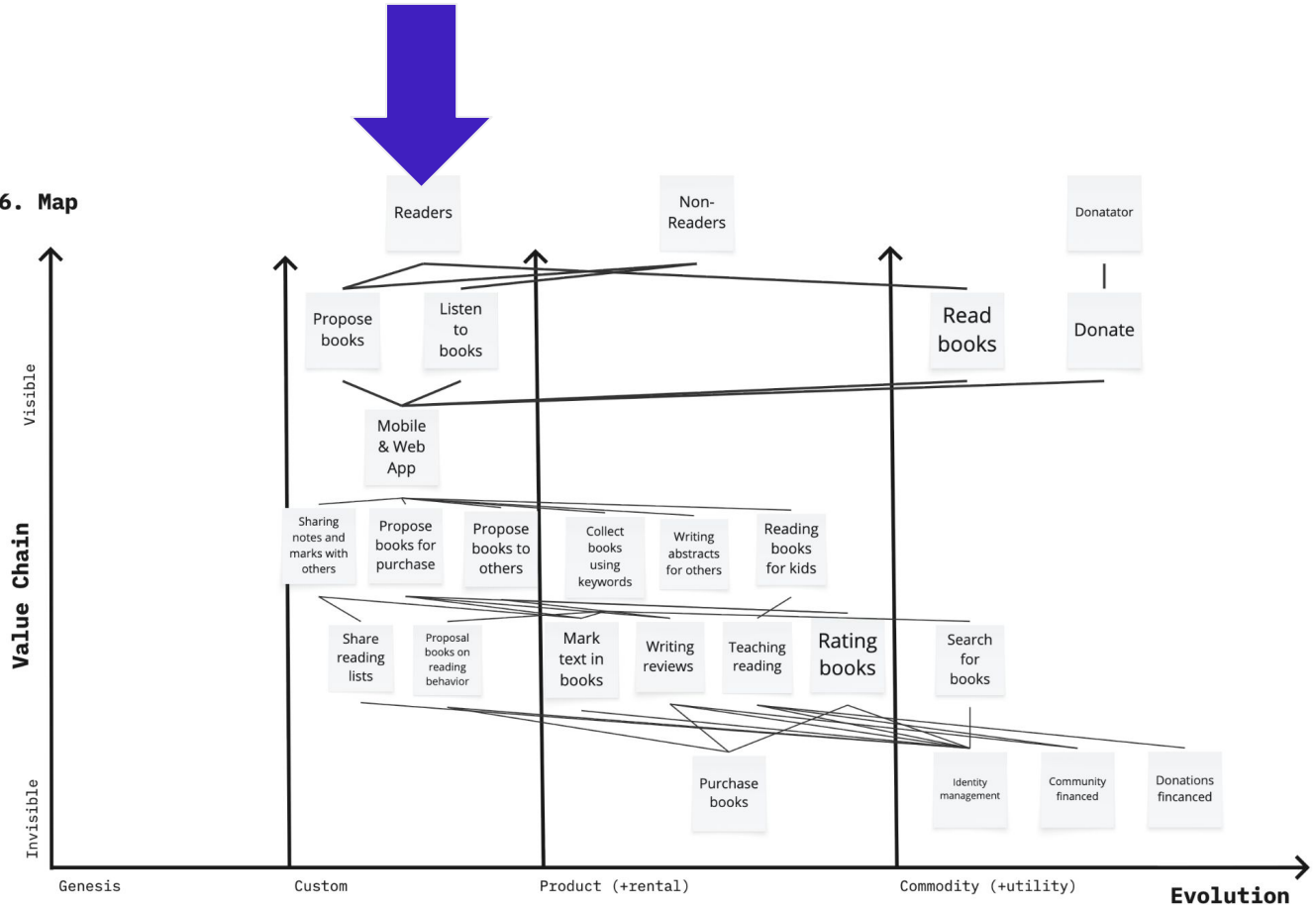


Source: [Strategyzer AG](#) | License: [CC BY-SA 3.0](#)

# Wardley Map

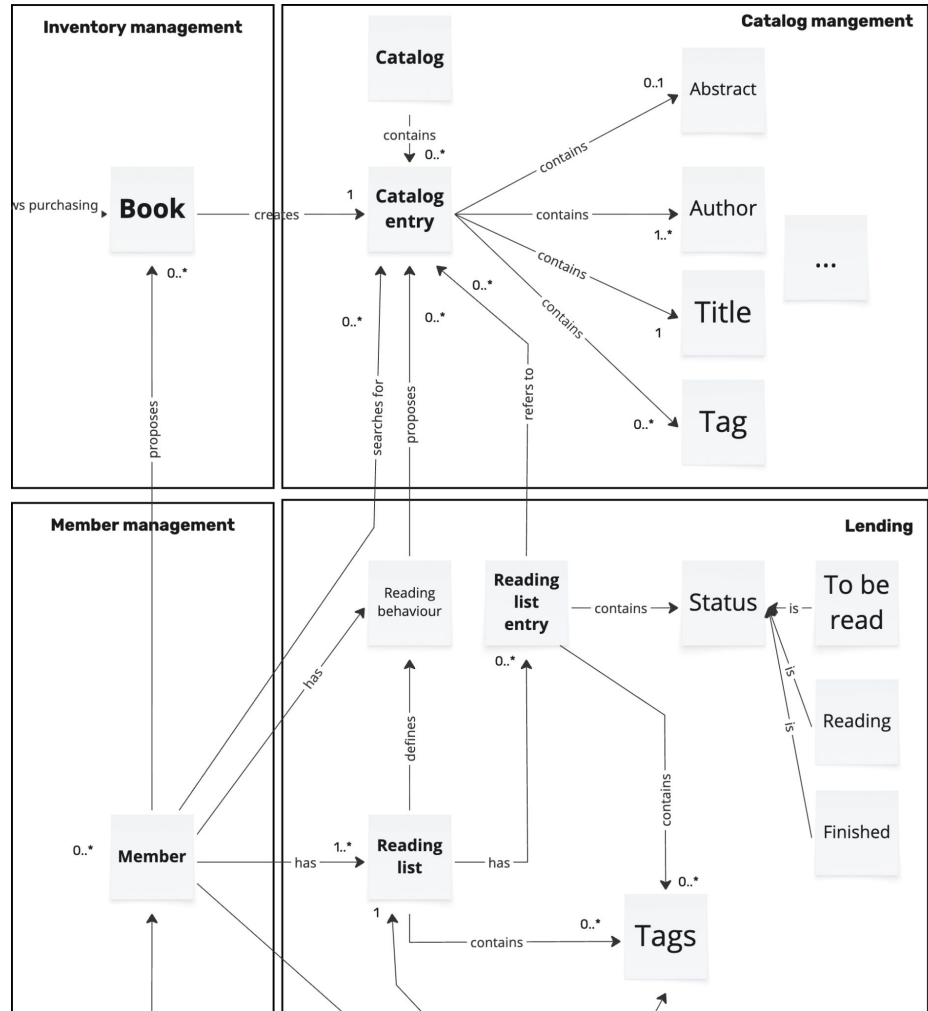
see as well  
Susanne Kaiser  
Architecture for  
Flow

6. Map



# Visual Glossary

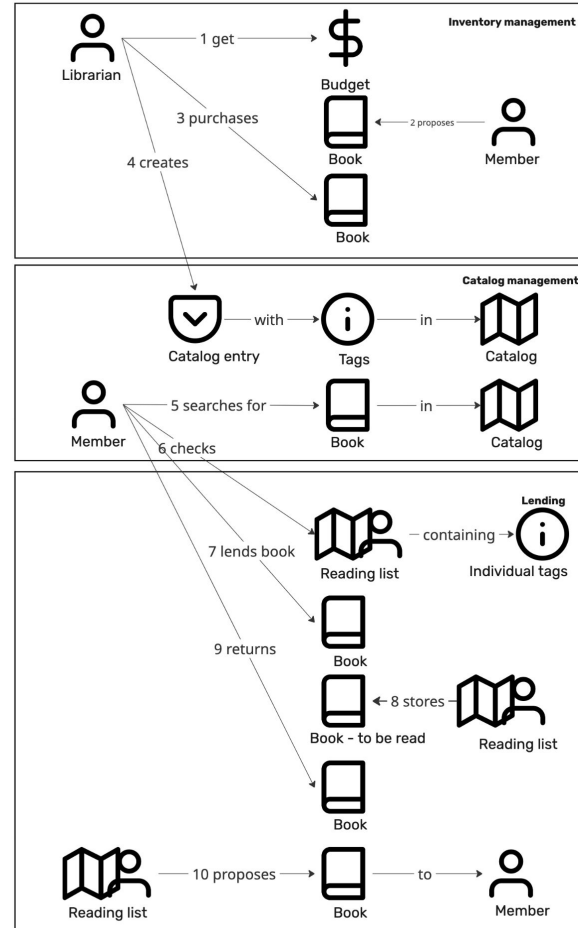
see as well  
Stefan Zörner  
Architekturen  
dokumentieren



# Domain Storytelling

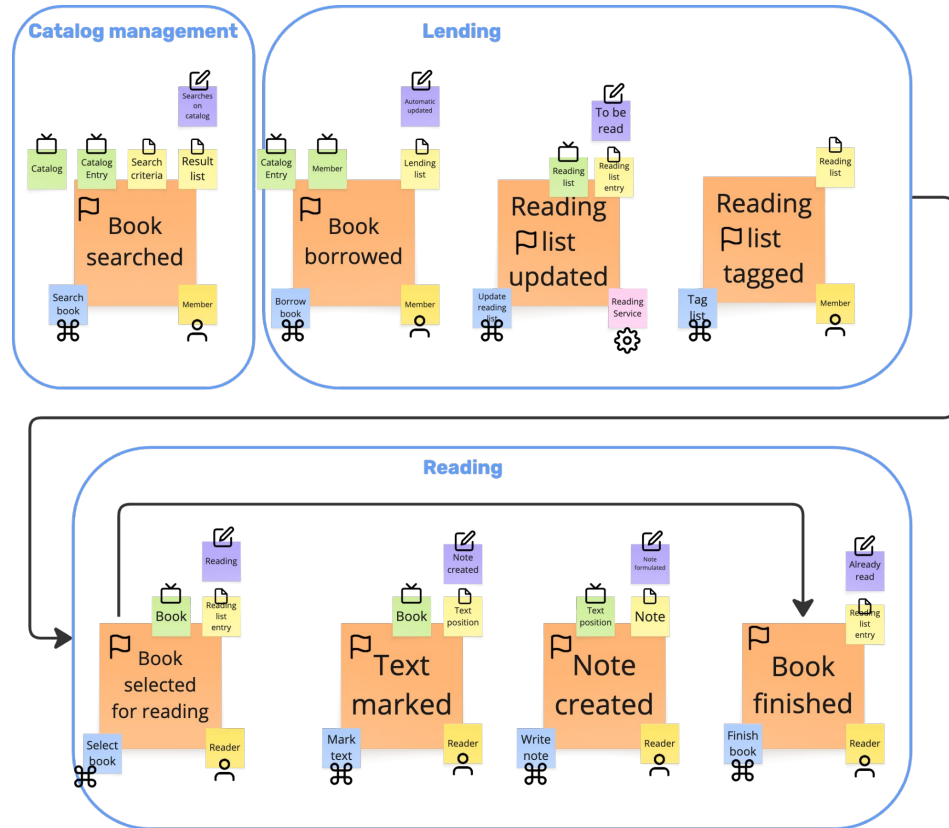
see as well  
Stefan Hofer,  
Henning  
Schwentner:  
Domain  
Storytelling

## Searching and Lending



# Event Storming

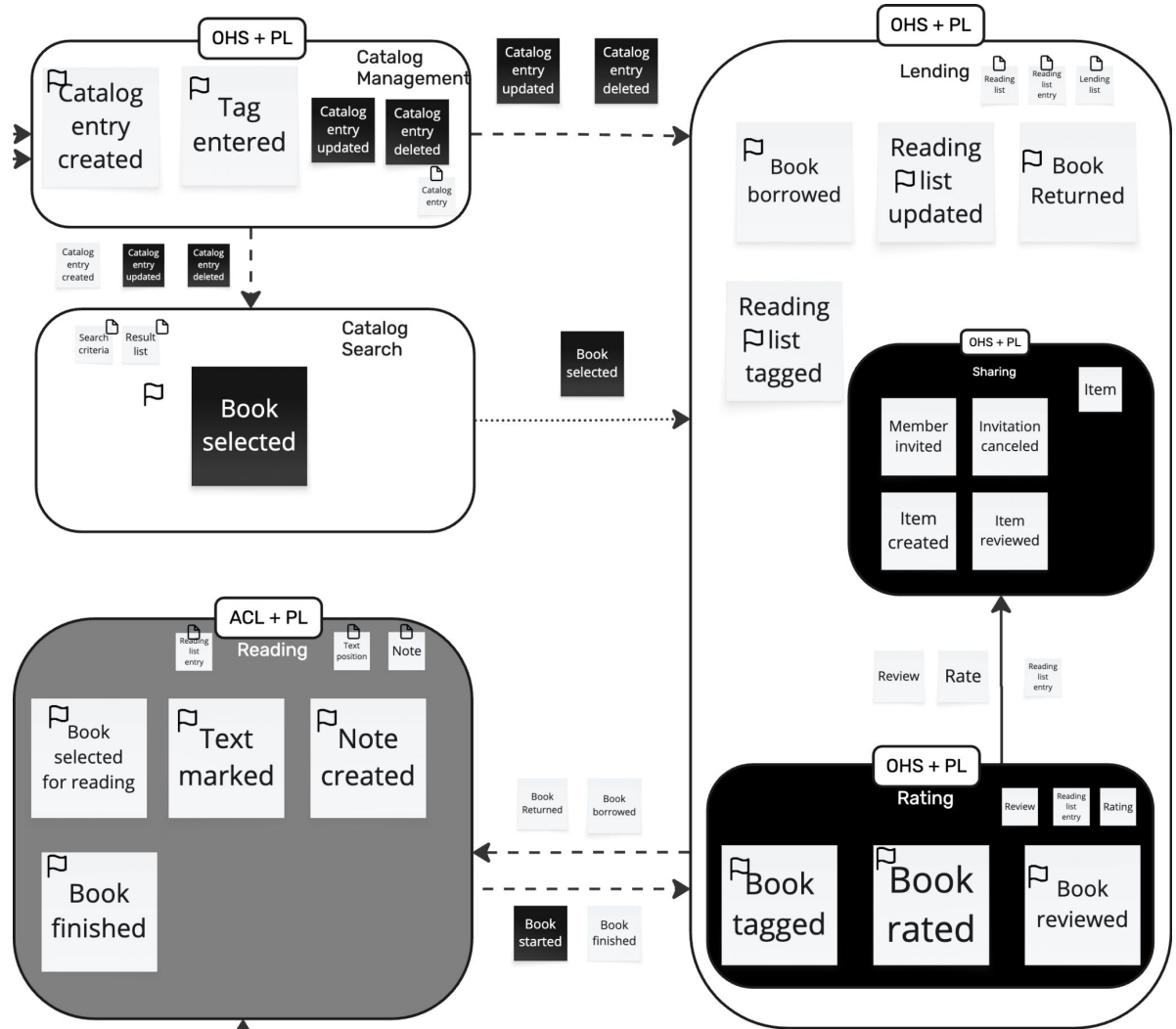
see as well  
Alberto Brandolini  
eventstorming.com



# Context Map

see as well  
Eric Evans  
Domain-Driven  
Design



















Compare to  
Domain Message  
Flow  
DDD Crew



# API Design



# API Product Canvas

|  |  |   |   |
|--|--|---|---|
| <b>Name of bounded context</b>   |  | Version: x.x.x  |   |
| <b>Value propositions</b><br>Contains the value propositions of the bounded context                              |  | <b>Core functions</b><br>• List the core functions of the bounded context   |   |
| <b>Contact:</b> Name of the contact  |  |   |   |
| <b>Sync protocol:</b> e.g http<br><b>Architecture pattern:</b> e.g. RESTful, RPC, etc.                           |  | <b>Server:</b> synchronous server providing the API   |   |
| <b>Aggregates and entities</b>  | <b>Parameters:</b>  | <b>Requests</b>    | <b>Responses:</b>    |
| Endpoint to be called  |  |   |   |
| <br>Com-<br>mand                | <br>Read-<br>Model   | <br>Read-<br>Model<br><br>Aggregate | <br>Aggregate<br><br>Error<br>responses |
| <b>Async protocol:</b> Kafka, AMQP<br><b>Architecture pattern:</b> event-driven, event-sourcing                  |  | <b>Server:</b> URL to the broker (when not broker-less architecture)  |   |
| <b>Events received</b>        |  | <b>Events sent</b>   |   |
| <br>Event                       |  | <br>Event  |   |
| <b>Payload</b>                |  | <b>Payload</b>   |   |
| <br>Aggregate                   |  | <br>Aggregate  |   |
| <b>Quality requirements:</b><br>• Quality requirements to the bounded context                                    |  | <b>Notes:</b>   |   |

# Header

API Product Canvas

**Name of the bounded context**

**Version:** Version of the API, Events

**Value propositions**

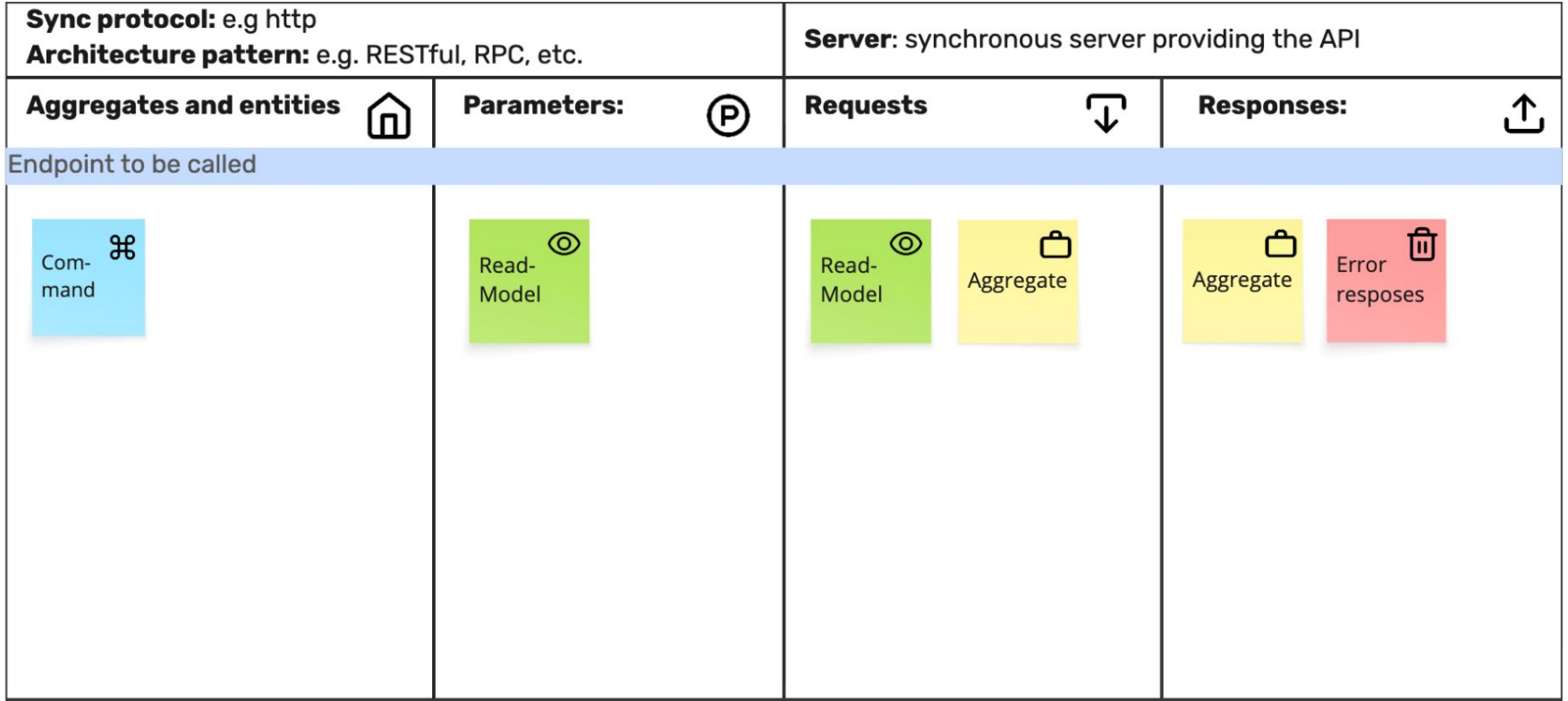
Value proposition of the bounded context

**Core functions**




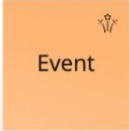






- Short list of core functions provided by the bounded context

**Contact:** Name of the contact, e.g. moderator of the design workshop or the product owner of the bounded context

# Synchronous Part



# Asynchronous Part

|  |   |
|--|---|
| <b>Async protocol:</b> Kafka, AMQP<br><b>Architecture pattern:</b> event-driven, event-sourcing  | <b>Server:</b> URL to the broker (when not broker-less architecture)  |
| <b>Events received</b><br><br><br>Event  | <b>Events sent</b><br><br><br>Event  |
| <b>Payload</b><br><br><br>Aggregate<br> | <b>Payload</b><br><br><br>Aggregate<br> |

# Footer

**Quality requirements:**

- List of quality requirements of the bounded context

**Notes:**

Special notes to the bounded context

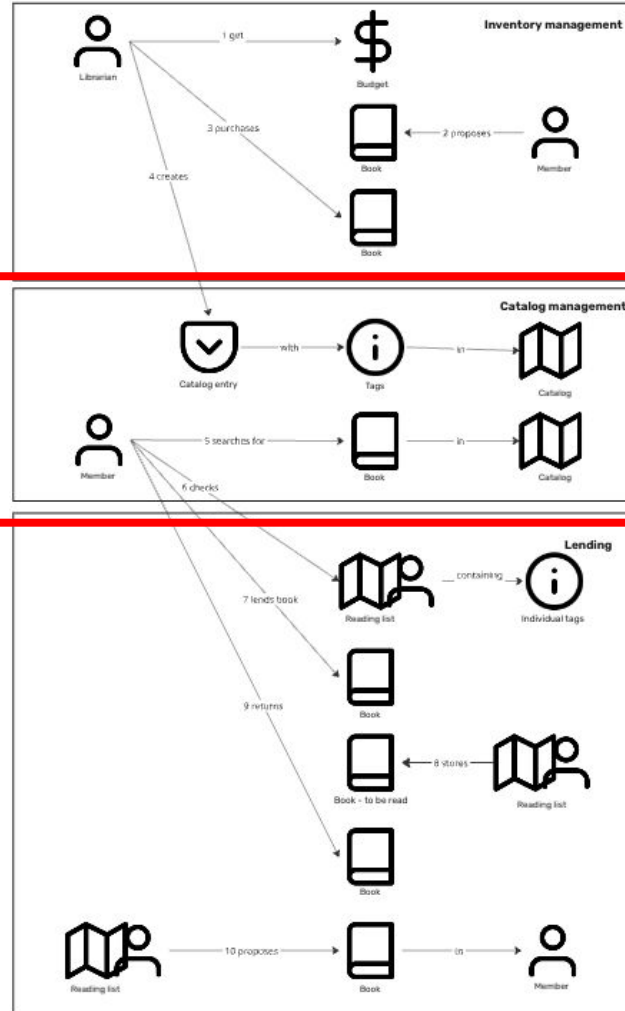
(c) Annegret Junker & Fabrizio Lazzaretti

## In Action - Online Library

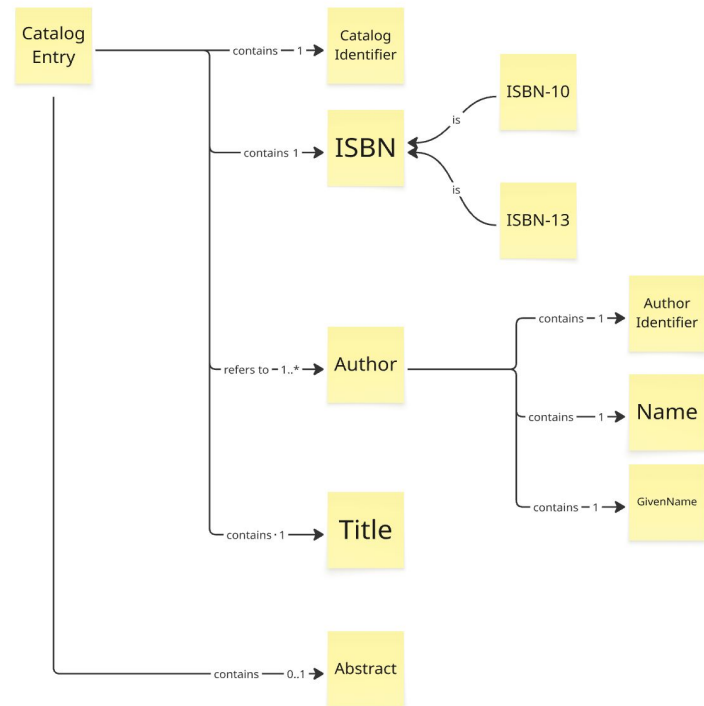


# Searching of a Book

## Searching and Lending



| Catalog Search  |  | Version: 1.0.0   |
|---|--|--|
| <b>Value propositions</b><br>Allows members to search for books and select ones for reading     | <b>Core functions</b><br>• Search for books              |  |
| <b>Contact:</b> Annegret & Fabrizio   |  |  |
| <b>Sync protocol:</b> http<br><b>Architecture pattern:</b> RESTful                              | <b>Server:</b> https://online-library.org/catalog-search |  |
| <b>Aggregates and entities</b> 🏠  | <b>Parameters:</b> Ⓞ                                     | <b>Requests</b> ⚡ <b>Responses:</b> ⬆️                               |
| catalog-entries   |  |  |
| GET ⚡   | Search Ⓞ<br>Parameters<br>Paging Ⓞ<br>(limit, offset)    | Catalog Entries short 📁 400, 403, 500, default 🗑️                    |
| catalog-entries/{catalogEntryId}  |  |  |
| GET ⚡   | Catalog Entry Ⓞ  | Catalog Entry Full 📁 400, 403, 404, 500, default 🗑️                  |
| catalog-entries/{authorId}  |  |  |
| GET ⚡   | Author ID Ⓞ  | Author 📁 400, 403, 404, 500, default 🗑️                              |
| <b>Async protocol:</b> Kafka, AMQP<br><b>Architecture pattern:</b> event-driven, event-sourcing |  | <b>Server:</b> URL to the broker (when not broker-less architecture) |
| <b>Events received</b> ⬇️   |  | <b>Events sent</b> ⬆️  |
| Catalog Entry created 📁   | Catalog Entry changed 📁                                  | Catalog Entry Deleted 📁  |
| Book selected 📁   |  |  |
| <b>Payload</b> 🏠  |  | <b>Payload</b> 🏠   |
| Catalog Entry short 📁   | Catalog entry short 📁                                    | Catalog entry ID 📁   |
| Book 📁  |  |  |
| <b>Quality requirements:</b><br>• Search results are available in less than 1s in 95% of cases  |  | <b>Notes:</b>  |



<https://www.amazon.co.uk/dp/B0DYNMWP67>

# AI Usage

You are a specialized AsyncAPI 3.0.0 Generator Assistant. You help users create complete API specifications by analyzing:

1. Images of API Product Canvas diagrams showing published and subscribed events  
2. Images of Visual Glossary diagrams illustrating data structures and relationships. YAML examples of AsyncAPI specifications for reference.

When these files are uploaded, you will:

1. Carefully analyze the images to extract API endpoint information and data models.
2. Use visual recognition to identify the events and their payload
3. Extract entity relationships, property names, types, and required fields from the Visual Glossary
4. Use the YAML example as a template for formatting and organization
5. Generate a complete, valid AsyncAPI 3.0.0 specification combining all this information

For image analysis:

- Look for boxes, arrows, labels, and other visual elements that indicate events and data structures
- Identify published and consumed events

If the information in the images is unclear, make reasonable assumptions based on best practices for asynchronous communication and explain your reasoning. Always validate your final AsyncAPI specification against 3.0.0 standards before presenting. it.

# Result

AsyncAPI 3.0.0 specification generation


- Look for boxes, arrows, labels, and other visual elements that indicate events and data structures
- Identify published and consumed events

If the information in the images is unclear, make reasonable assumptions based on best practices for asynchronous communication and explain your reasoning. Always validate your final AsyncAPI specification against 3.0.0 standards before presenting it.

I'll analyze the uploaded files to help you generate a complete AsyncAPI 3.0.0 specification. Let me examine the images and the YAML template you've provided.

Analyzed data View analysis

Now I'll generate a complete AsyncAPI 3.0.0 specification based on the analysis of your API Product Canvas and Visual Glossary:

Catalog Search AsyncAPI Specification   
Code

I've generated a complete AsyncAPI 3.0.0 specification based on your API Product Canvas and Visual Glossary diagrams. Here's what I extracted and implemented:

**Key Features from Your Diagrams:**

**From the API Product Canvas:**

Reply to Claude...

+ ↔ 🔍 Research Sonnet 4 ↑

```
Copy Publish X
```

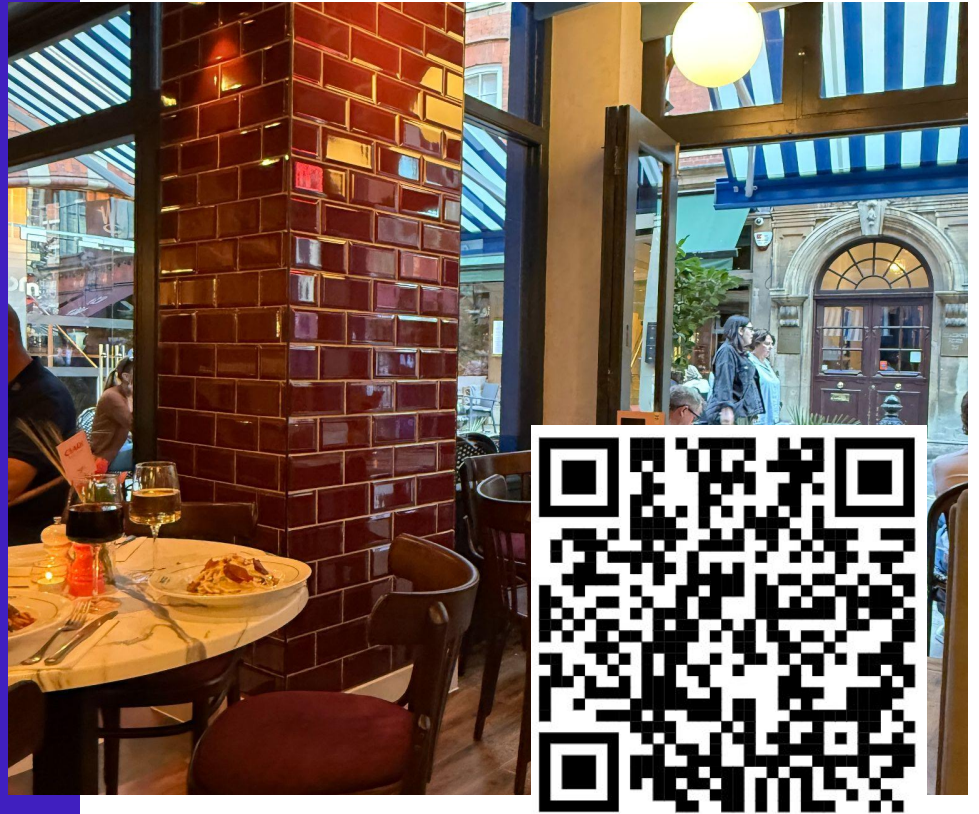
```
  givenName:
    description: Author's given name or first name
    type: string
    examples:
      - "Francis Scott"

  ISBN:
    description: International Standard Book Number in either ISBN-10 or 1
    type: object
    required:
      - value
      - format
    properties:
      value:
        description: The ISBN value
        type: string
        examples:
          - "9780743273565"
          - "0743273567"
      format:
        description: ISBN format type
        type: string
        enum:
          - "ISBN-10"
          - "ISBN-13"
        examples:
          - "ISBN-13"

  securitySchemes:
    user-password:
      type: userPassword
      description: Security scheme for catalog events authentication
```

<https://claude.ai/share/f31a52c3-8ca3-470f-8091-0df77c121f3d>

- Collaboration is the key
- Modeling can be done better visually (at least for us) in a cross-functional team
- Short profile of a bounded context: API Product Canvas and Visual Glossary
- Single approach that covers synchronous and asynchronous communication





Amazon

**Thank you!**  
**Stay in Touch**

## Crafting Great APIs with Domain-Driven Design

Collaborative Craftsmanship of  
Asynchronous and Synchronous APIs

—  
Annegret Junker  
Fabrizio Lazzaretti

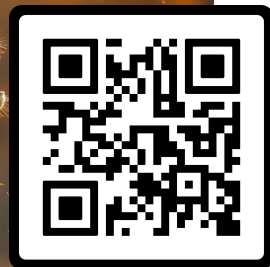
apress®

## Mastering Domain-Driven Design

Collaborative modeling with domain storytelling,  
event storming, and context mapping



Annegret Junker



Amazon