AI-Driven Architecture

Structurizr × OpenCode

Mansun Kuo
2026-10-14

CC BY NC SA

Outline

- Environment Setup
- All About an Architect
- C4 Model & Structurizr
- Architectural Decision Records(ADR)
- Key Takeaways

Environment Setup

https://tinyurl.com/32p6bnaa

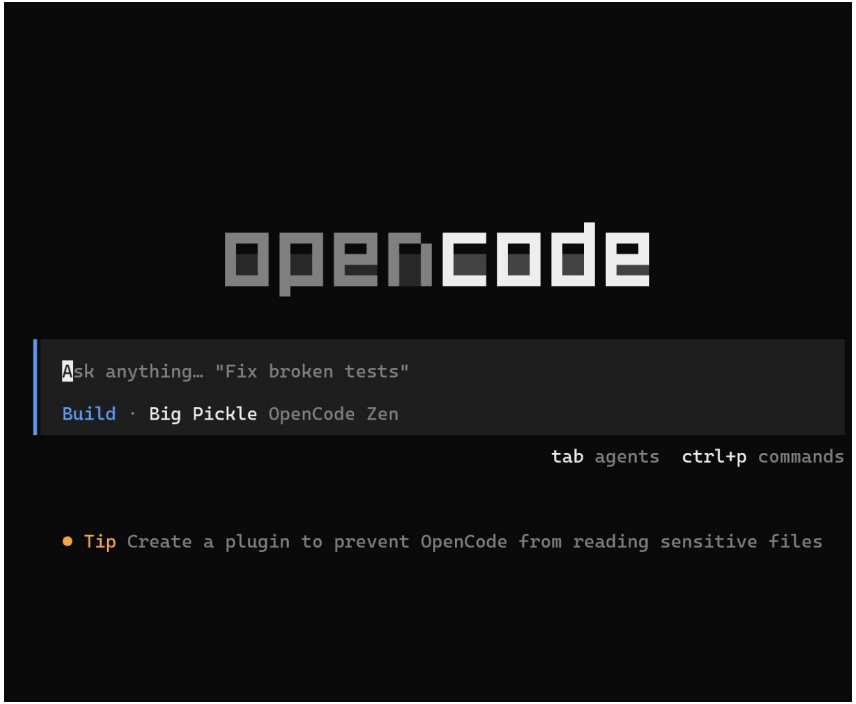


OpenCode

The open source AI coding agent with any (free) LLM provider. Running **opencode** in terminal starts the TUI for the current directory.

- Installation
- OpenCode Console
  - /connect
  - /models
  - https://opencode.ai/console/

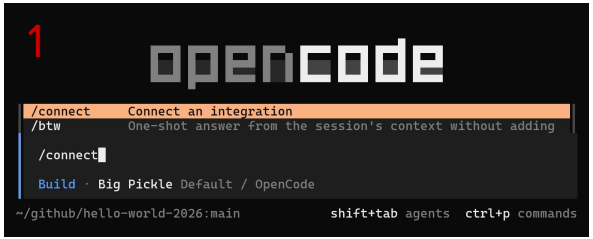
MODEL	INPUT	OUTPUT	CACHED READ
Big Pickle	Free	Free	Free
MiMo-V2.5 Free	Free	Free	Free
Ling 3.0 Flash Fin Free	Free	Free	Free
Nemotron 3 Ultra Free	Free	Free	Free
Nemotron 3.5 Lightning Free	Free	Free	Free
Muse Spark 1.3 Contributor Free	Free	Free	Free



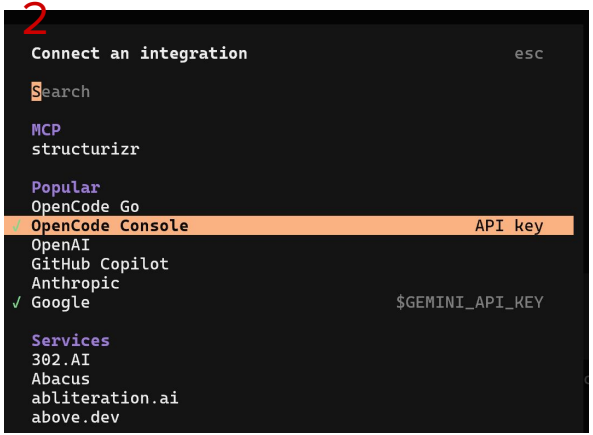
https://opencode.ai/v2/docs

How to Connect

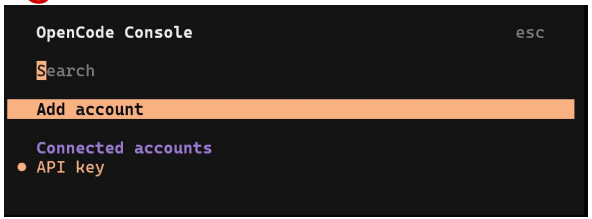
1



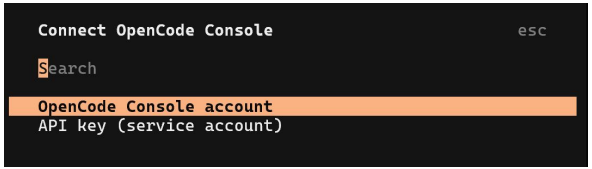
2



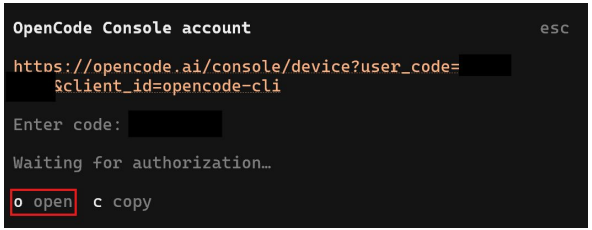
3



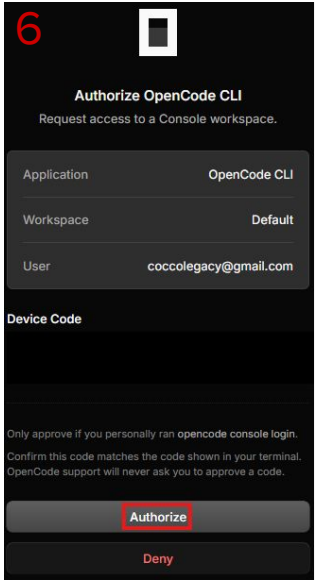
4



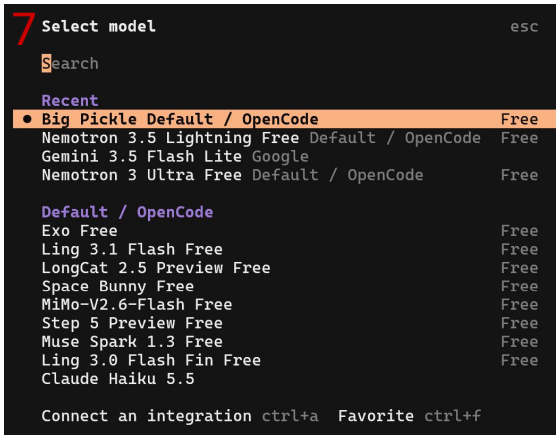
5



6



7



https://github.com/mansunkuo/hello-world-2026/blob/main/docs/setup.md#2-%E9%80%A3%E6%8E%A5%E6%A8%A1%E5%9E%8B
https://github.com/mansunkuo/hello-world-2026/blob/main/docs/setup-en.md#2-connect-a-model

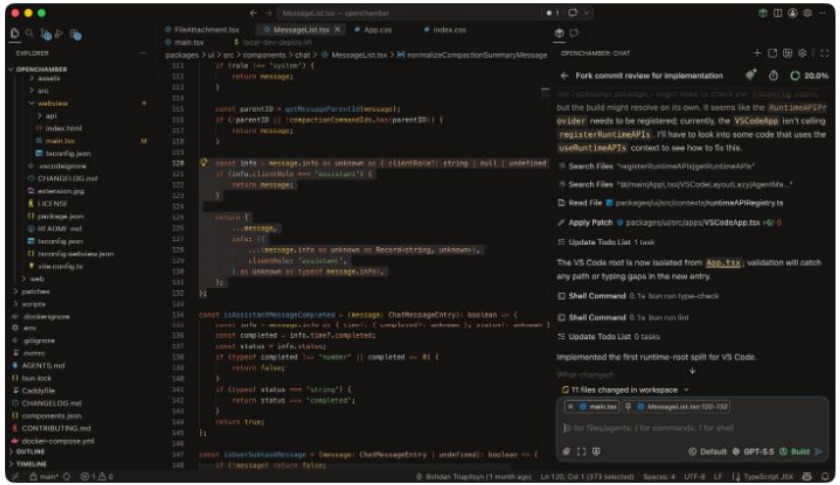
Other than TUI

Install OpenChamber and Structurizr VS Code extension. Use **`opencode pair`** in V2 (or opencode web in V1) to launce web UI.

OpenChamber VS Code Extension

stars 9.9k | release v1.23.2 | Discord | Support Project

OpenCode AI coding agent, right inside your editor. No tab-switching, no context loss.



Like the extension? There's also a desktop app and web version with even more features.

https://marketplace.visualstudio.com/items?itemName=FedaykinDev.openchamber
https://marketplace.visualstudio.com/items?itemName=ciarant.vscode-structurizr
https://opencode.ai/v2/docs#web

```
> opencode pair

Open a link to connect. Links work once and expire in 5 minutes.

http://127.0.0.1:49374/auth/connect/JzBGN6oxgZVvpad_Z1CF6A

Over SSH? Forward the port, then open the link on your machine:
ssh -L 49374:127.0.0.1:49374 <host>
If port 49374 is busy locally, forward another port and use it in the link.
```

Get Your Own Template

```
git clone --branch init git@github.com:mansunkuo/hello-world-2026.git

cd hello-world-2026

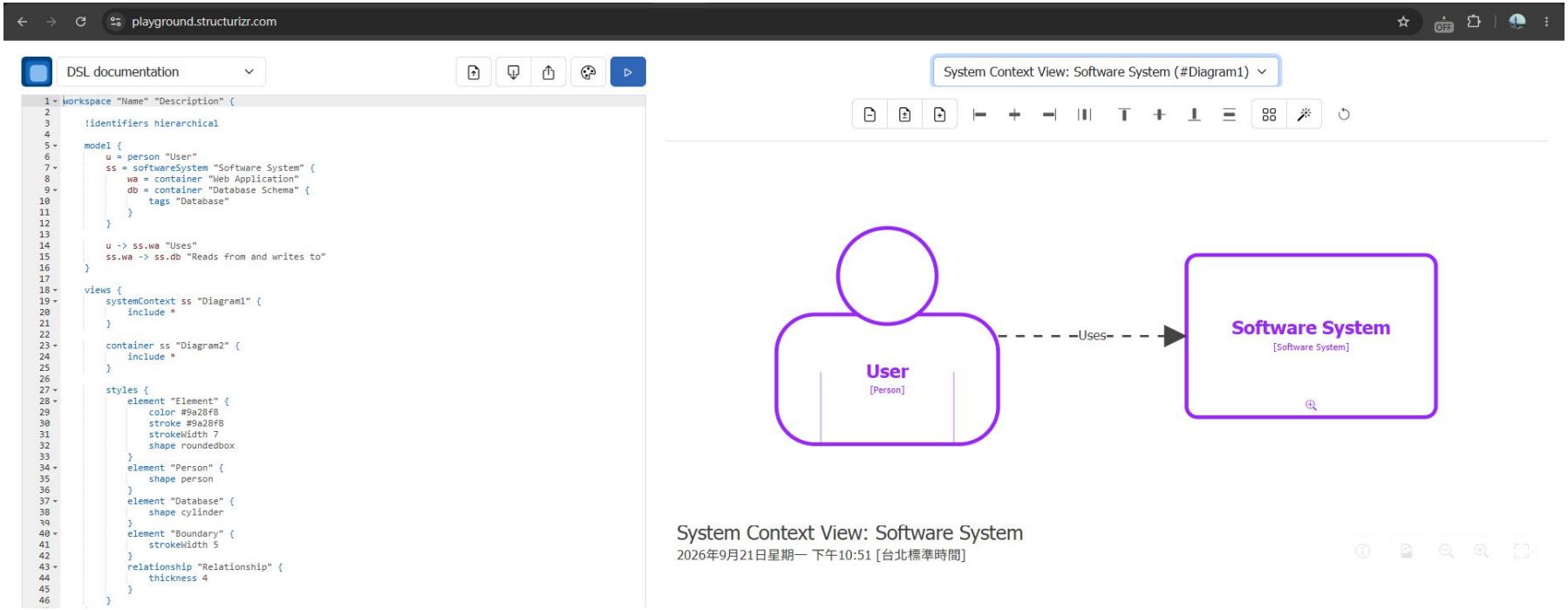
git checkout -b main

git branch -M main
```

https://github.com/mansunkuo/hello-world-2026

In Case You Don’t Have OpenCode

Use Structurizr playground with ChatGPT, Claude, or Gemini



https://playground.structurizr.com/
https://chatgpt.com/
https://claude.ai/
https://gemini.google.com/

All About an Architect

Architects Own Accountability

Building organizational trust through transparent architectural decisions

Take Ownership

Own team decisions and outcomes to earn organizational trust.

Be Transparent

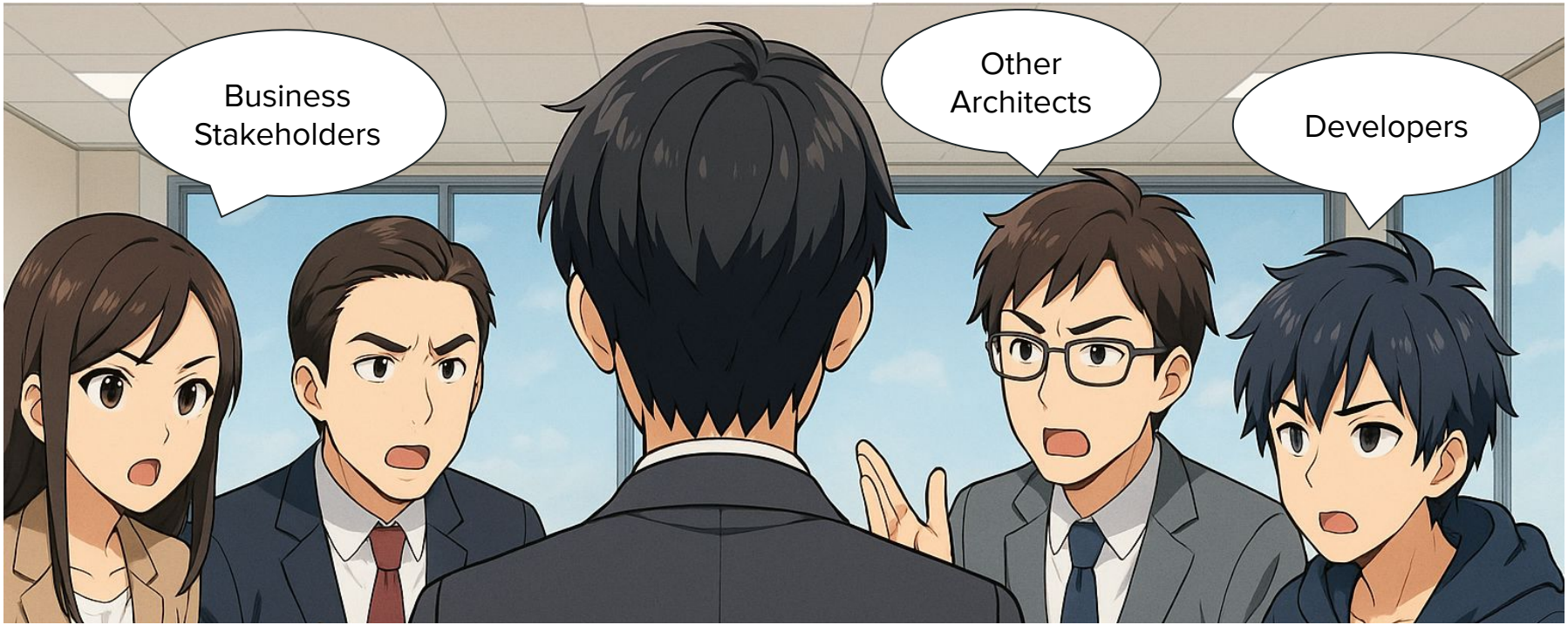
Share trade-offs and reasoning so everyone understands key choices.

Get It Done

Deliver reliably and step up when challenges.

Architects Bridge Stakeholder Needs

Aligning technical vision with business goals

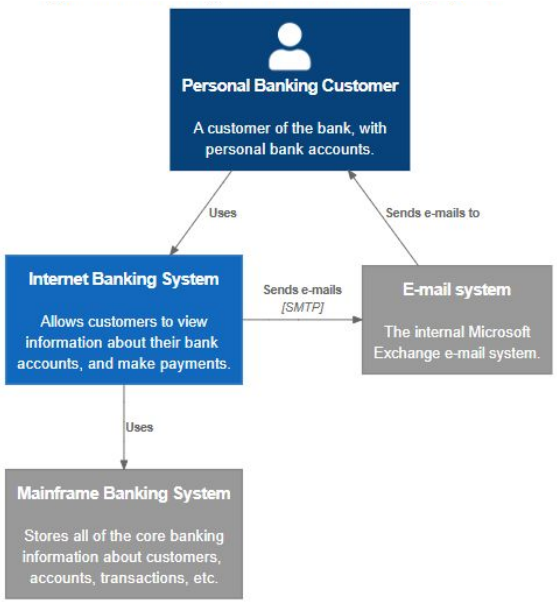


https://learning.oreilly.com/library/view/fundamentals-of-software/9781098175504/ch25.html
Generated by ChatGPT

Diagram as Code

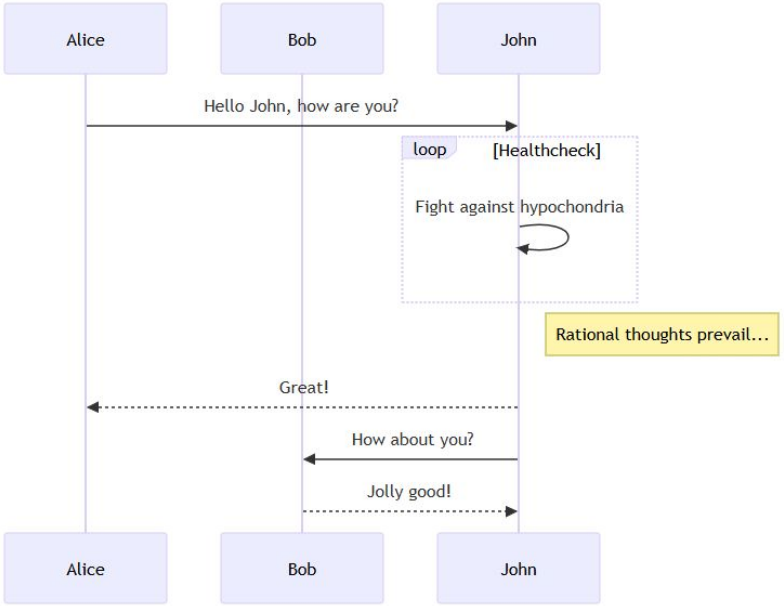
Maintainable, version-controlled, and AI-accelerated diagrams

Structurizr or C4 PlantUML



https://kroki.io/examples.html

Sequence Diagram - Mermaid



Context Before You Get Started

Functional Requirements

- Core functionality
- Business goals

Non-Functional Requirements

- Scalability
- Performance
- Availability
- Security
- ...

Technical Constraints

- Cloud providers,
- Programming languages
- Framework

Existing Systems

Integration points and boundaries for existing systems.

Evolutionary Design

Beginning with a primitive whole





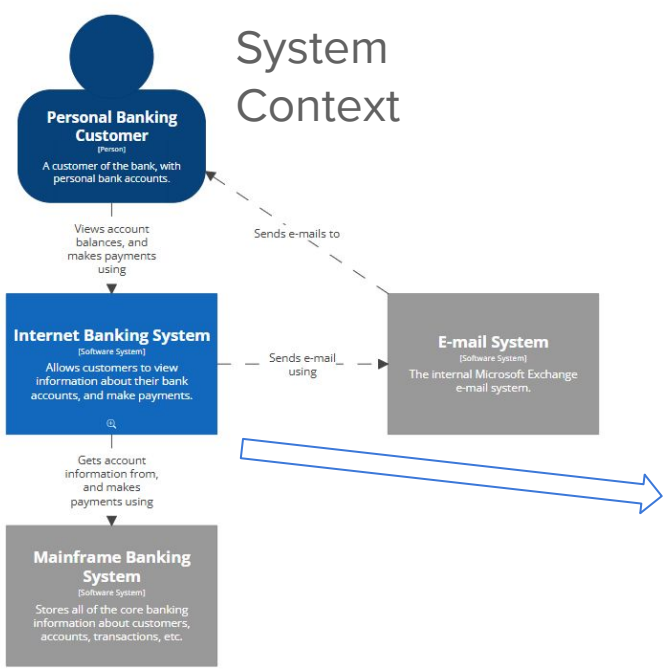


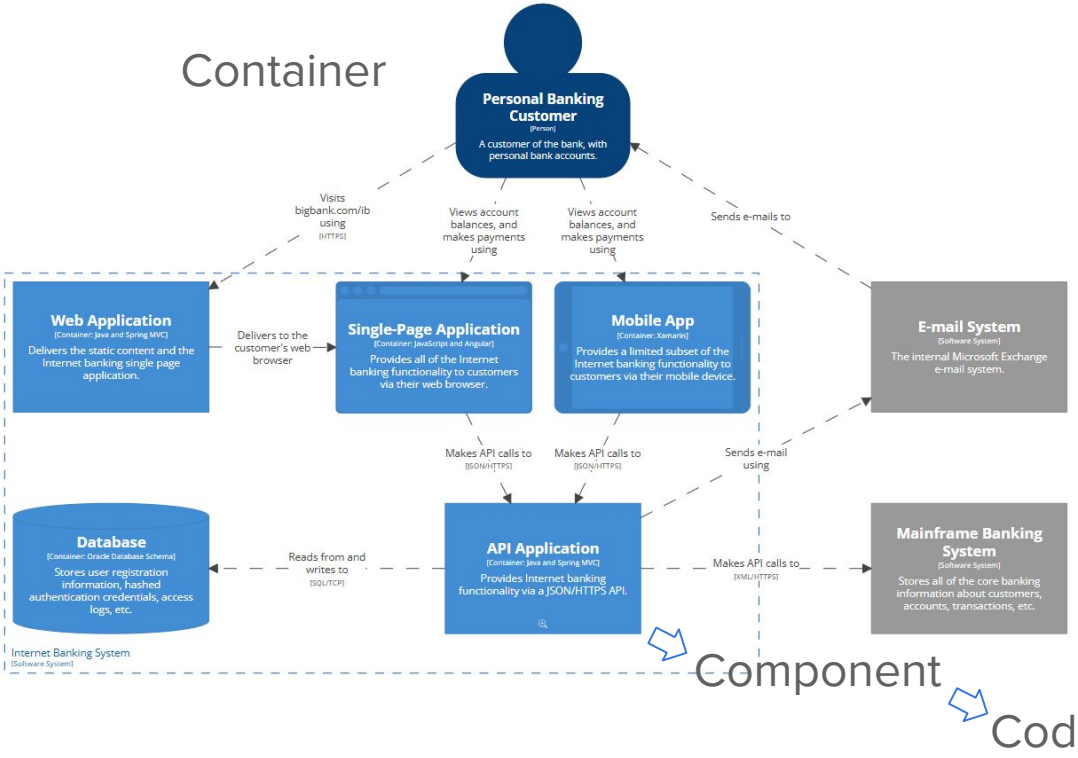
https://www.industriallogic.com/blog/evolutionary-design/
Generated by Gemini

C4 Model & Structurizr

Four Levels of Abstraction

C4: Systems **C**ontext ➔ **C**ontainers ➔ **C**omponents ➔ **C**ode



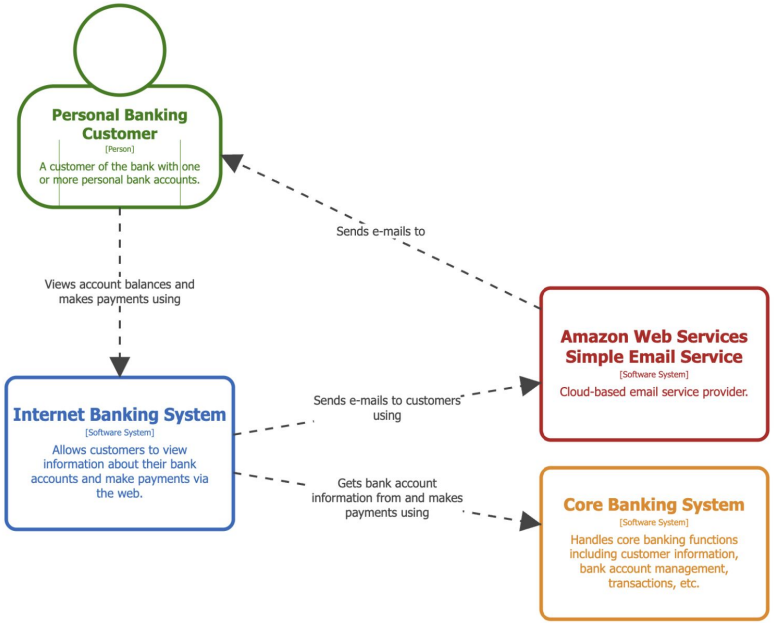


https://c4model.com/example/
https://c4model.com/abstractions

System Context Diagram

A high-level view showing value delivered to stakeholders without focusing on low-level technologies or protocols

- **Value**: The highest level of abstraction that delivers business value
- **Abstraction**: The boundary of a system could be an **organization**, **infrastructure**, or **business goal**.
- **People**: Focus on how actors, roles, personas interact with software systems rather than technologies, protocols and other low-level details.

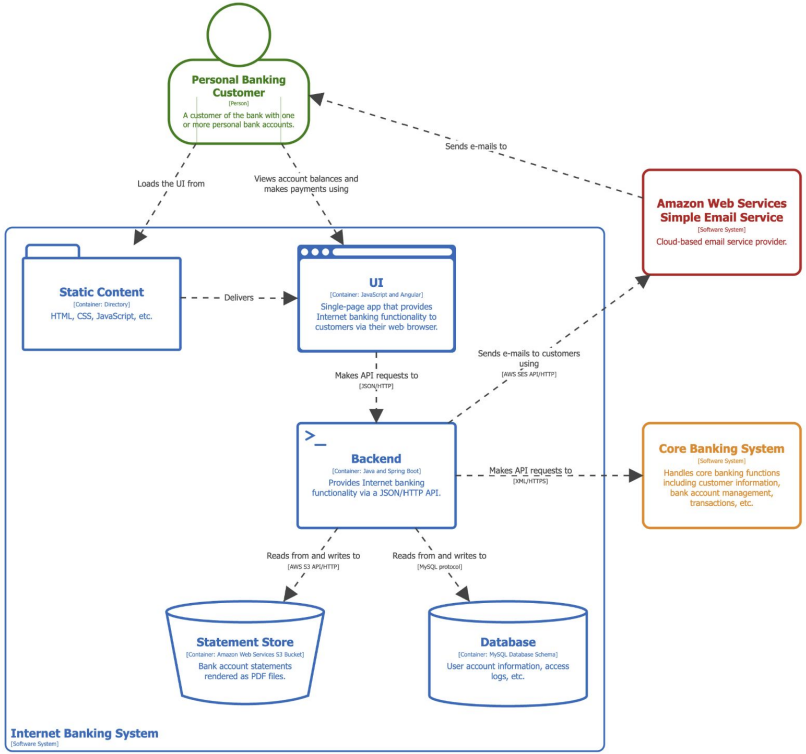


https://c4model.com/diagrams/system-context
https://c4model.com/abstractions/software-system

Container Diagram

A simple, high-level technology focussed diagram for software developers and support/operations staff

- C4 container
  - Not a Docker container
  - An application or a data store
  - A runtime boundary around code being executed or data being stored
- Multiple C4 containers can run within the same container, VM, or host instance.

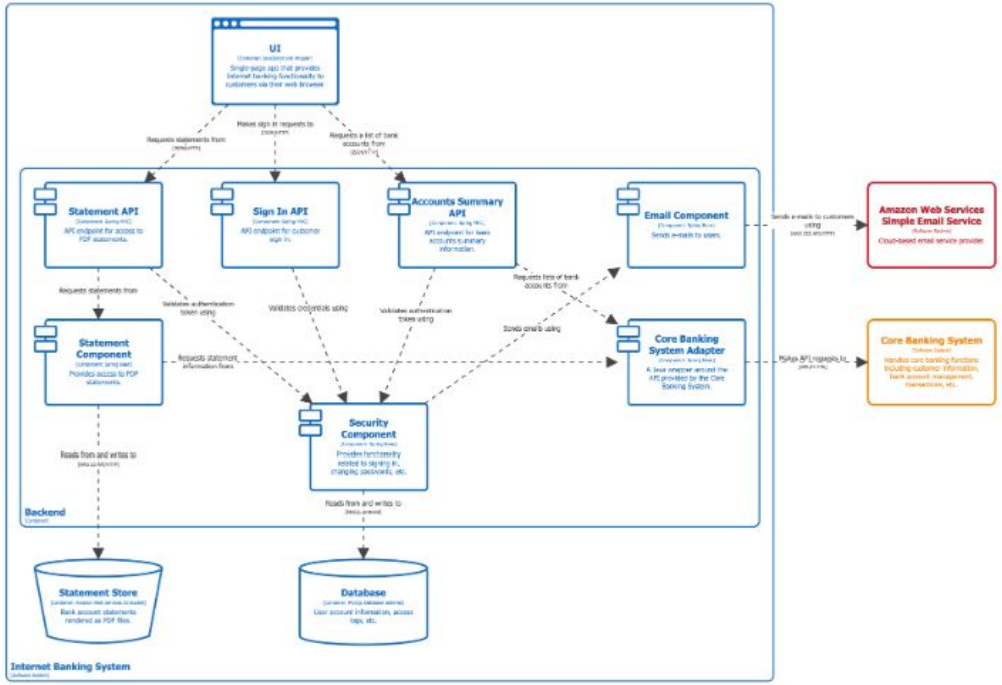


https://c4model.com/diagrams/container
https://c4model.com/abstractions/container

Component Diagram

A diagram that illustrates the logical building blocks of a container and their dependencies

- **Abstraction**: Shows code-level abstractions like classes and modules that help you to understand how it works.
- **Noise Reduction**: Filters out low-level noise to highlight the structural architecture.
- **Best Practice**: Only create component diagrams if you feel they add value, and consider automating it.



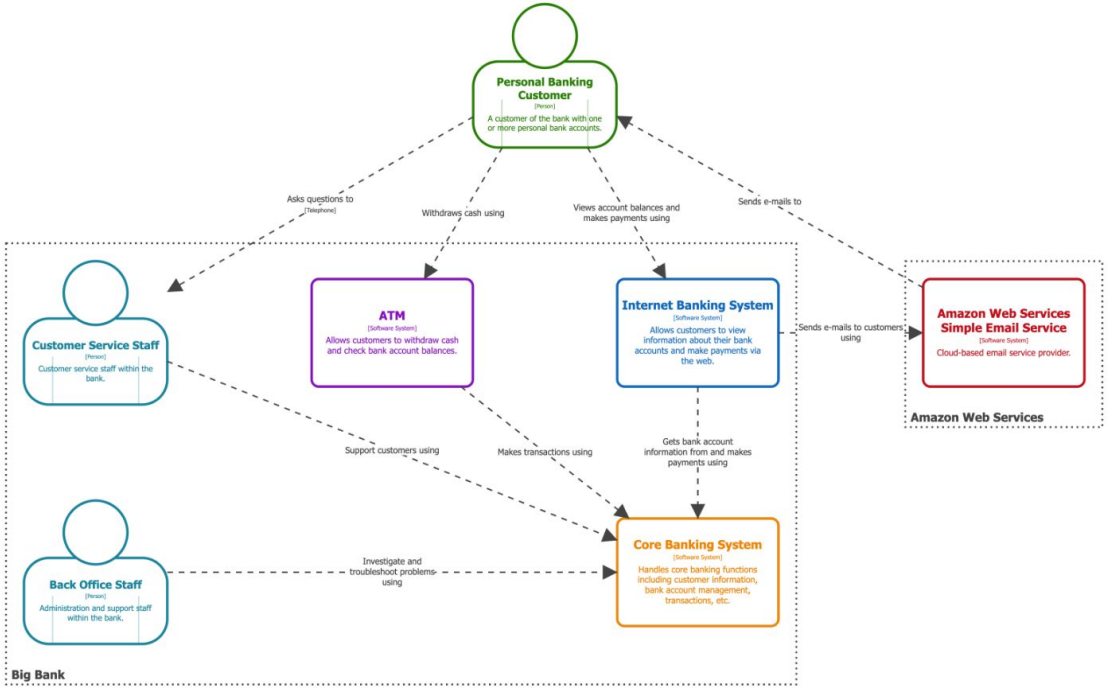
https://c4model.com/diagrams/component
https://c4model.com/abstractions/component

Code Diagram

- Ignored in most cases
- Check details with IDE

System Landscape Diagram

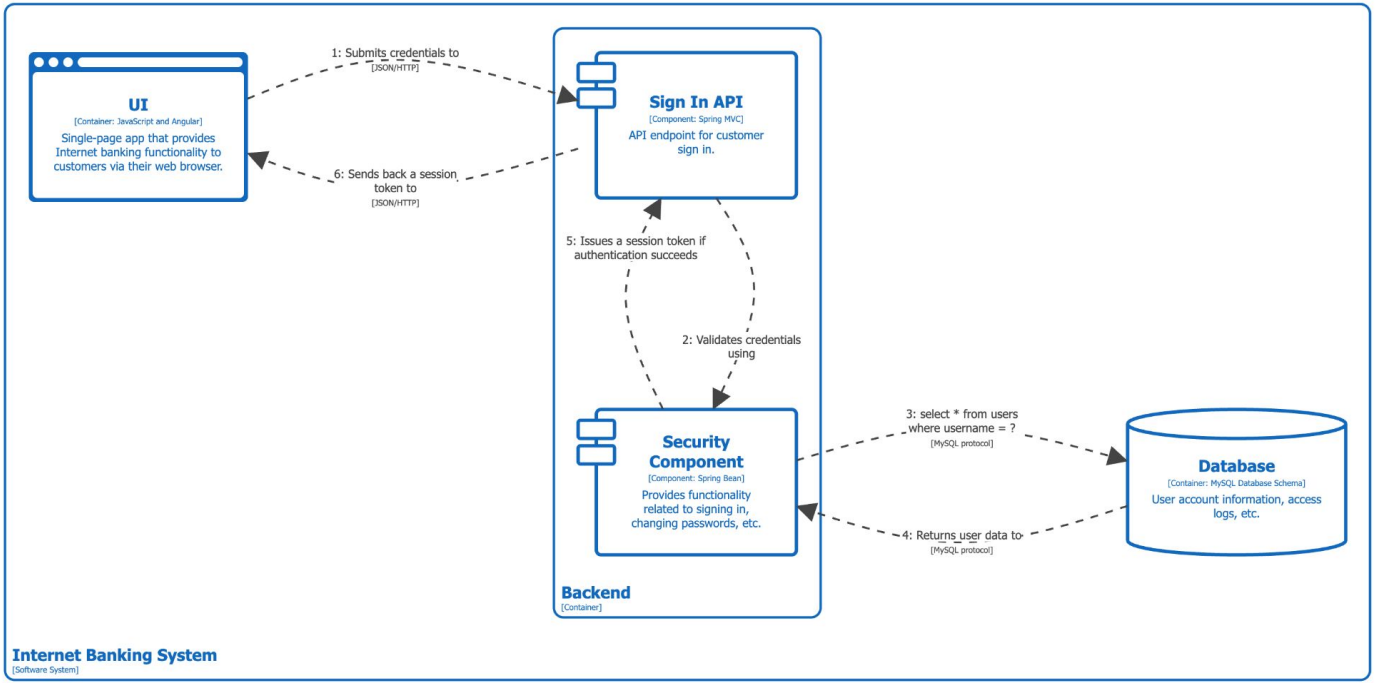
A system context diagram without a specific focus on a particular software system



https://c4model.com/diagrams/system-landscape

Dynamic Diagram - Collaboration Style

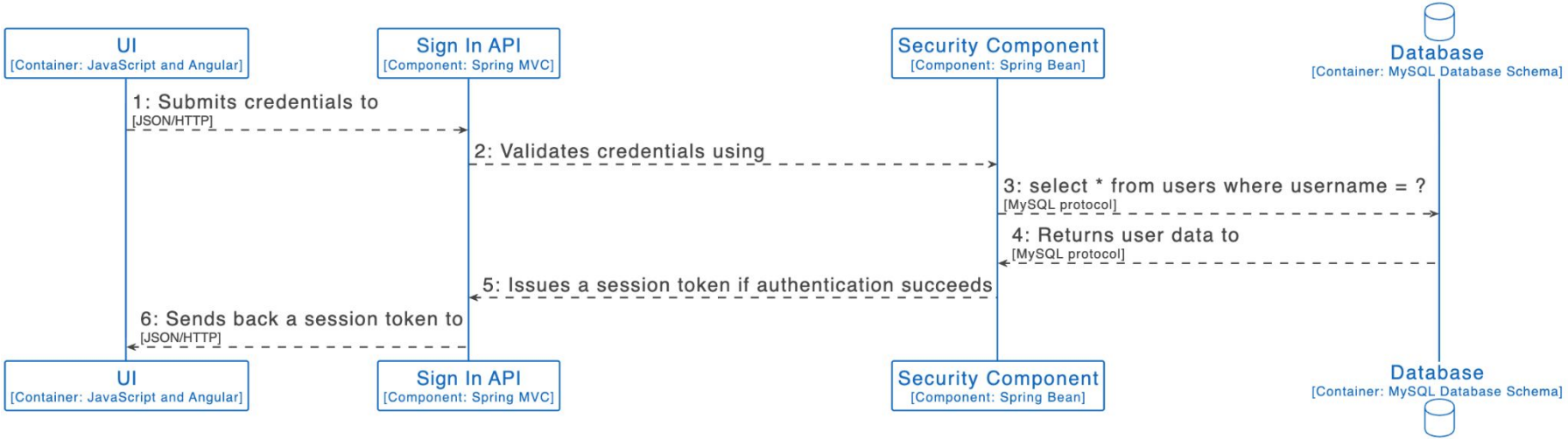
A diagram shows how elements collaborate at runtime to implement a user story, use case, feature, etc.



https://c4model.com/diagrams/dynamic
https://c4model.com/example/#Dynamic-Collaboration
https://en.wikipedia.org/wiki/Communication_diagram

Dynamic Diagram - Sequence Style

Integrating PlantUML, Mermaid, and Kroki diagrams into your workspace via Image Views

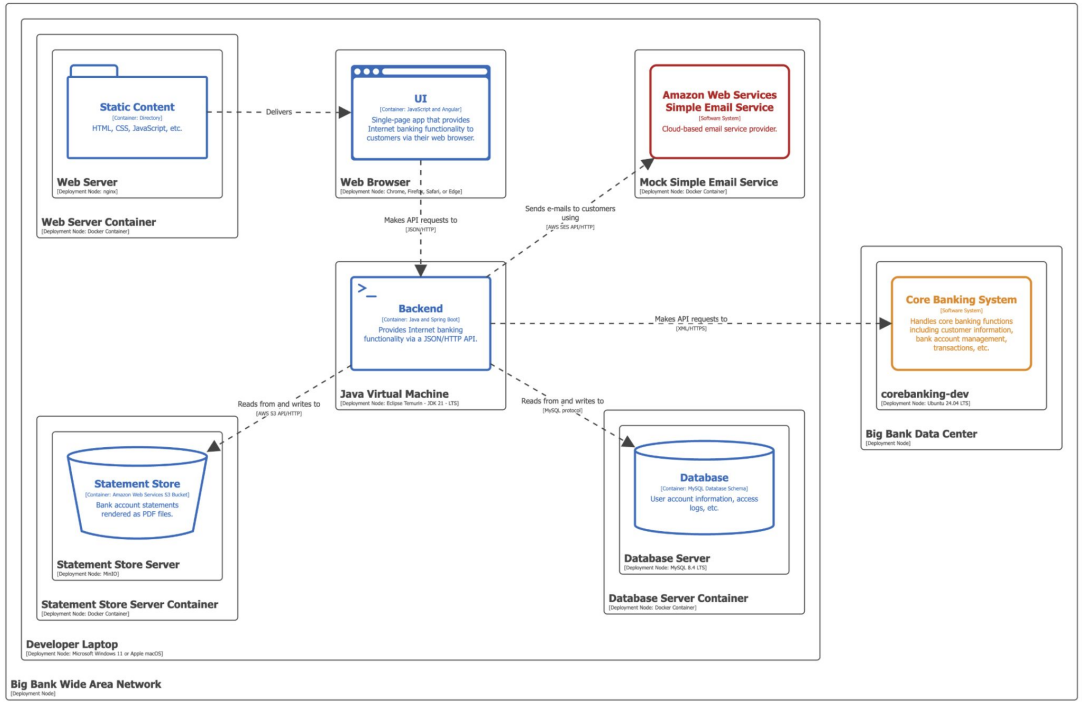


https://c4model.com/example/#Dynamic-Sequence
https://en.wikipedia.org/wiki/Sequence_diagram
https://docs.structurizr.com/dsl/cookbook/image-view/

Deployment diagram

A diagram to illustrate how systems and/or containers are deployed on to the infrastructure within a given deployment environment

- **Deployment node**: represents where a software system or container is running
- **Infrastructure node**: interact with or route traffic between deployment nodes such as DNS, load balancers, firewalls, etc.



Architectural Decision Records (ADR)

Alignment Without Authority

How architects create alignment without formal authority

Vertical Visibility

Translates technical choices into business context for leadership.

See the context

Horizontal Alignment

Connects engineering, security, and product around shared priorities.

Connect the teams

Reason Over Authority

Makes trade-offs explicit instead of relying on organizational rank.

Make the trade-off explicit

Architectural Context Lost Over Time

When teams lose track of why decisions were made, simple work becomes costly friction

ROOT CAUSES

- Team Member Turnover
- Undocumented Silo
- Time & Scale Decay

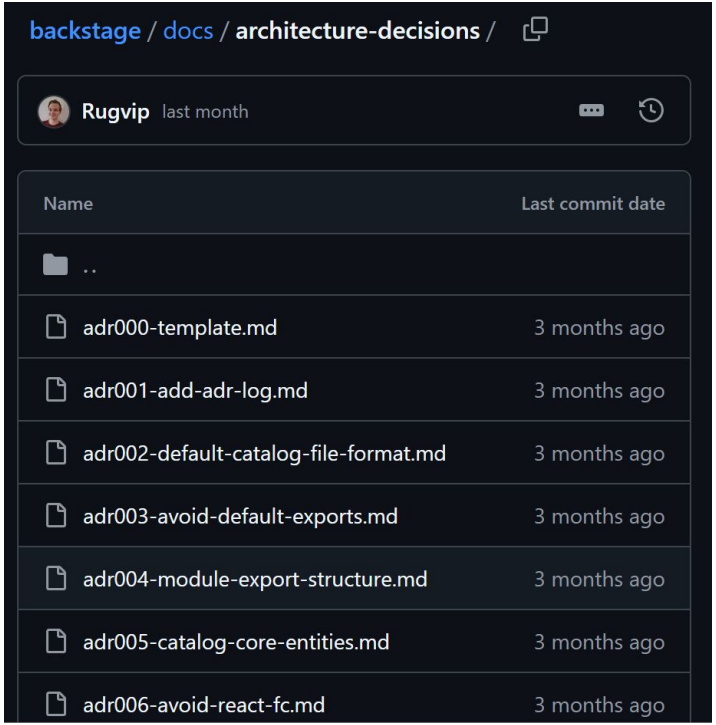
CONSEQUENCES

- Repeated Discussion
- Accidental Violations
- Fear of System Changes

Architectural Decision Records (ADR)

A document for a justified design choice

- Title
- Status
  - Proposed, accepted, rejected, deprecated, ...
- Context
  - What is the issue that we're seeing that is motivating this decision or change?
- Options
  - Pros and Cons for alternative choices
- Decision
  - What is the change that we're proposing and/or doing?
- Consequences
  - What becomes easier or more difficult to do because of this change?



https://adr.github.io/
https://github.com/joelparkerhenderson/architecture-decision-record/tree/main/locales/en/templates/decision-record-template-by-michael-nygard
https://www.thoughtworks.com/en-us/radar/techniques/lightweight-architecture-decision-records
https://github.com/backstage/backstage/tree/master/docs/architecture-decisions

Lab

Toy Preorder Lottery Service

You are the system vendor for an agent. The agent **already has a real-name verification app** where buyers log in and complete real-name verification, and it already has a **store system**. The agent wants to run toy preorders at the **lowest possible cost**:

- Operators set each preorder campaign's **period** and the **total number of toys each buyer may enter for**;
- Buyers register for the items they want to enter during the campaign period;
- When the campaign **closes, a lottery** picks the winners;
- Winners choose which **store** they want to pick their goods up from.

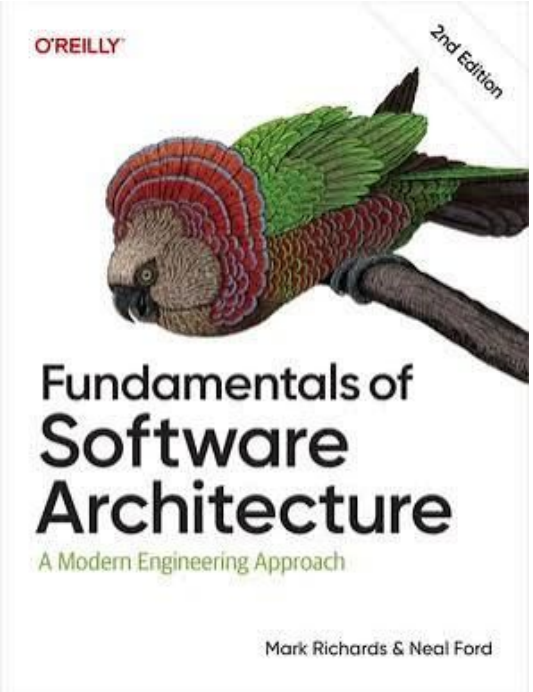
Our job is to **plug all of this into the existing real-name app**, not to build a separate buyer app, and not to rebuild the store system.

https://mansunkuo.github.io/hello-world-2026/
https://github.com/mansunkuo/hello-world-2026/blob/main/docs/workshop.md
https://github.com/mansunkuo/hello-world-2026/blob/main/docs/workshop-en.md

Key Takeaways

The Laws of Software Architecture

- Everything in software architecture is a trade-off
- Why is more important than how
- Most architecture decisions aren’t binary but rather exist on a spectrum between extremes.



https://learning.oreilly.com/library/view/fundamentals-of-software/9781098175504/ch27.html

References

- https://github.com/mansunkuo/hello-world-2026
- https://simonbrown.je/workshops/
- https://adr.github.io/
- https://learning.oreilly.com/library/view/fundamentals-of-software/9781098175504/

Thanks