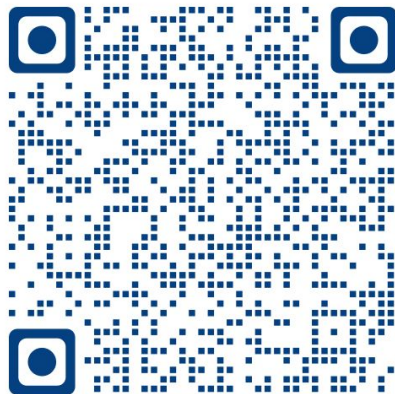


The API Product Canvas

How to Design AsyncAPIs in Workshops: The API Product Canvas

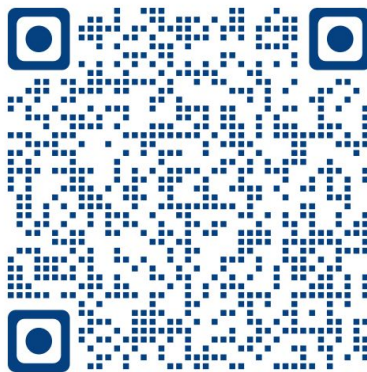
Dr. Annegret Junker &
Fabrizio Lazzaretti





Architect at codecentric

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



Architect at Wavestone

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

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

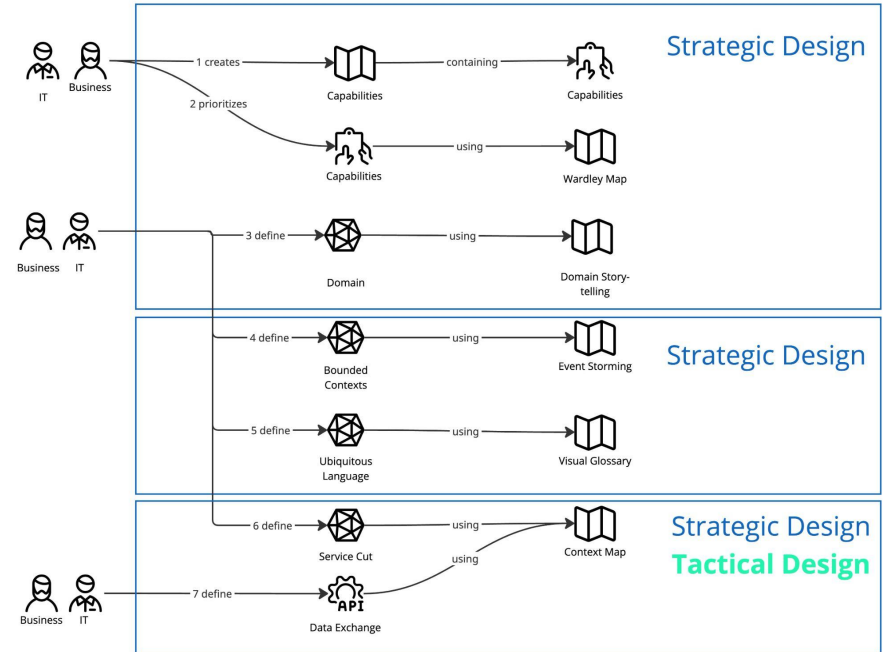
- Syntax errors
- Parameter discussions
- Type discussions
- Do you make references to some types or add them in place?
-

Our design approach

The best APIs come from
collaboration between business
and IT

Design approach - Synergetic Blueprint

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>

AI Maturity Starts with Asking the Right Questions

- We don't yet have a standard for AI-ready API metadata
- But teams can start assessing APIs today
 - Is the API's purpose clear?
 - Are typical usage flows documented?
 - Is behavior like retries or side effects explicit?
 - Can others understand and use it without extra help?
- Even informal answers help shape future strategy
- Be ready to re-assess and refactor



Erik Wilde June, 2025 apidays
Munich



The API Product Canvas

Our solution: the API Product Canvas

- One high level view on all interactions on a bounded context

<https://miro.com/miroverse/api-product-canvas/>

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

(c) Annegret Junker & Fabrizio Lazzaretti

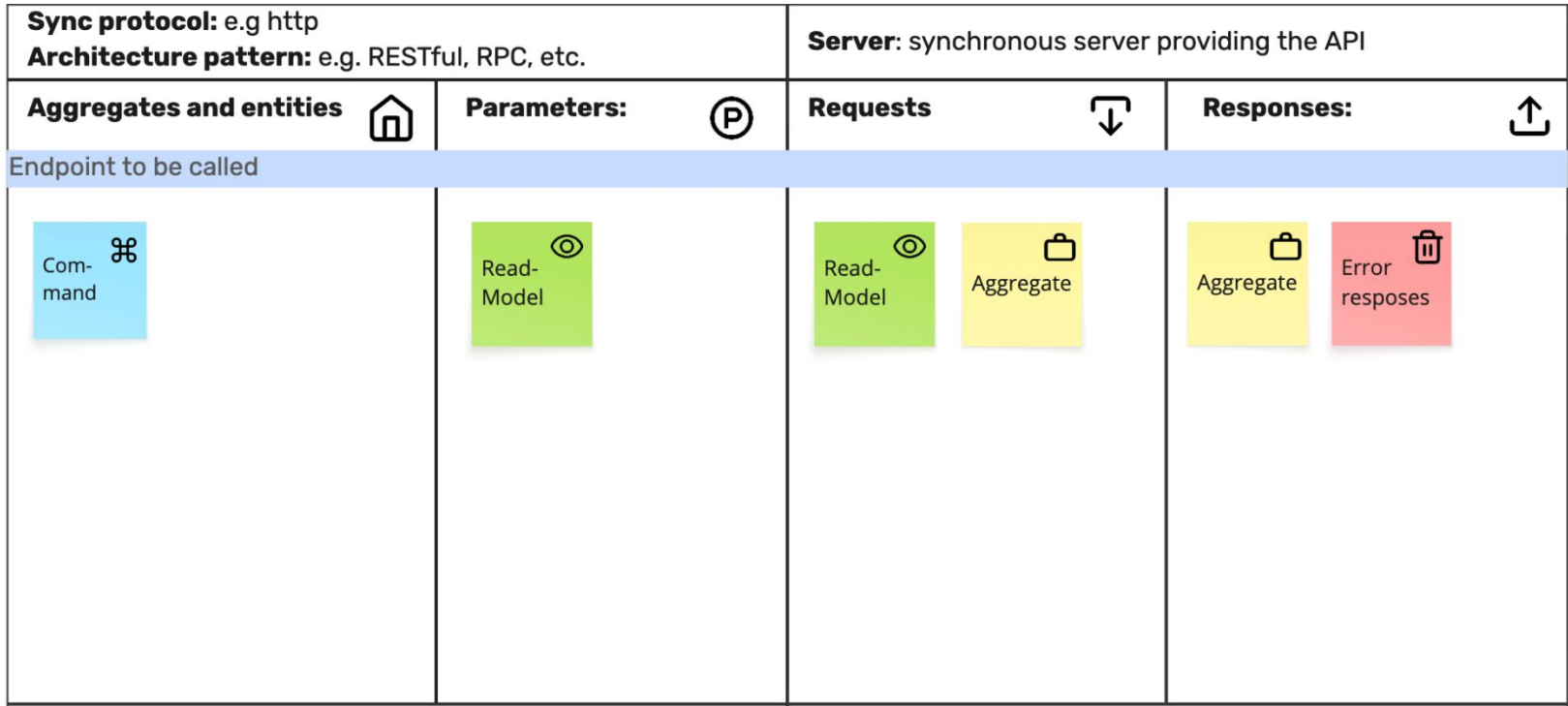
The Header - Bounded Context

API Product Canvas

Name of the bounded context		Version: Version of the API, Events
Value propositions Value proposition of the bounded context	Core functions <ul style="list-style-type: none">• 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		





<https://miro.com/miroverse/api-product-canvas/>

The synchronous part



<https://miro.com/miroverse/api-product-canvas/>

The asynchronous part

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

<https://miro.com/miroverse/api-product-canvas/>

Quality requirements:

- List of quality requirements of the bounded context

Notes:

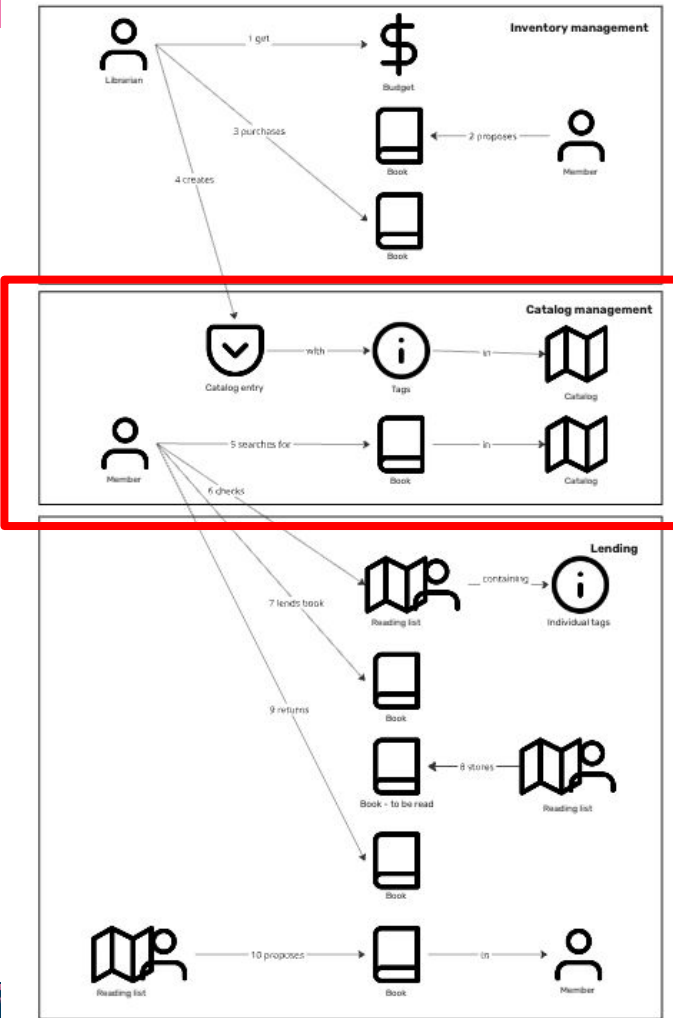
Special notes to the bounded context

(c) Annegret Junker & Fabrizio Lazzaretti

API Product Canvas in Action: Online Library

Example Online Library

Searching and Lending



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

Catalog Search

Version: 1.0.0

Value propositions
Allows members to search for books and select ones for reading

Core functions
• Search for books

Contact: Annegret & Fabrizio

Sync protocol: http
Architecture pattern: RESTful

Server: https://online-library.org/catalog-search

Aggregates and entities 🏠 **Parameters:** Ⓞ **Requests** ⬇️ **Responses:** ⬆️

catalog-entries

GET 🌀	Search parameters Ⓞ Paging (limit, offset) Ⓞ	Catalog Entries short 📁 400, 403, 500, default 🗑️ Catalog Entries full 📁 400, 403, 404, 500, default 🗑️
-------	---	--

catalog-entries/{catalogEntryId}

GET 🌀	Catalog Entry Ⓞ	Catalog Entry Full 📁 400, 403, 404, 500, default 🗑️ Author 📁 400, 403, 404, 500, default 🗑️
-------	-----------------	--

catalog-entries/{authorId}

GET 🌀	Author ID Ⓞ	Author 📁 400, 403, 404, 500, default 🗑️ Title 📁 400, 403, 404, 500, default 🗑️
-------	-------------	---

Async protocol: Kafka, AMQP
Architecture pattern: event-driven, event-sourcing

Server: URL to the broker (when not broker-less architecture)

Events received ⬇️

Catalog Entry created 📁
Catalog Entry changed 📁
Catalog Entry Deleted 📁

Events sent ⬆️

Book selected 📁

Payload 🏠

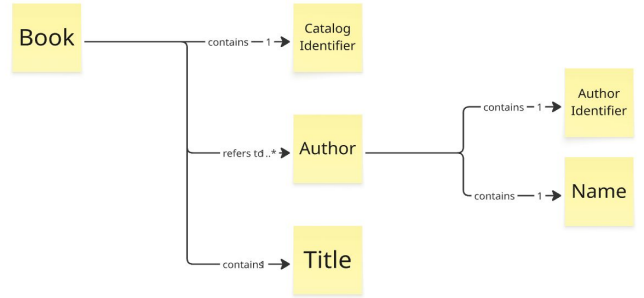
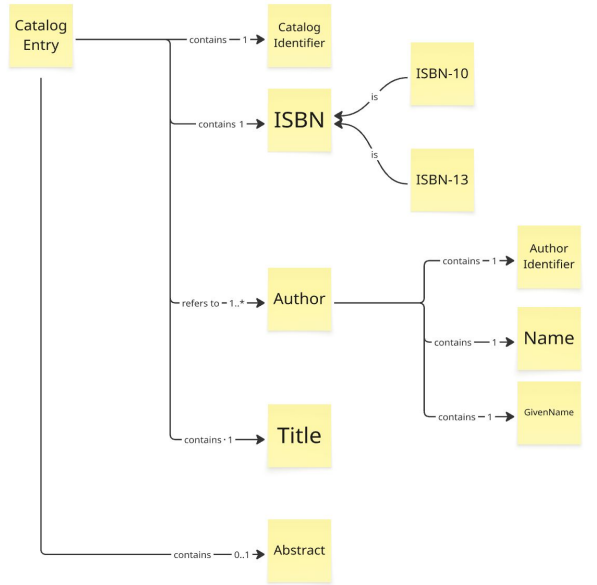
Catalog Entry short 📁
Catalog entry short 📁
Catalog entry ID 📁

Payload 🏠

Book 📁

Quality requirements:
• Search results are available in less than 1s in 95% of cases

Notes:



<https://www.amazon.co.uk/dp/BODYNMWP67>

What happens

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.

What happens

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

Thank you for listening

Contact Annegret

<https://github.com/Grinsetddy>

<https://www.linkedin.com>

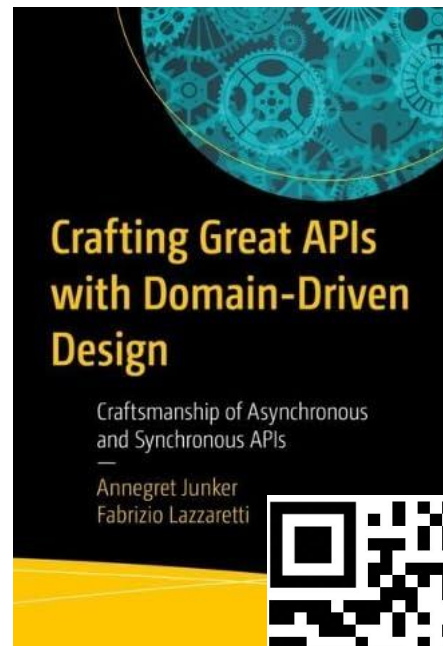
[/in/dr-annegret-junker-141a99a4/](https://www.linkedin.com/in/dr-annegret-junker-141a99a4/)

Contact Fabrizio

<https://github.com/Lazzaretti>

<https://www.linkedin.com>

[/in/fabrizio-lazzaretti/](https://www.linkedin.com/in/fabrizio-lazzaretti/)



THANK YOU!