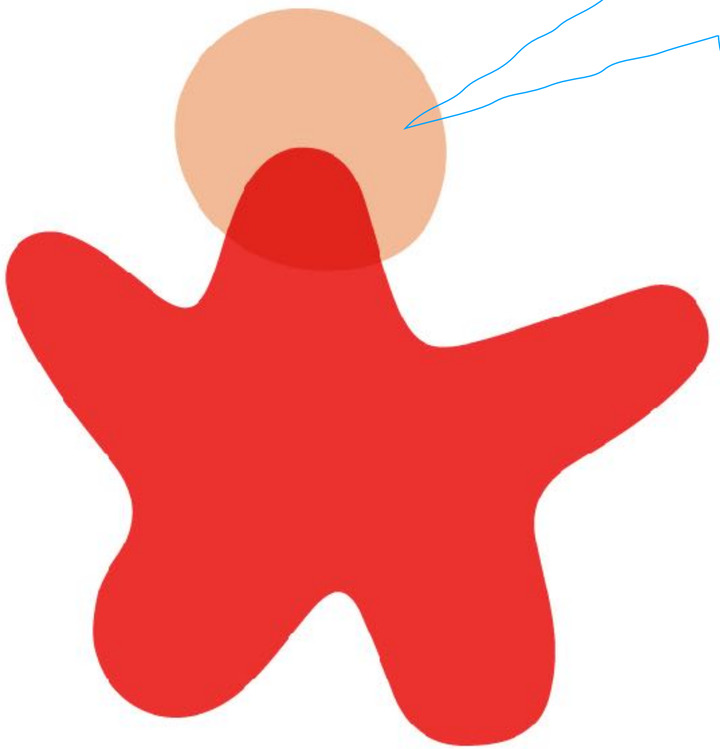
O'REILLY® LIVE ONLINE TRAINING

Describing Software Architecture

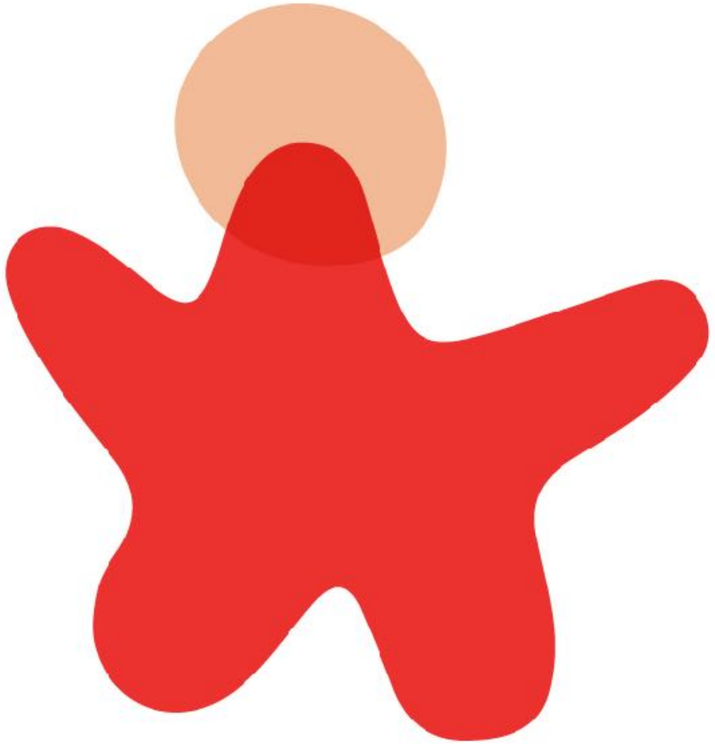
Neal Ford
Thoughtworks
Director / Software Architect / Meme Wrangler
http://www.nealford.com
@neal4d

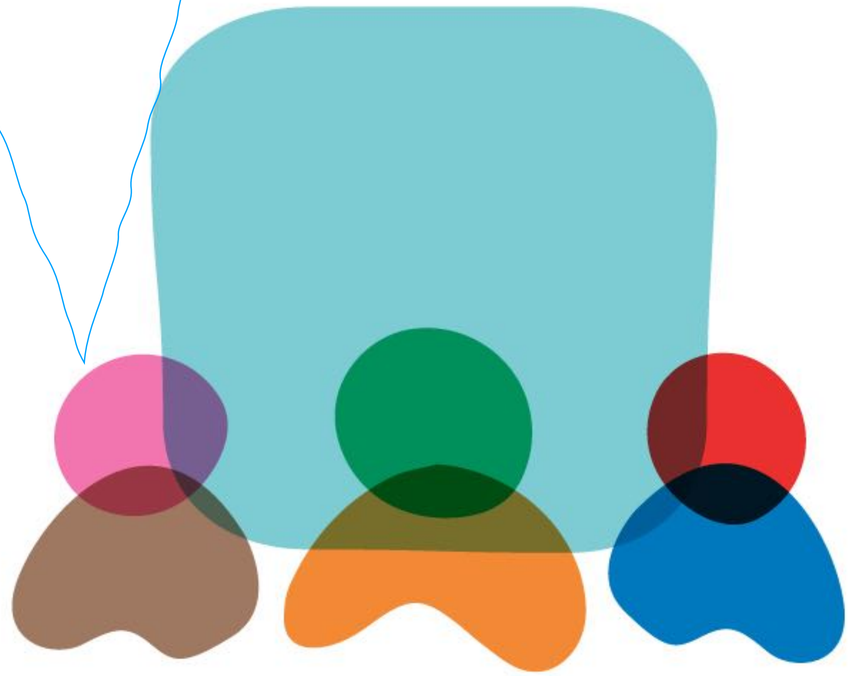
Mark Richards
Independent Consultant
Hands-on Software Architect, Published Author
Founder, DeveloperToArchitect.com
@markrichardssa



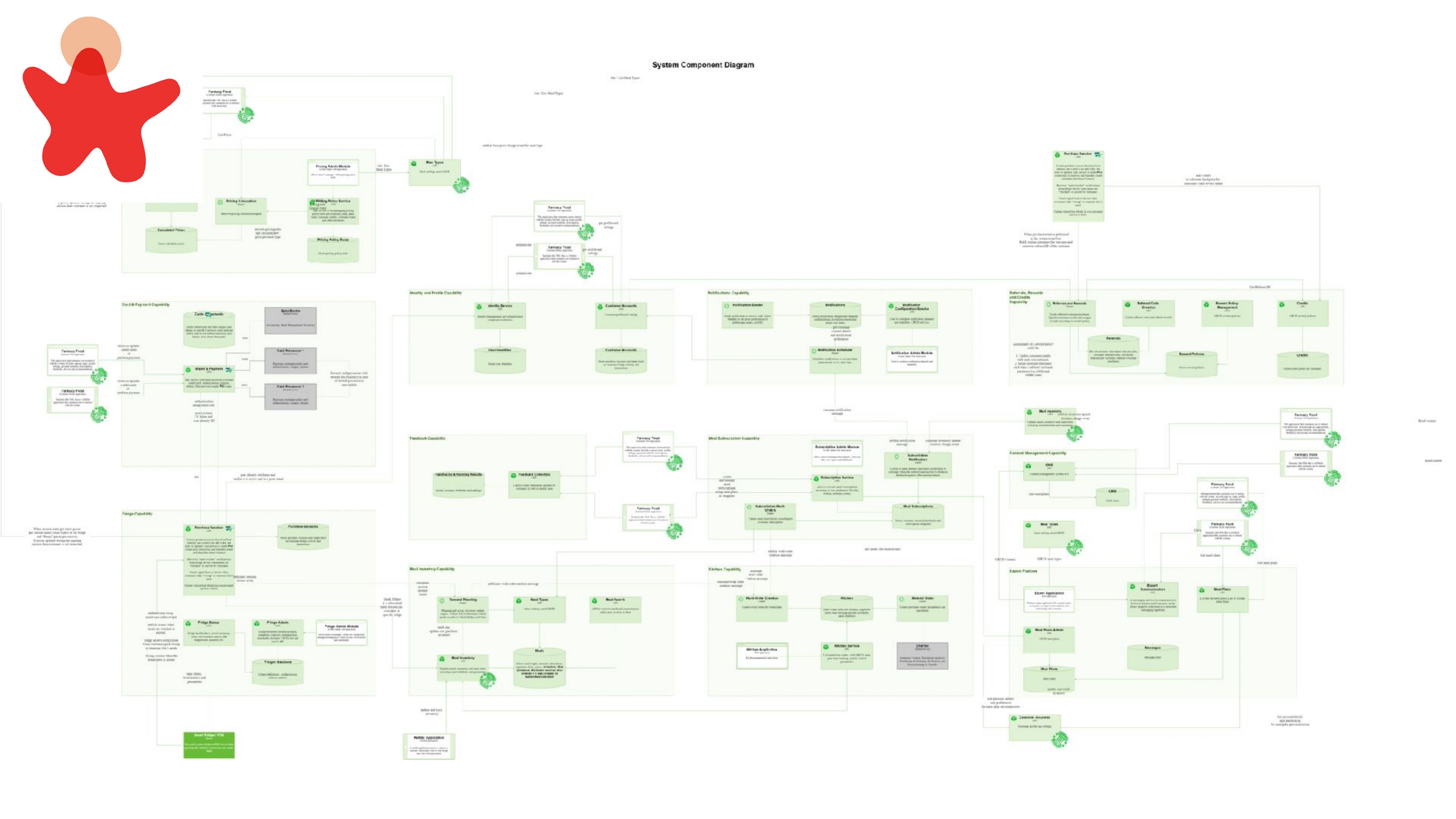
OK, I’ve finished creating the architecture for the the Farmacy Food system!

Great! Let’s see your solution.

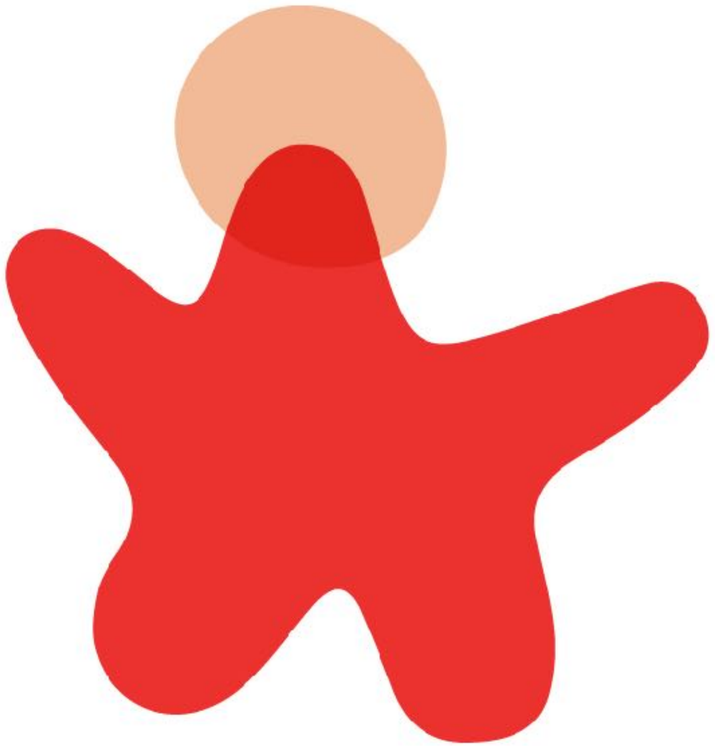


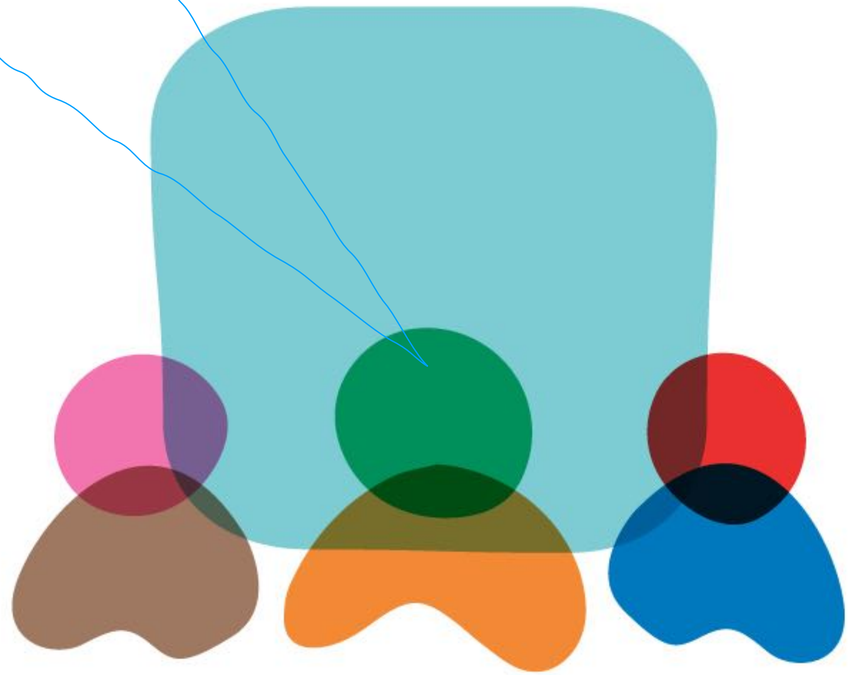


System Component Diagram

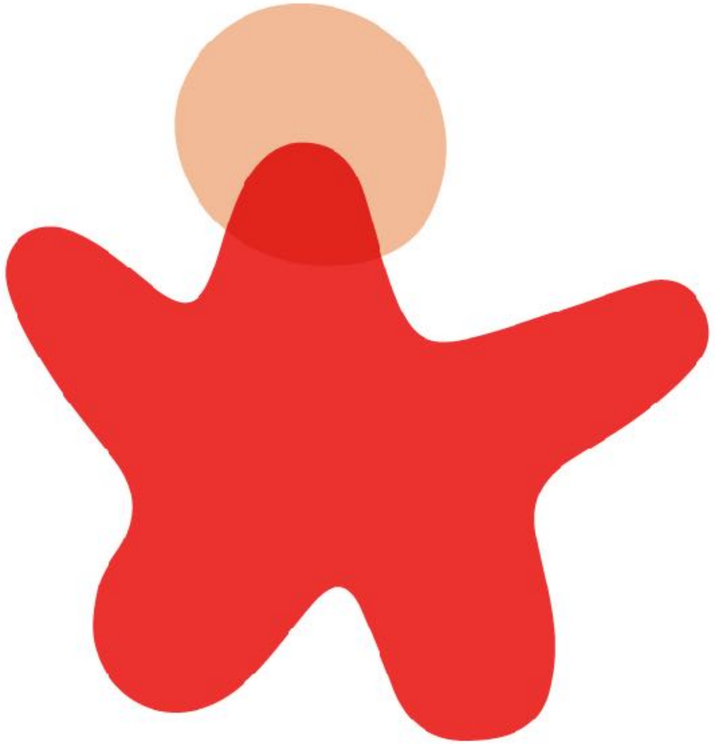


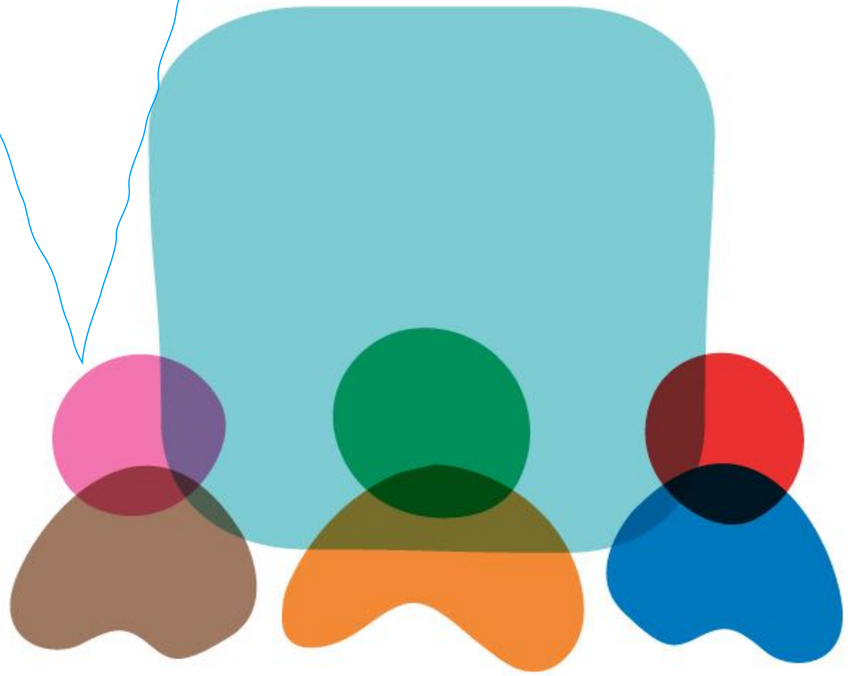
?!!?



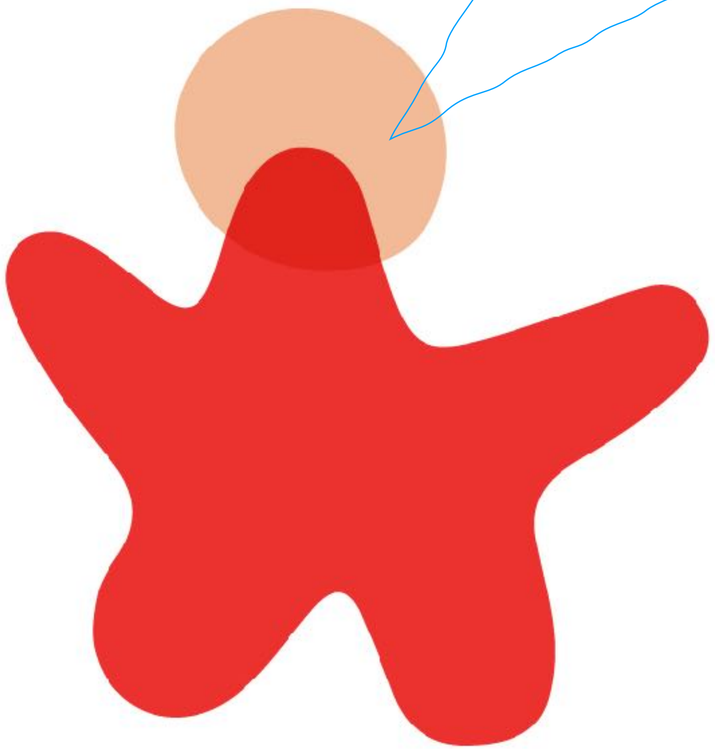


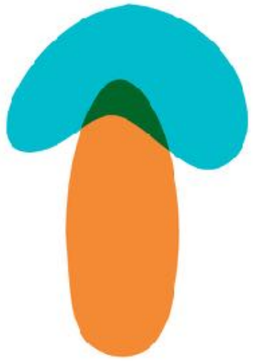
What in the world is this ?!?
Can’t you describe this in a way we can understand it?





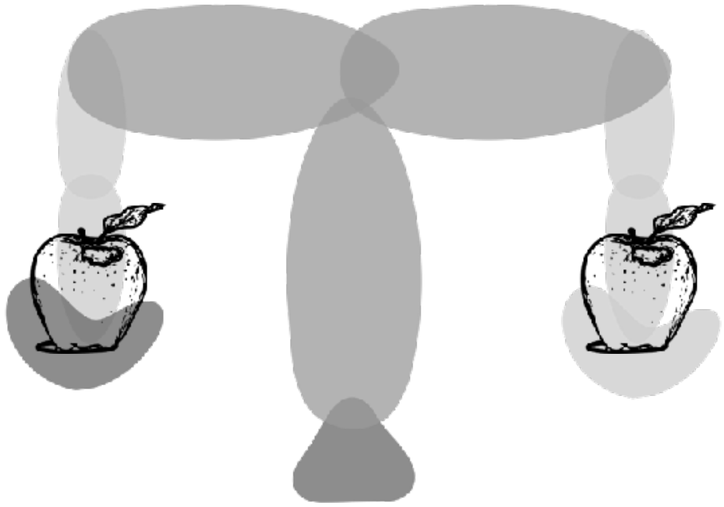
?!!?





Higher level of abstraction

Describing architecture



Corpus of solutions to the same problem

farmacytood.com

Farmacy Food

ABOUT US INTERESTED? Sign Up 0

Let Food be Thy Medicine

LEARN MORE >

Farmacy Food

Healthy, locally sourced meals for delivery or pick-up.

GET STARTED

Customer
using the app

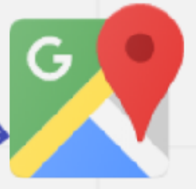
Farmacy Food system

PayPal (or other payment gateway provider)

Auth0 (or other authentication provider)

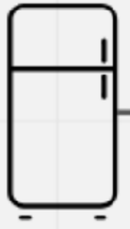
Farmacy Food central kitchen

eDietitian (expert system for diet recommendation)



Google maps (or other map provider)

Customer
(in person)



Smart fridge

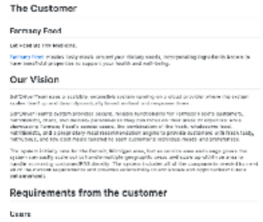
Smart fridge mgmt system (in the cloud)

Customer
(in person)

Cashier POS

Vendor mgmt system (in the cloud)

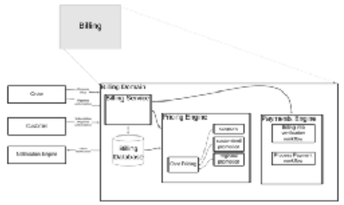
describing software architecture



architecture narratives

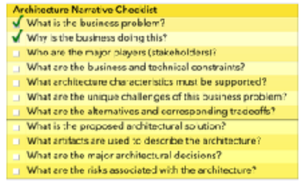
driving characteristics

architecture decisions



architecture diagrams

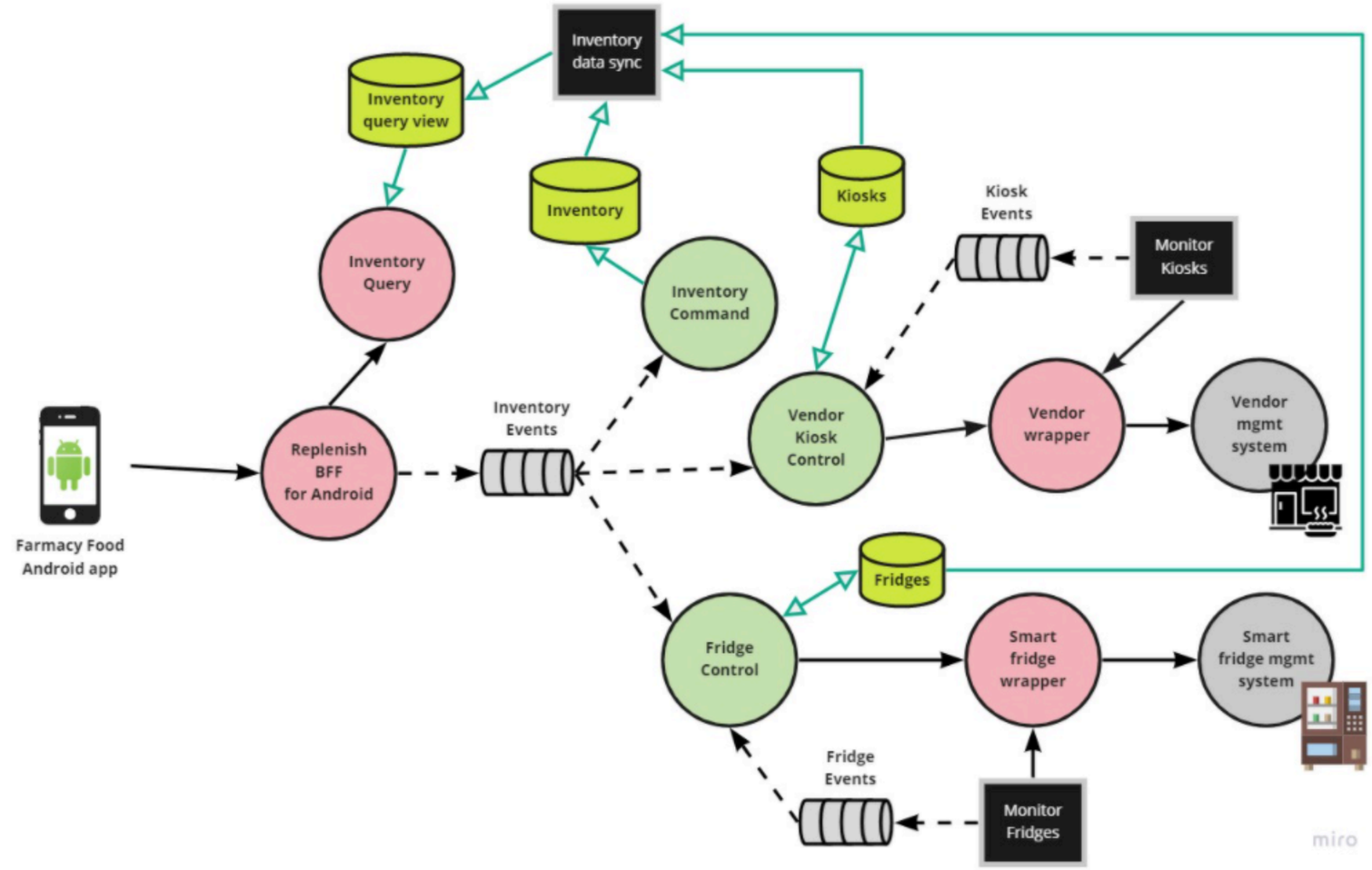
architecture solutions



winners and resources

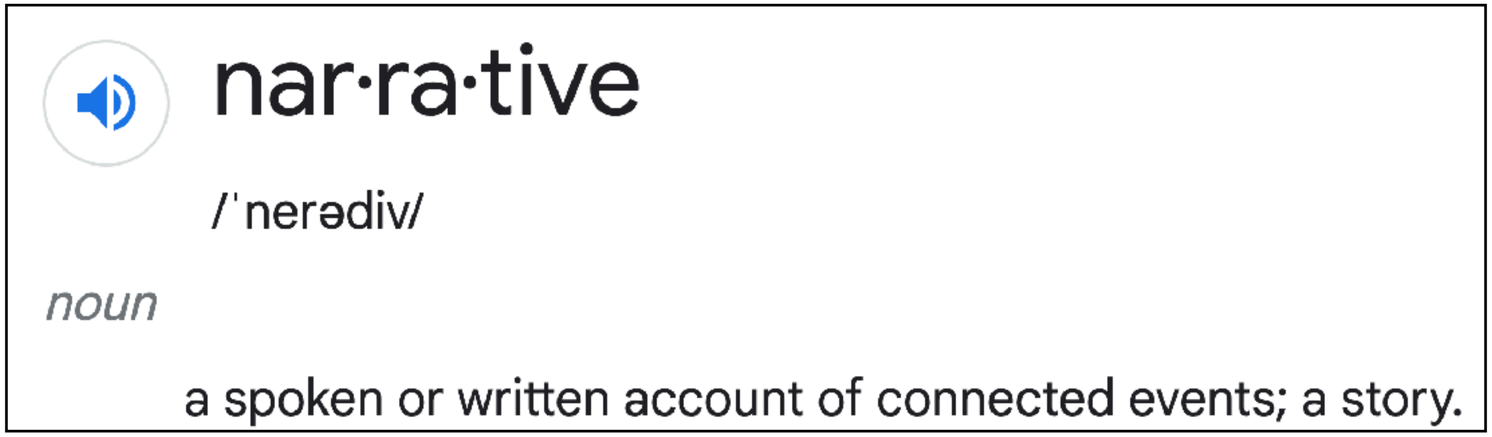
Architecture Narratives

How do you describe an architectural solution?



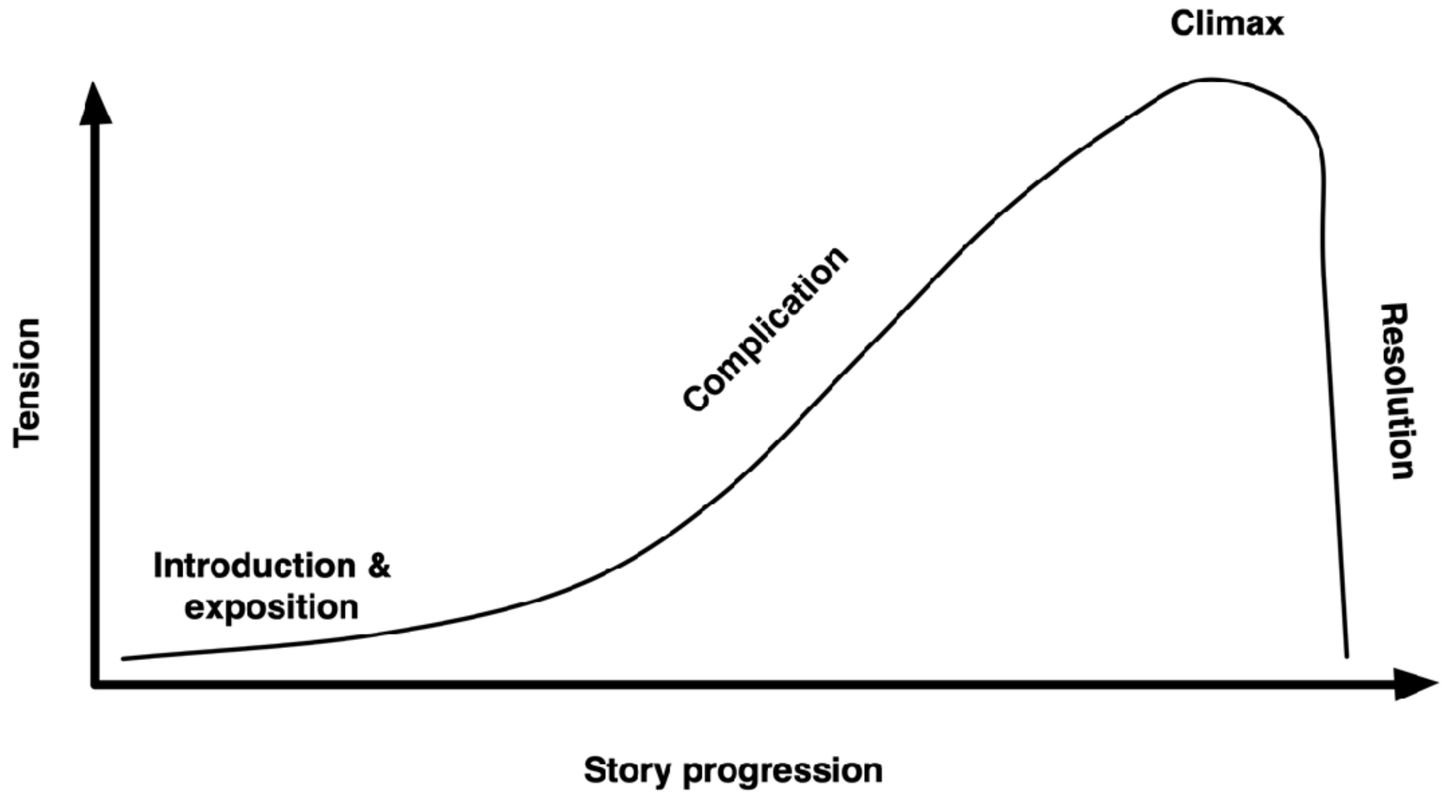
architecture narrative

A narrative tells the story of the architectural solution

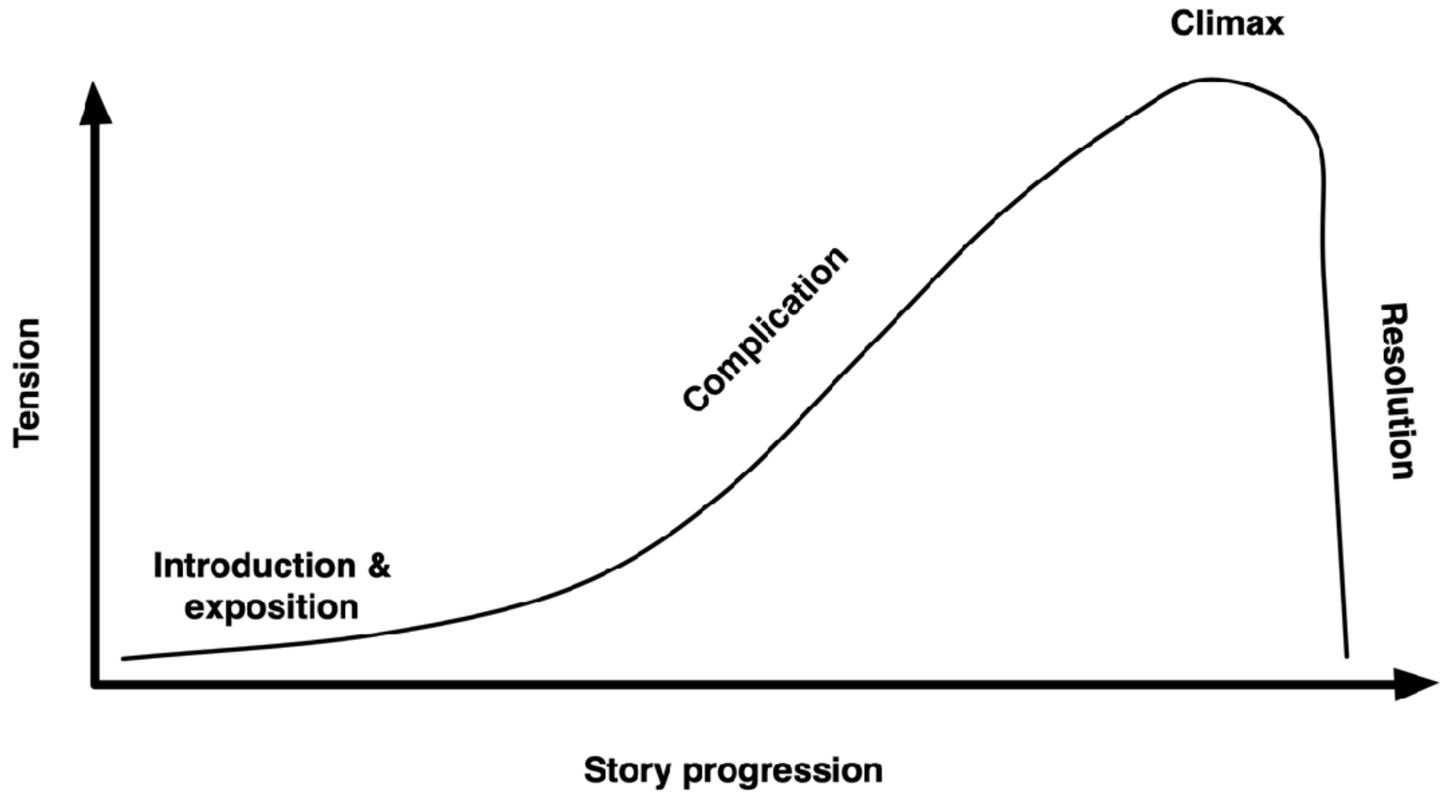


narrative arc and the three act structure

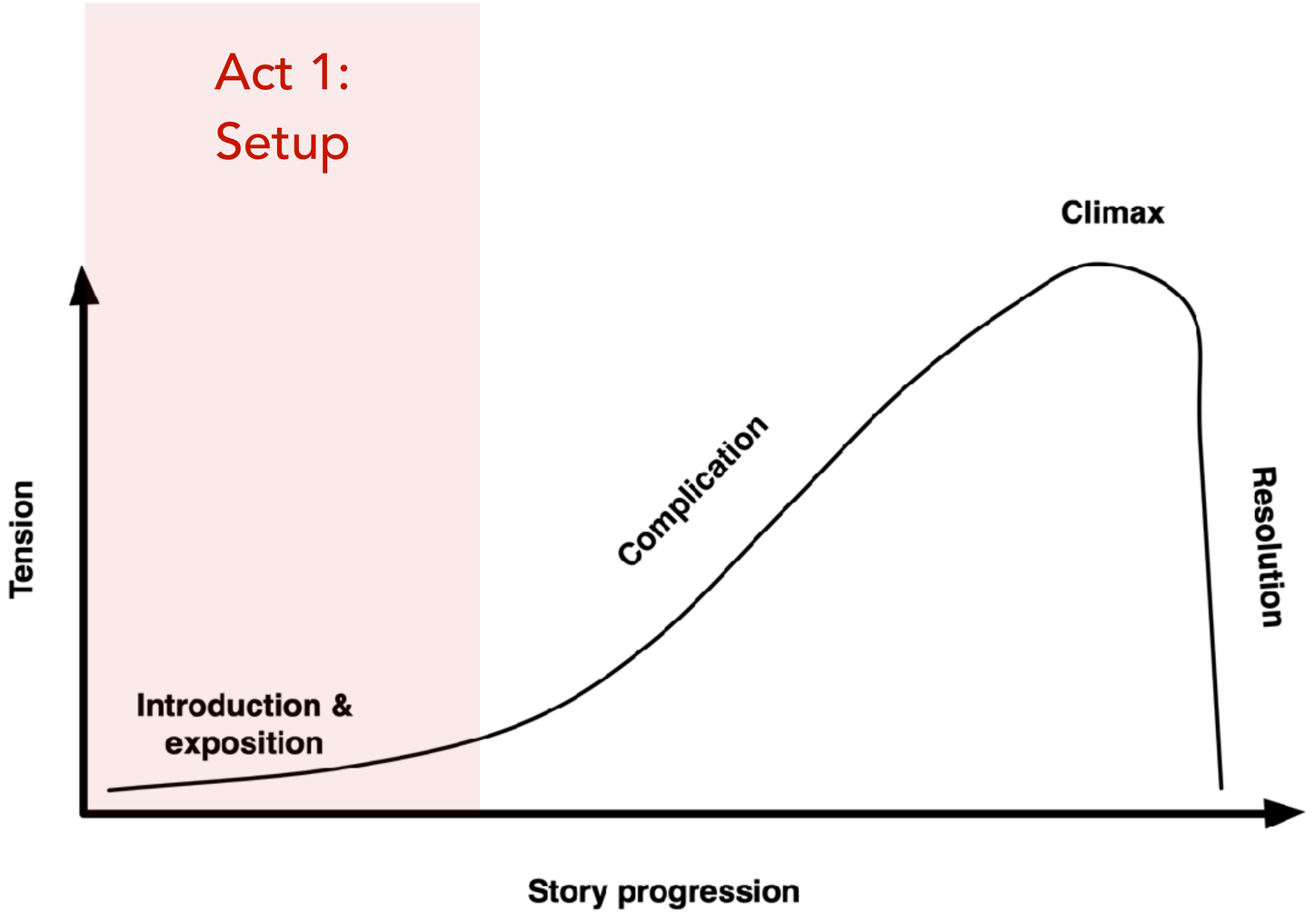
A *narrative arc* a literary term for the path a story follows. It provides a backbone by providing a clear beginning, middle, and end of the story



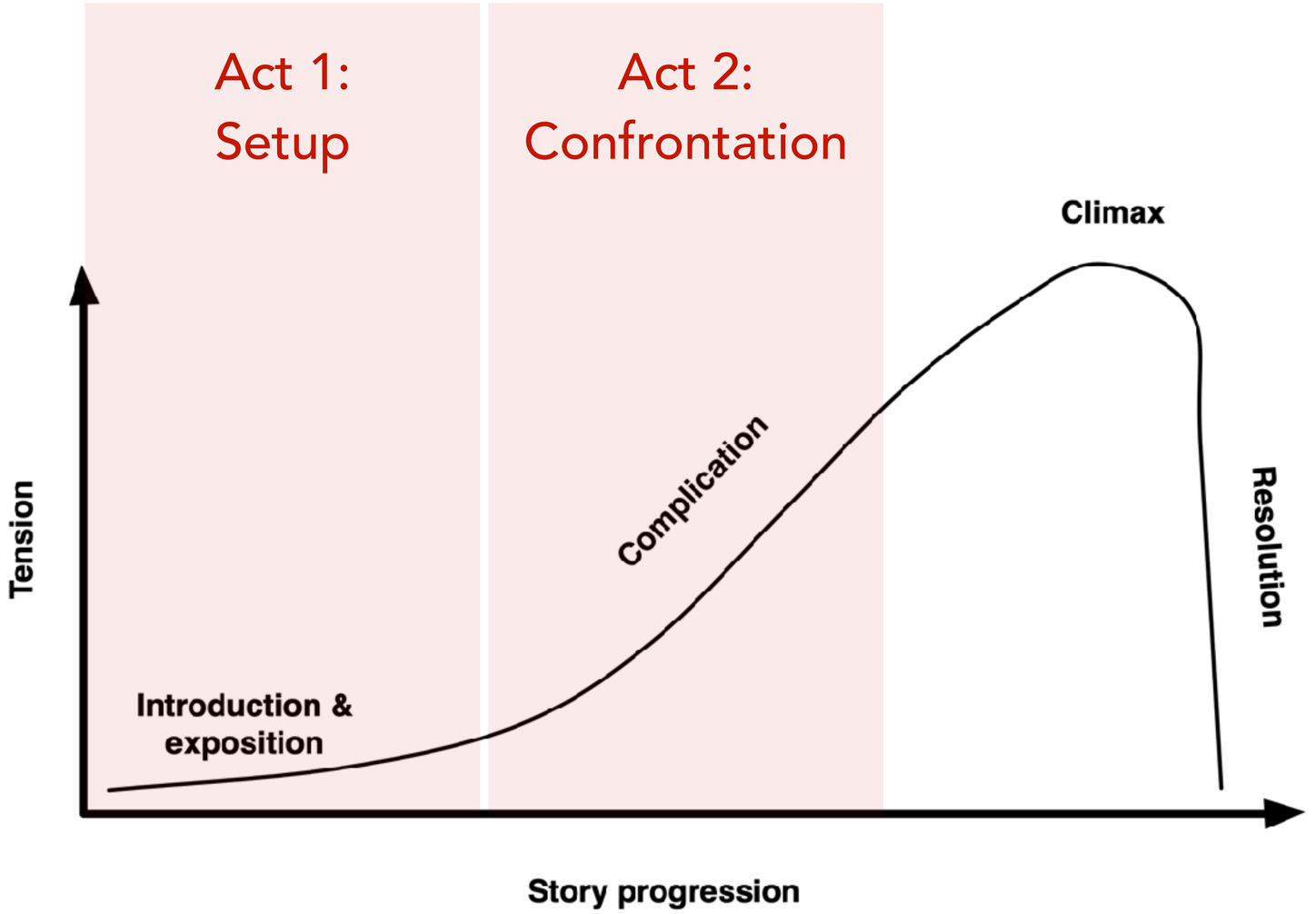
narrative arc and the three act structure



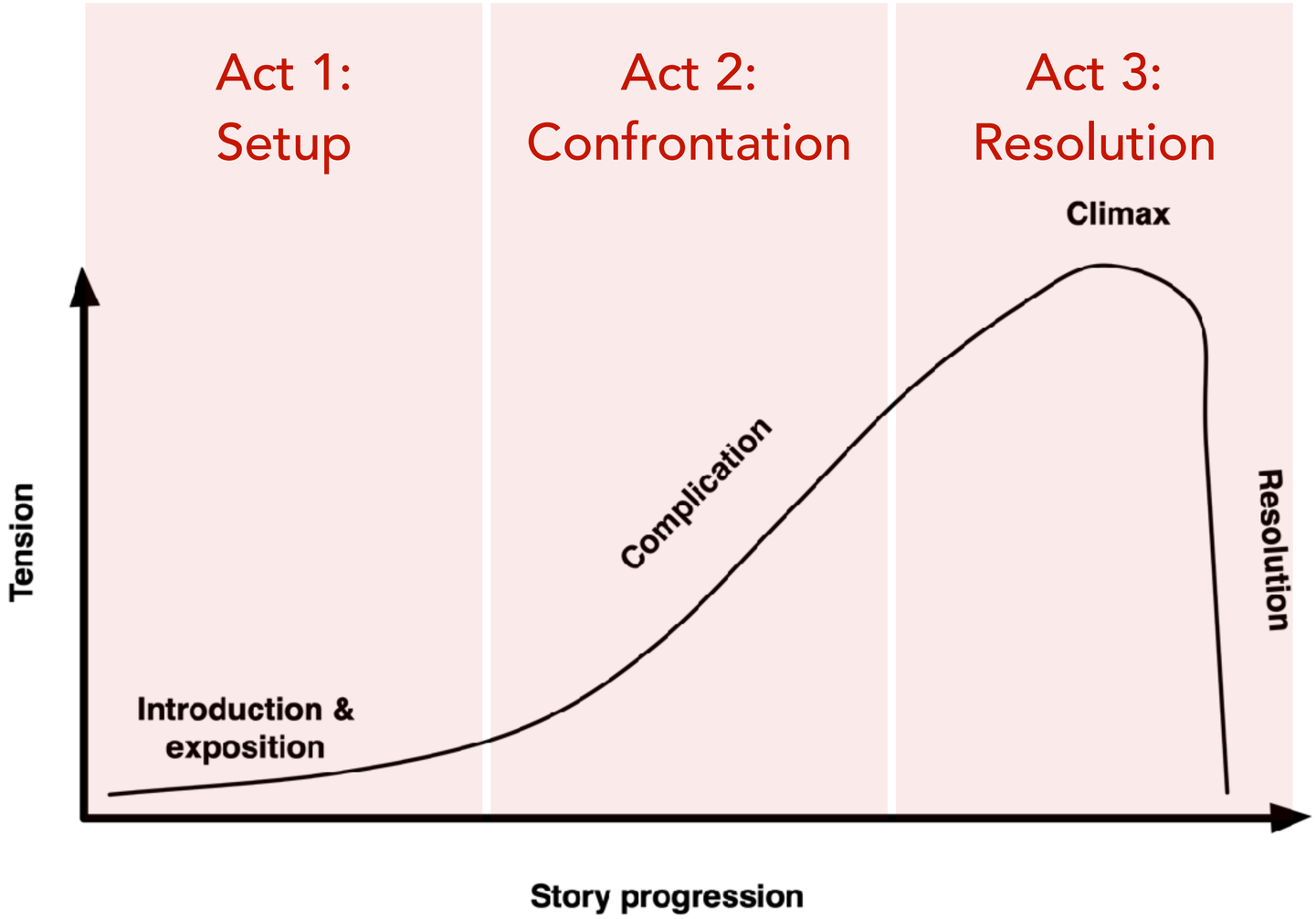
narrative arc and the three act structure



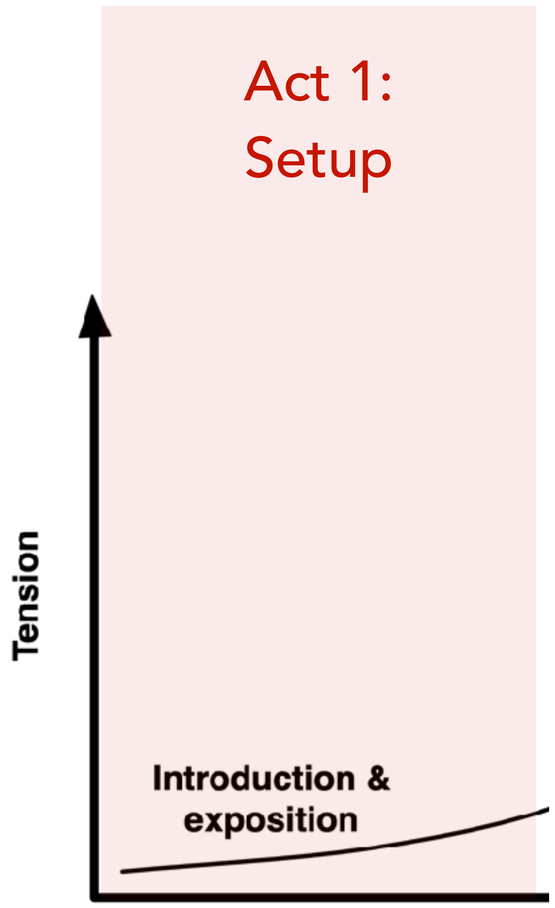
narrative arc and the three act structure



narrative arc and the three act structure



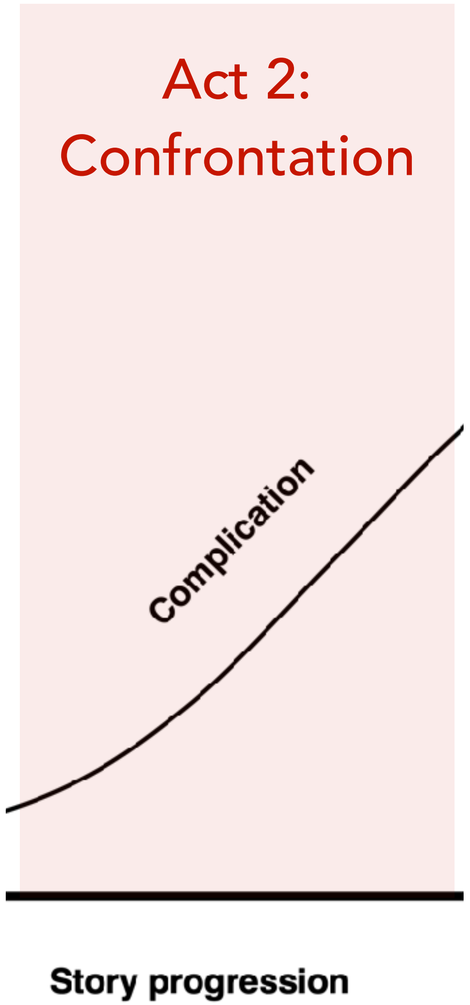
narrative arc and the three act structure



What is the business problem we are trying to solve?

- ☑ What is the business problem?
- ☑ Why is the business doing this?
- ☑ Who are the major players (stakeholders)?

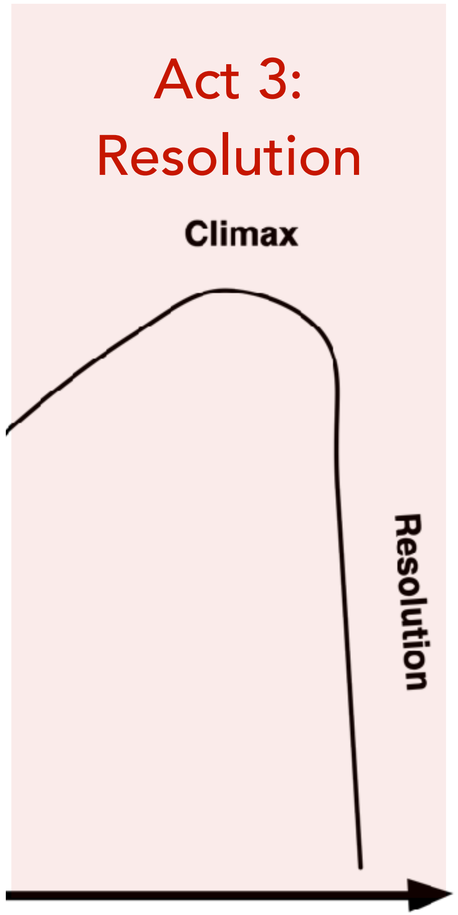
narrative arc and the three act structure



What are the conditions and constraints?

- ☑ What are the business and technical constraints?
- ☑ What architecture characteristics must be supported?
- ☑ What are the unique challenges of this business problem?
- ☑ What are the alternatives and corresponding tradeoffs?

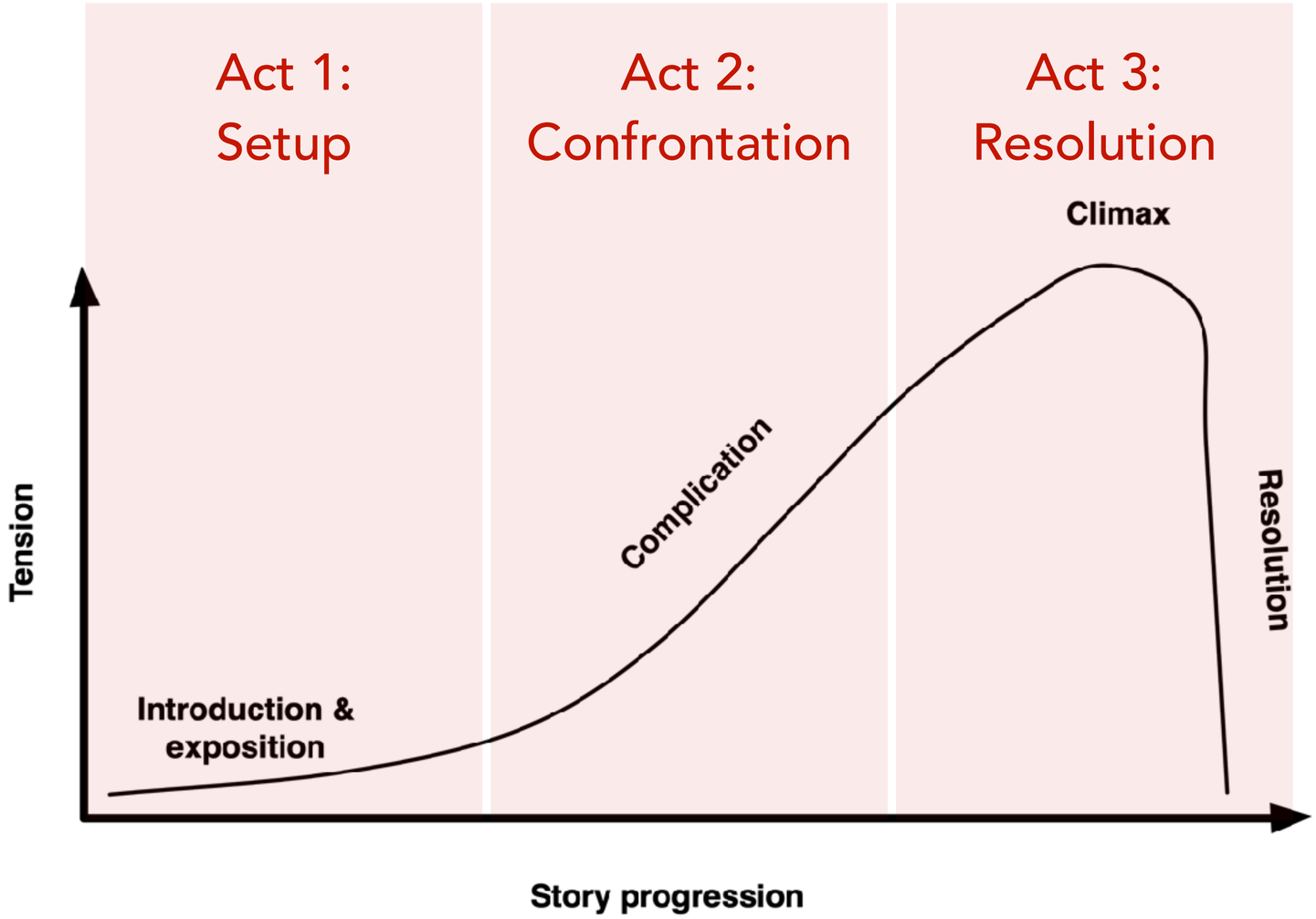
narrative arc and the three act structure



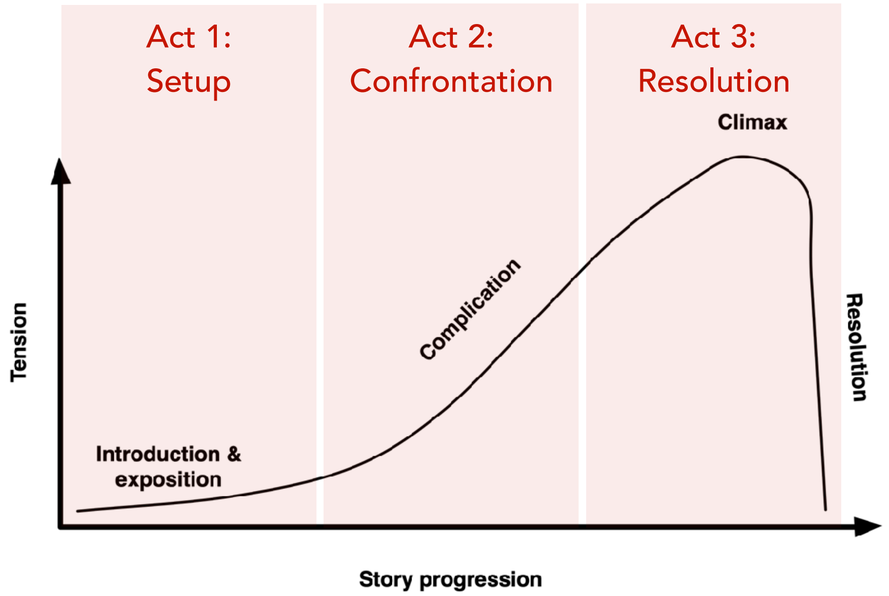
What is the proposed solution?

- ☑ What is the proposed architectural solution?
- ☑ What artifacts are used to describe the architecture?
- ☑ What are the major architectural decisions?
- ☑ What are the risks associated with the architecture?

narrative arc and the three act structure



Architecture Narrative Checklist
☐ What is the business problem?
☐ Why is the business doing this?
☐ Who are the major players (stakeholders)?
☐ What are the business and technical constraints?
☐ What architecture characteristics must be supported?
☐ What are the unique challenges of this business problem?
☐ What are the alternatives and corresponding tradeoffs?
☐ What is the proposed architectural solution?
☐ What artifacts are used to describe the architecture?
☐ What are the major architectural decisions?
☐ What are the risks associated with the architecture?



https://github.com/lookfwd/archkata

Jiakaturi

Act 1	Overview Vision Goals and Opportunities Use Cases
Act 2	Architecture Characteristics Design Constraints
Act 3	High-Level Architecture Mid-Level Architecture Milestones ADRs

Architecture Narrative Checklist
☐ What is the business problem?
☐ Why is the business doing this?
☐ Who are the major players (stakeholders)?
☐ What are the business and technical constraints?
☐ What architecture characteristics must be supported?
☐ What are the unique challenges of this business problem?
☐ What are the alternatives and corresponding tradeoffs?
☐ What is the proposed architectural solution?
☐ What artifacts are used to describe the architecture?
☐ What are the major architectural decisions?
☐ What are the risks associated with the architecture?

https://github.com/lookfwd/archkata

Jiakaturi

Act 1	Overview Vision Goals and Opportunities Use Cases
Act 2	Architecture Characteristics Design Constraints
Act 3	High-Level Architecture Mid-Level Architecture Milestones ADRs

Architecture Narrative Checklist
☐ What is the business problem?
☐ Why is the business doing this?
☐ Who are the major players (stakeholders)?
☐ What are the business and technical constraints?
☐ What architecture characteristics must be supported?
☐ What are the unique challenges of this business problem?
☐ What are the alternatives and corresponding tradeoffs?
☐ What is the proposed architectural solution?
☐ What artifacts are used to describe the architecture?
☐ What are the major architectural decisions?
☐ What are the risks associated with the architecture?

https://github.com/lookfwd/archkata

Jiakaturi

Act 1

- Overview
- Vision
- Goals and Opportunities
- Use Cases
- Architecture Characteristics
- Design Constraints
- High-Level Architecture
- Mid-Level Architecture
- Milestones
- ADRs

I. Overview

Farmacy Food is a healthy food startup that takes the “let food be thy medicine” quote literally. This document describes high-level architecture of Farmacy Food’s cloud-based IT infrastructure. It provides essential context followed by a list of key requirements, captured during interviews and research on the startup. A high-level architecture that satisfies those requirements is presented. The document concludes with a project breakdown in delivery milestones. This is a living document and is constantly being updated to reflect changes as the architecture evolves.

II. Vision

We are a community casual restaurant that takes the “let food be thy medicine” quote literally. We want to make healthy meals affordable and accessible that support customers’ health and well-being.

III. Goals and Opportunities

Immediate Business Goals

- Make food that is delicious, healthy, nutritionally dense
- Scale operations to enable dozens of automated fridges. Handle 1000s of customers.
- Expand to apartment buildings, college campuses, hospitals, and within businesses
- Scale from City of Detroit to other geographies (Michigan, Ohio, Illinois and nationwide)
- Retain profitability and margins
- Branding; Secure, Affordable and Delivers
- Don’t make any health claims and focus on dietary needs instead
- Donate and support health care workers, senior citizens, and so on and so forth. Support partnership non-profit organizations. We shifted to basically just creating free meals for frontline workers, low-income folks, and folks that were in care in Detroit
- Engage customers with health education and marketing

Architecture Narrative Checklist	
✓	What is the business problem?
✓	Why is the business doing this?
☐	Who are the major players (stakeholders)?
☐	What are the business and technical constraints?
☐	What architecture characteristics must be supported?
☐	What are the unique challenges of this business problem?
☐	What are the alternatives and corresponding tradeoffs?
☐	What is the proposed architectural solution?
☐	What artifacts are used to describe the architecture?
☐	What are the major architectural decisions?
☐	What are the risks associated with the architecture?

https://github.com/lookfwd/archkata

Jiakaturi

Act 1

- Overview
- Vision
- Goals and Opportunities
- Use Cases
- Architecture Characteristics
- Design Constraints
- High-Level Architecture
- Mid-Level Architecture
- Milestones
- ADRs

Architecture Narrative Checklist	
✓	What is the business problem?
✓	Why is the business doing this?
☐	Who are the major players (stakeholders)?
☐	What are the business and technical constraints?
☐	What architecture characteristics must be supported?
☐	What are the unique challenges of this business problem?
☐	What are the alternatives and corresponding tradeoffs?
☐	What is the proposed architectural solution?
☐	What artifacts are used to describe the architecture?
☐	What are the major architectural decisions?
☐	What are the risks associated with the architecture?

https://github.com/lookfwd/archkata

Jiakaturi

Act 1

- Overview
- Vision
- **Goals and Opportunities**
- Use Cases
- Architecture Characteristics
- Design Constraints
- High-Level Architecture
- Mid-Level Architecture
- Milestones
- ADRs

Business Opportunity

- Large market: 18 to 65, young professionals, often dealing with specific health challenges such as high blood pressure, diabetes, et cetera
- Immediately profitable: Most customers pay 12 bucks. It costs an average of $4.25 to produce, package and distribute our meals
- Low setup costs: Able to open up a location for an average of $7,000 comparison to the $500,000 to a million dollars for a restaurant
- Currently 400 meals a week. Between 1,500 to 2,000 meals a week by December
- Six to eight locations before the end of 2020
- 1,000 dedicated customers in early 2021. 10% of them subscription customers. The average meal subscription customer buys 10 meals a week (two meals per weekday)

Maintenance and Failure Operations

- Emergency Stock. The *stock controller* must ensure there's always enough safety stock to support the *kitchen*.
- Clean the smart fridges. Leftover orders must be returned by the *smart fridge* operator (e.g. coffee shop). The *delivery* must be notified via a web/mobile form to pick those up
- Accidents e.g. spills on the fridge must be reported by the *smart fridge* operator via a web/mobile form. Power outages must be monitored and when prolonged, the food/orders should be considered expired.
- The *customer* might not receive an order. In case of home delivery, this might happen because the *delivery* was late or unavailable or any other reason. The *customer* must be able to dispute the order and get a refund. In case of *smart fridge* delivery this might happen either due to *delivery* problems (same as before) or because the fridge is jammed, non-operational, there's a power outage or the operator is unavailable (e.g. store is closed), Aggregate stats must be collected in both cases and reported to the *management*.
- The *customer* might complain about the quality of the order such as meal quality, temperature, quality of cooking, delivery etc. The *management* is required to follow up in those cases with the goal to retain the customer and improve the process.
- The *kitchen* might run out of stock or lag behind or be unable to deliver on time for any other reasons. Detailed metrics must be collected and communicated to the *management* in real time. The *management* can decide to issue a refund or take other actions to mitigate the issue.
- The *stock controller* might be unable to purchase ingredients on a price that gives acceptable margin. It must notify the *management* in order to monitor the situation and take actions such as mark meals as unavailable or provide alternative ingredients.

Architecture Narrative Checklist	
✓	What is the business problem?
✓	Why is the business doing this?
☐	Who are the major players (stakeholders)?
✓	What are the business and technical constraints?
☐	What architecture characteristics must be supported?
✓	What are the unique challenges of this business problem?
☐	What are the alternatives and corresponding tradeoffs?
☐	What is the proposed architectural solution?
☐	What artifacts are used to describe the architecture?
☐	What are the major architectural decisions?
☐	What are the risks associated with the architecture?

https://github.com/lookfwd/archkata

Jiakaturi

Act 1

- Overview
- Vision
- **Goals and Opportunities**
- Use Cases
- Architecture Characteristics
- Design Constraints
- High-Level Architecture
- Mid-Level Architecture
- Milestones
- ADRs

Architecture Narrative Checklist	
✓	What is the business problem?
✓	Why is the business doing this?
☐	Who are the major players (stakeholders)?
✓	What are the business and technical constraints?
☐	What architecture characteristics must be supported?
✓	What are the unique challenges of this business problem?
☐	What are the alternatives and corresponding tradeoffs?
☐	What is the proposed architectural solution?
☐	What artifacts are used to describe the architecture?
☐	What are the major architectural decisions?
☐	What are the risks associated with the architecture?

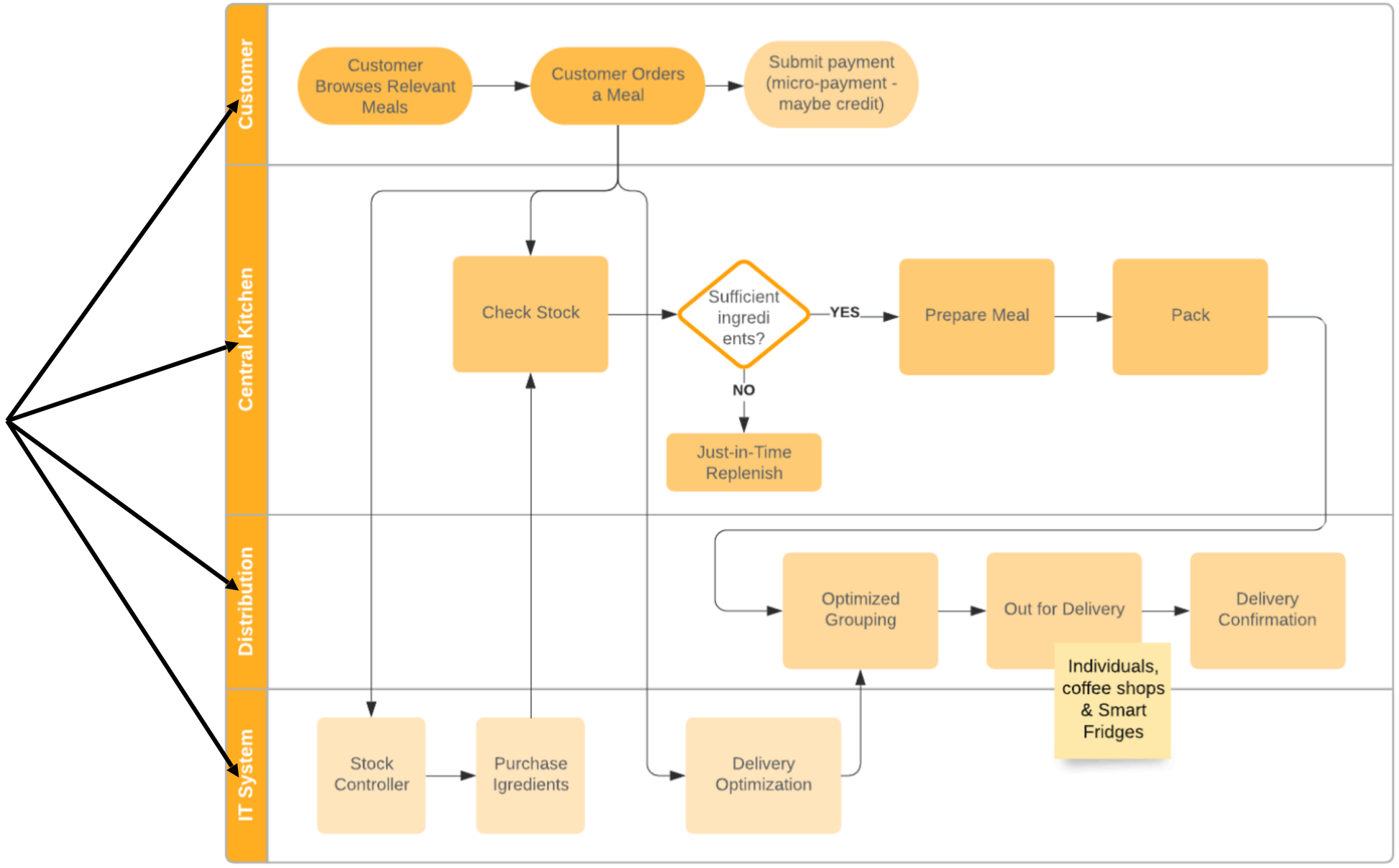
https://github.com/lookfwd/archkata

Jiakaturi

Act 1

- Overview
- Vision
- Goals and Opportunities
- Use Cases
- Architecture Characteristics
- Design Constraints
- High-Level Architecture
- Mid-Level Architecture
- Milestones
- ADRs

IV. Use Cases


Customer
Central Kitchen
Distribution
IT System
Customer Browses Relevant Meals
Customer Orders a Meal
Submit payment (micro-payment - maybe credit)
Check Stock
Sufficient ingredi ents?
YES
NO
Prepare Meal
Pack
Just-in-Time Replenish
Optimized Grouping
Out for Delivery
Delivery Confirmation
Individuals, coffee shops & Smart Fridges
Stock Controller
Purchase Igredients
Delivery Optimization

Architecture Narrative Checklist	
✓	What is the business problem?
✓	Why is the business doing this?
✓	Who are the major players (stakeholders)?
✓	What are the business and technical constraints?
☐	What architecture characteristics must be supported?
✓	What are the unique challenges of this business problem?
☐	What are the alternatives and corresponding tradeoffs?
☐	What is the proposed architectural solution?
☐	What artifacts are used to describe the architecture?
☐	What are the major architectural decisions?
☐	What are the risks associated with the architecture?

https://github.com/lookfwd/archkata

Jiakaturi

Act 1

- Overview
- Vision
- Goals and Opportunities
- Use Cases
- Architecture Characteristics
- Design Constraints
- High-Level Architecture
- Mid-Level Architecture
- Milestones
- ADRs

Architecture Narrative Checklist	
✓	What is the business problem?
✓	Why is the business doing this?
✓	Who are the major players (stakeholders)?
✓	What are the business and technical constraints?
☐	What architecture characteristics must be supported?
✓	What are the unique challenges of this business problem?
☐	What are the alternatives and corresponding tradeoffs?
☐	What is the proposed architectural solution?
☐	What artifacts are used to describe the architecture?
☐	What are the major architectural decisions?
☐	What are the risks associated with the architecture?

https://github.com/lookfwd/archkata

Jiakaturi

Prioritized Architecture Characteristics

1. Enable Discovery - Agility

The customer experience in acquisition channels (mobile, web and even SMS) must be

2. Affordable DevSecOps - Viability

The startup must be able to implement the architecture given budget and time constraints.

3. Easy to Pivot - Flexibility

The design must be able to grow as the startup grows. It's expected that Farmacy Food will

4. Availability

This is a business critical system and this reflects on the Service-Level Objectives (SLOs).

5. Security

The design must be secure to protect the brand. To have both high security and low costs

Architecture Narrative Checklist	
✓	What is the business problem?
✓	Why is the business doing this?
✓	Who are the major players (stakeholders)?
✓	What are the business and technical constraints?
✓	What architecture characteristics must be supported?
✓	What are the unique challenges of this business problem?
☐	What are the alternatives and corresponding tradeoffs?
☐	What is the proposed architectural solution?
☐	What artifacts are used to describe the architecture?
☐	What are the major architectural decisions?
☐	What are the risks associated with the architecture?

https://github.com/lookfwd/archkata

Jiakaturi

Architecture Narrative Checklist	
✓	What is the business problem?
✓	Why is the business doing this?
✓	Who are the major players (stakeholders)?
✓	What are the business and technical constraints?
✓	What architecture characteristics must be supported?
✓	What are the unique challenges of this business problem?
☐	What are the alternatives and corresponding tradeoffs?
☐	What is the proposed architectural solution?
☐	What artifacts are used to describe the architecture?
☐	What are the major architectural decisions?
☐	What are the risks associated with the architecture?

https://github.com/lookfwd/archkata

Jiakaturi

Design Constraints

- ChefTec is used for Restaurant Inventory Management & Purchasing
- Toast is used as Point of Sales solution
- Intuit QuickBooks is used for accounting
- There's an existing mobile application

ChefTec doesn't have any public available information about their API. Given that most of their off-the-shelf solutions cost in the order of serveral thousand dollars, its consultancy services is unlikely to be cheap. This is a significant risk given how important the central kitchen is for Farmacy Food. ChefTec integrates with QuickBooks for invoicing through add-ons for extra cost.

Toast has an API that one can access after accepting their "API Documentation License Agreement", here.

QuickBooks has an elaborate API and SDKs.

Architecture Narrative Checklist	
✓	What is the business problem?
✓	Why is the business doing this?
✓	Who are the major players (stakeholders)?
✓	What are the business and technical constraints?
✓	What architecture characteristics must be supported?
✓	What are the unique challenges of this business problem?
☐	What are the alternatives and corresponding tradeoffs?
☐	What is the proposed architectural solution?
☐	What artifacts are used to describe the architecture?
☐	What are the major architectural decisions?
☐	What are the risks associated with the architecture?

https://github.com/lookfwd/archkata

Jiakaturi

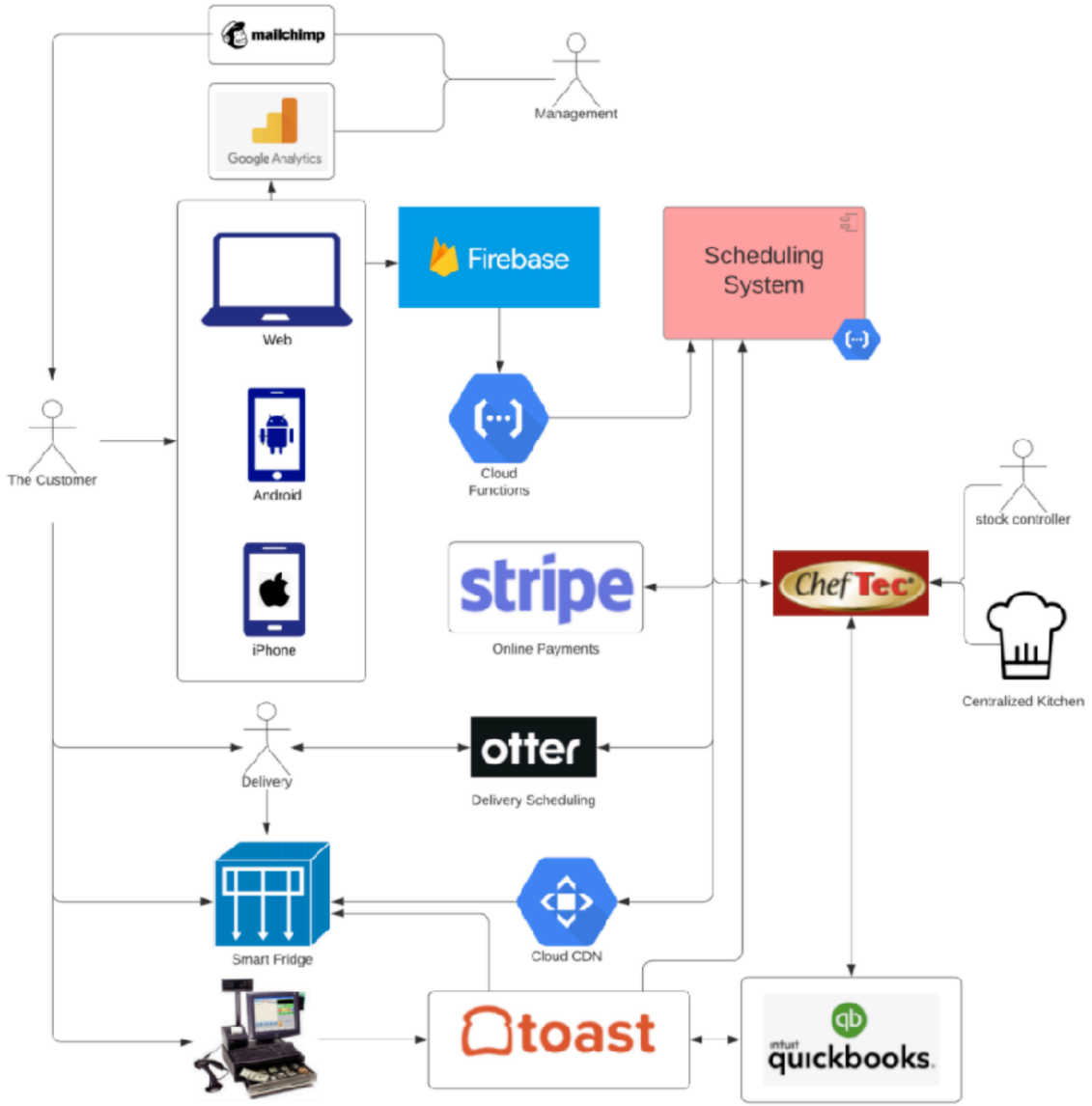
Architecture Narrative Checklist	
✓	What is the business problem?
✓	Why is the business doing this?
✓	Who are the major players (stakeholders)?
✓	What are the business and technical constraints?
✓	What architecture characteristics must be supported?
✓	What are the unique challenges of this business problem?
☐	What are the alternatives and corresponding tradeoffs?
☐	What is the proposed architectural solution?
☐	What artifacts are used to describe the architecture?
☐	What are the major architectural decisions?
☐	What are the risks associated with the architecture?

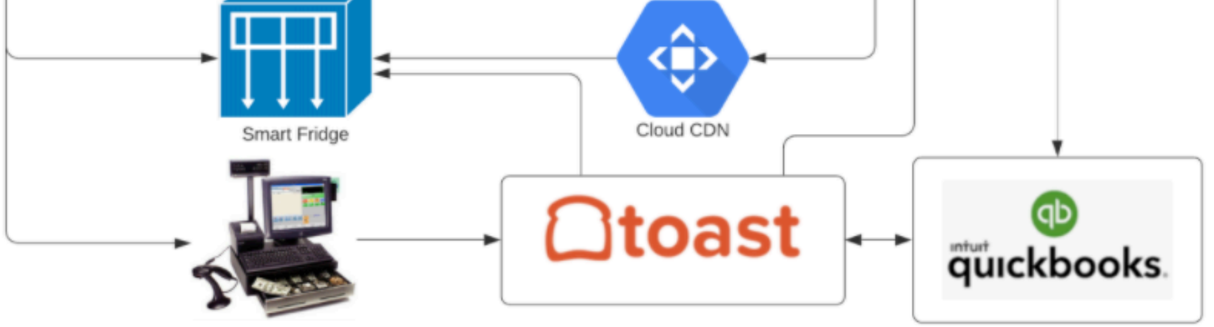
https://github.com/lookfwd/archkata

Jiakaturi

VI. High-Level Architecture

Directly out of the basic listing of the requirements, and by taking the ADRs into account, this high-level diagram can provide all the necessary use cases.





This is a customer-centric architecture. The customer interacts with Farmacy Food through well designed touch-points. Those are the:

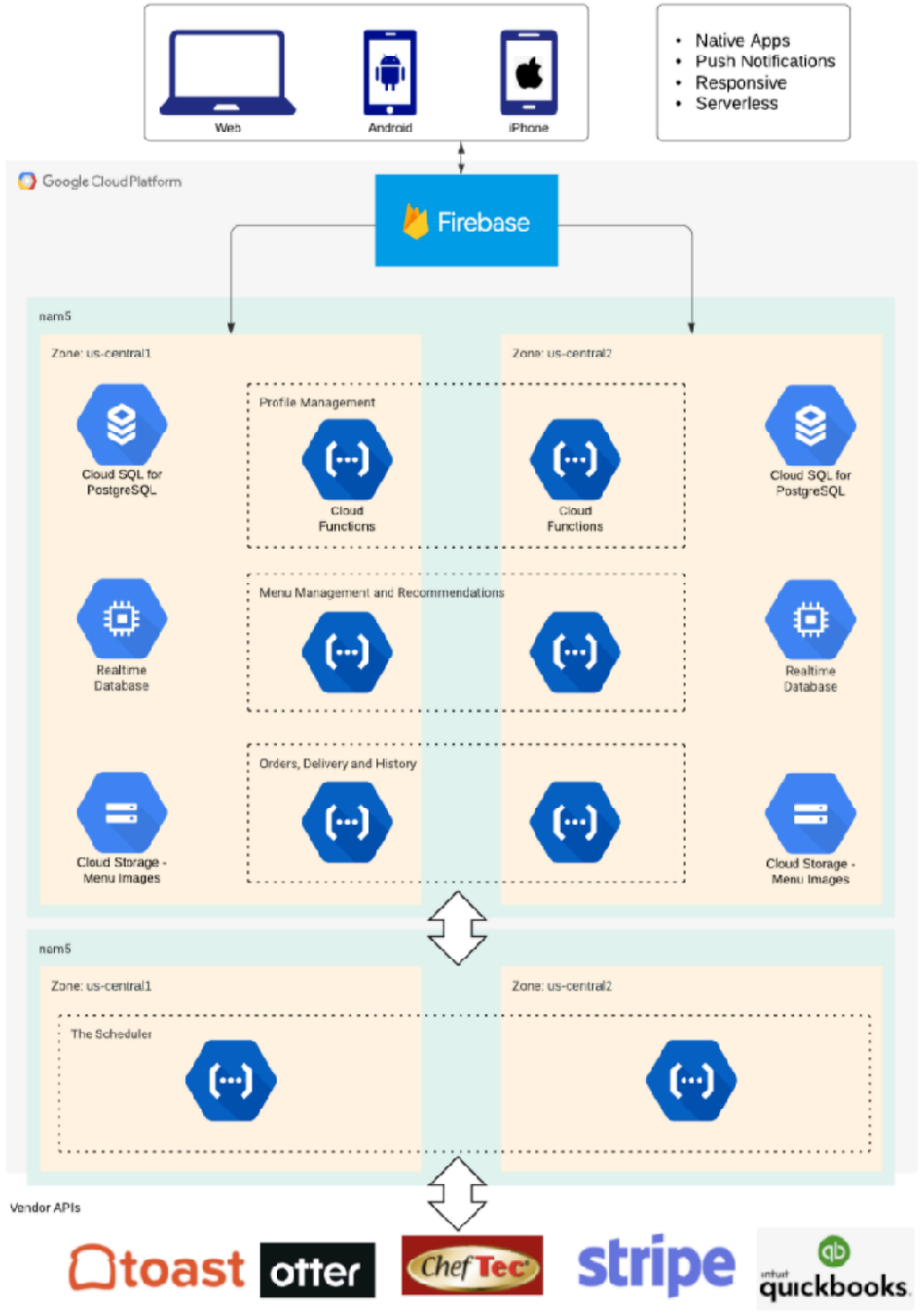
- Cross-platform web and mobile apps
- Smart Fridges also on stores
- Point of Sales Devices on stores
- Web Campaigns that educate and engage

The kitchen is already familiar and satisfied with their ChefTec solution. As long as an API integration solution is found, ChefTec can integrate with the rest of the system. Their backup mechanism can be another way to export data. If these don't work, import/export and manual operations will be required. As the startup scales, it's likely that this will become a friction point that will make Farmacy Food investigate ChefTec alternatives.

Management oversees operations with emphasis on Marketing and Analytics. This is to be expected for a startup in growth mode. On the other hand, back-end operations are automated to a great extent and don't require management involvement.

QuickBooks takes care of accounting and payroll. It is integrated with ChefTec and the Toast PoS. The online payments must also be integrated with QuickBooks by using the relevant app.

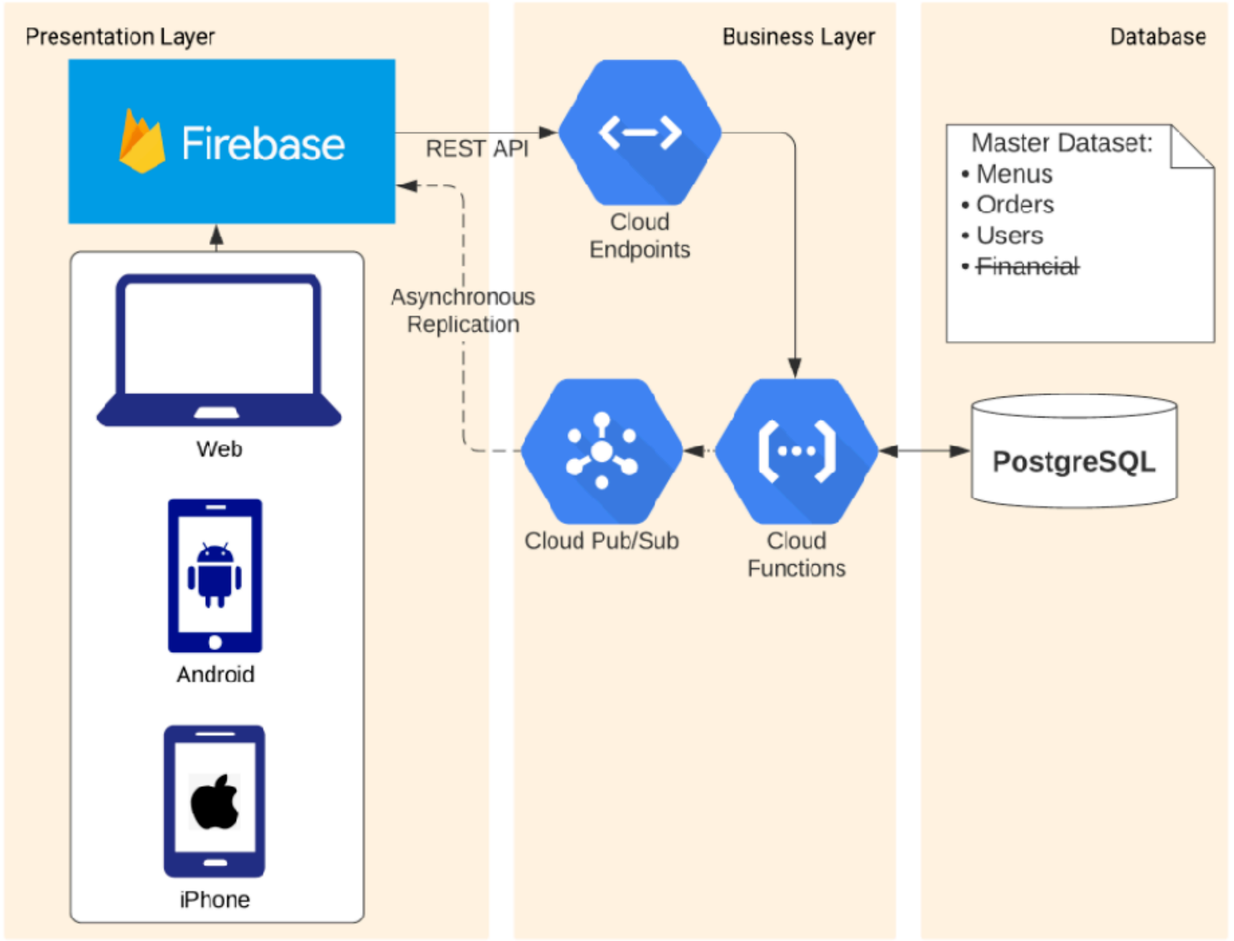
This Architecture uses SaaS vendors for most of the mundane aspects of the system. This allows the development team to focus on developing the differentiating factors of Farmacy Food. Those are:


Web
Android
iPhone
Native Apps
Push Notifications
Responsive
Serverless
Google Cloud Platform
Firebase
nam5
Zone: us-central1
Zone: us-central2
Profile Management
Cloud SQL for PostgreSQL
Cloud Functions
Cloud Functions
Cloud SQL for PostgreSQL
Menu Management and Recommendations
Realtime Database
Realtime Database
Orders, Delivery and History
Cloud Storage - Menu Images
Cloud Storage - Menu Images
nam5
Zone: us-central1
Zone: us-central2
The Scheduler
Vendor APIs
toast
otter
ChefTec
stripe
quickbooks

By using React Native and Firebase, we avoid spending development time on mundane tasks like login and cross-platform mobile development. The development team can focus on the development of value-adding online functionality and seamless UX.

How we Avoid Firebase Lock-in

The main reported drawback of Google Firebase is vendor Lock-In. More specifically it's reported that there are limitted ways to export data and few options that allow you to export data. This has been partially addressed but it's better to be prepared. More specifically we use Firebase as presentation layer. Its databases are used as a form of caching, but the master dataset lives outside firebase, in a managed PostgreSQL. To enable an extra level of decoupling, we use an Event Bus implemented using Cloud Pub/Sub. Modifications to the dataset are also replicated in the Event Bus and Firebase uses those events to update its internal databases and keep in sync. It also uses those updates to push notifications on the Web/App.



Architecture Narrative Checklist	
✓	What is the business problem?
✓	Why is the business doing this?
✓	Who are the major players (stakeholders)?
✓	What are the business and technical constraints?
✓	What architecture characteristics must be supported?
✓	What are the unique challenges of this business problem?
☐	What are the alternatives and corresponding tradeoffs?
✓	What is the proposed architectural solution?
✓	What artifacts are used to describe the architecture?
☐	What are the major architectural decisions?
☐	What are the risks associated with the architecture?

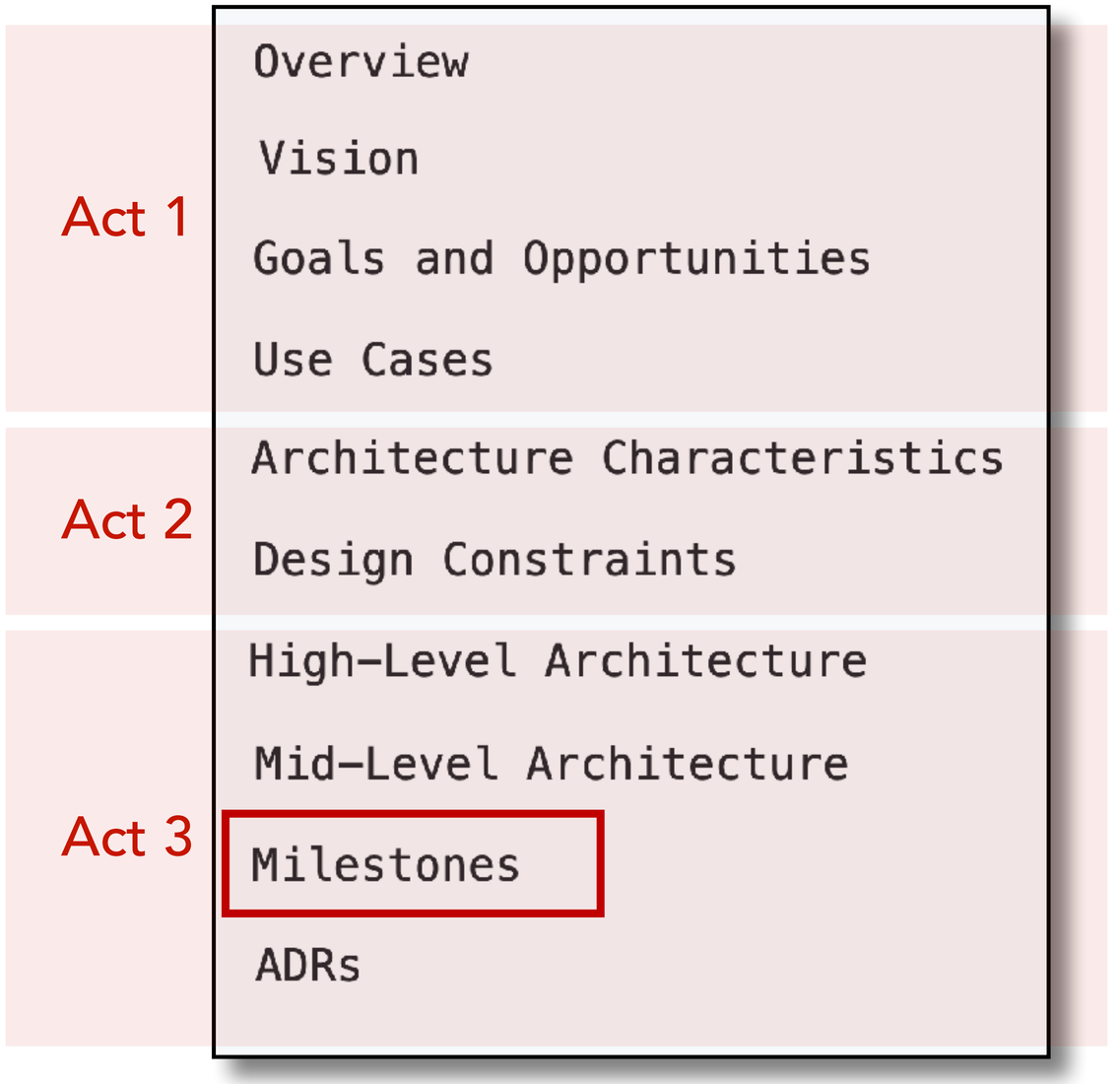
https://github.com/lookfwd/archkata

Jiakaturi

Architecture Narrative Checklist	
✓	What is the business problem?
✓	Why is the business doing this?
✓	Who are the major players (stakeholders)?
✓	What are the business and technical constraints?
✓	What architecture characteristics must be supported?
✓	What are the unique challenges of this business problem?
☐	What are the alternatives and corresponding tradeoffs?
✓	What is the proposed architectural solution?
✓	What artifacts are used to describe the architecture?
☐	What are the major architectural decisions?
☐	What are the risks associated with the architecture?

https://github.com/lookfwd/archkata

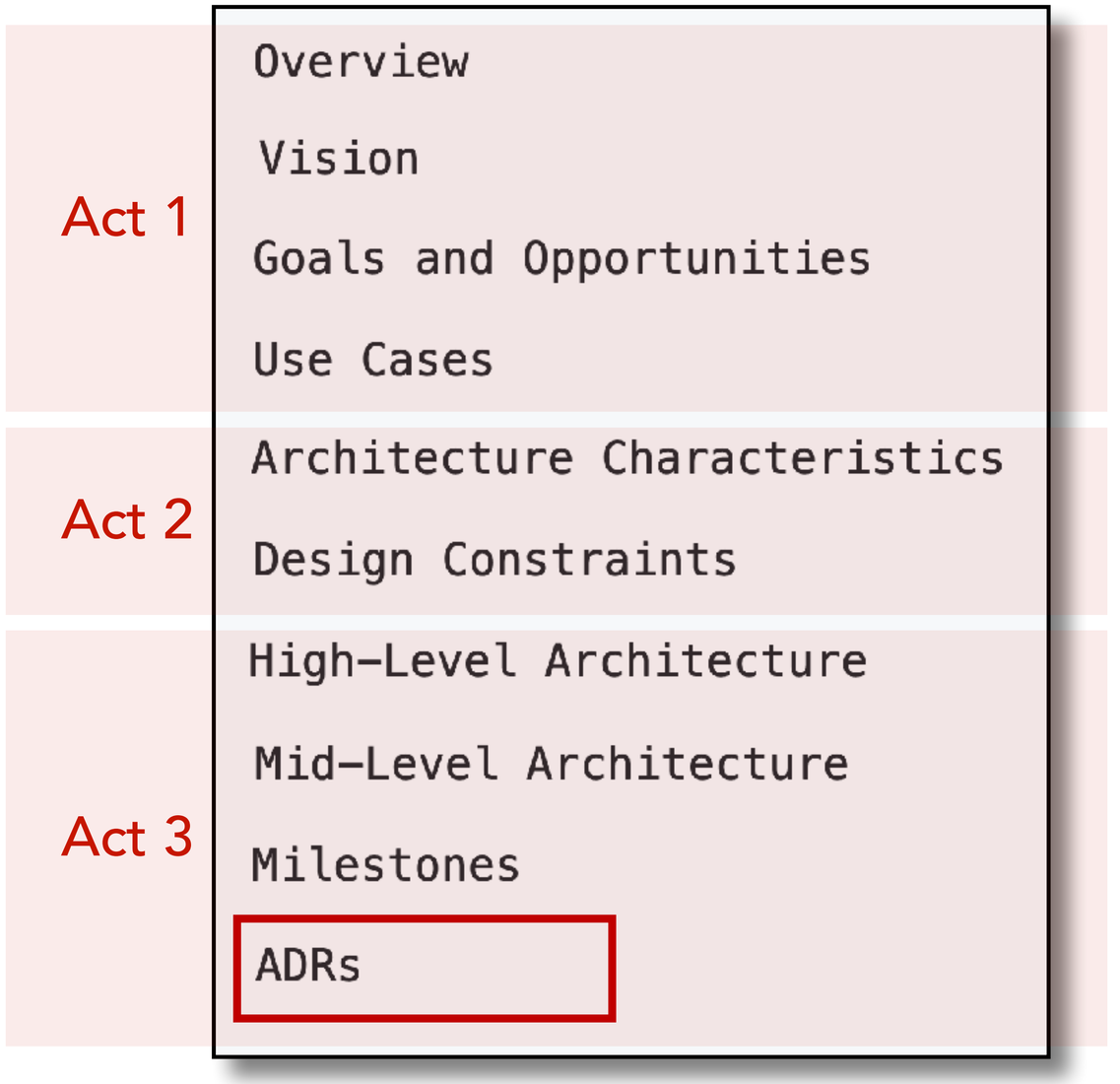
Jiakaturi



Architecture Narrative Checklist	
✓	What is the business problem?
✓	Why is the business doing this?
✓	Who are the major players (stakeholders)?
✓	What are the business and technical constraints?
✓	What architecture characteristics must be supported?
✓	What are the unique challenges of this business problem?
☐	What are the alternatives and corresponding tradeoffs?
✓	What is the proposed architectural solution?
✓	What artifacts are used to describe the architecture?
☐	What are the major architectural decisions?
☐	What are the risks associated with the architecture?

https://github.com/lookfwd/archkata

Jiakaturi



IX. ADRs

ADR 1. Hosting Platform

Status: proposed Context: The architecture needs to be hosted either on dedicated servers or in the cloud. The choice of cloud provider affects the cost of the solution and the time of the implementation. Decision: The platform of choice is the Google Cloud Platform (GCP) on the basis of ease of development, security, versatility and price. To avoid vendor lock-in, no components that don't have an AWS-equivalent should be used without explicit permission. Consequences: The implementation is expected to scale elastically on demand. We can run experiments and develop staging and test environments easily. AWS remains a migration option if factors like host dictate it at some point. Notably, Google App Engine and BigQuery can *not* be used because of vendor lock-in concerns.

ADR 2. Mobile App Platform

- Status: Proposed
- Context: The existing mobile application will need to be revamped to enable further use cases.
- Decision: Google Firebase will be used to support. It's a more mature framework compared to the alternative AWS Amplify. React Native will be used for cross-platform app development. This enables us to have one app instead of one for each platform. React Native is a mature technology developed by Facebook.
- Consequences: The contractors that will be developing the app should be experts on those technologies. Given that they're the two most widely supported, through competitive bidding, this is expected to reduce development and maintenance costs.

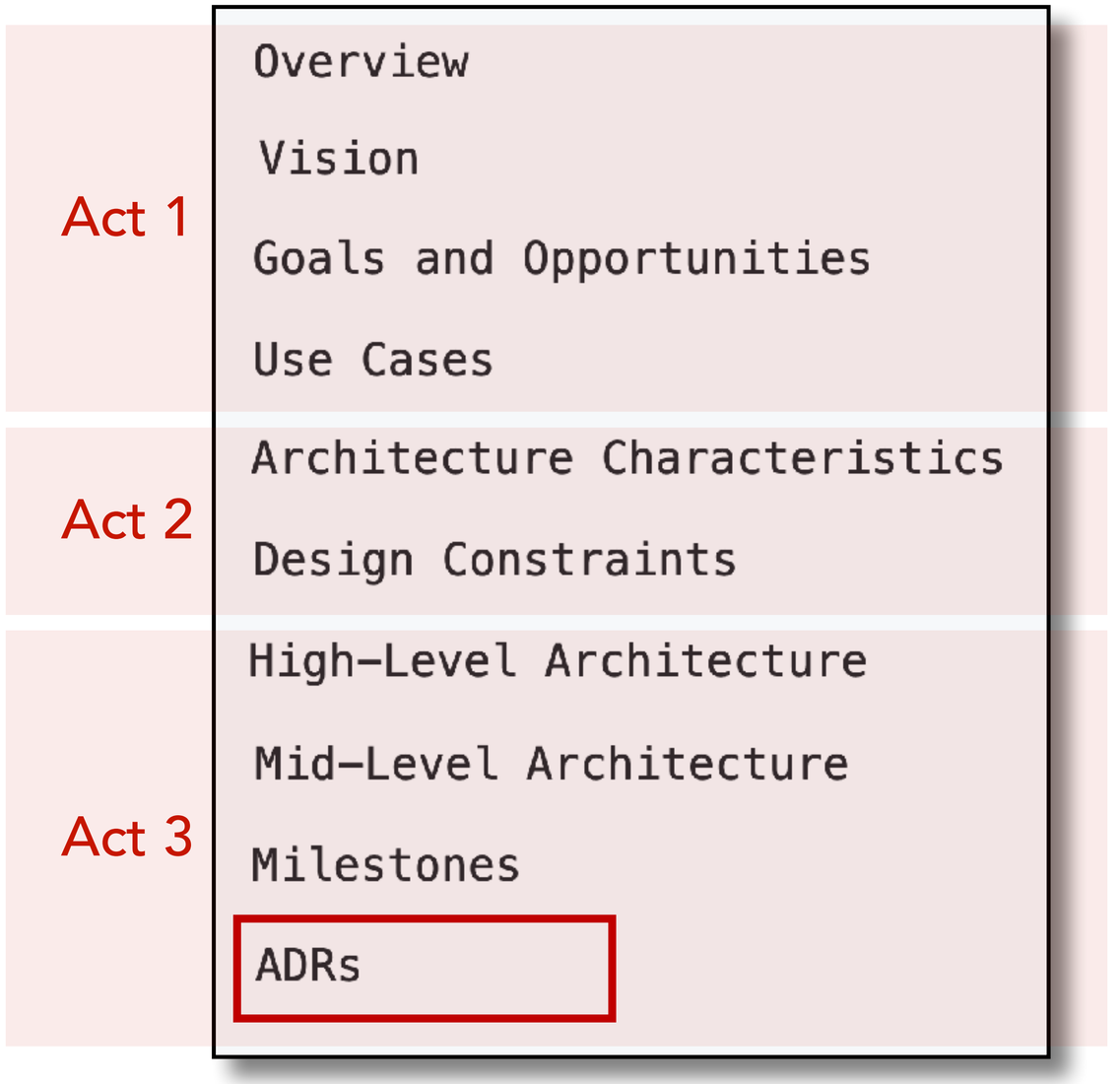
ADR 3. Serverless

- Status: proposed
- Context: Since traffic isn't an issue, we can reduce hosting costs by running software in Virtual Machines in GCP (Compute Engine). This means that we will be responsible for building machines and operating them. We can install explicitly RabbitMQ/Kafka, Databases like PostgreSQL and business logic as well as front-end servers (nginx, custom code). On the other hand we could use serverless technologies.
- Decision: Building virtual machines and self-hosting means that the developers will have to spend time maintaining and operating machines. That cost is significant and will slow down development time, since soon a fraction of developer's time will need to be spent in KTLO. This isn't a good investment of developers' time

Architecture Narrative Checklist	
✓	What is the business problem?
✓	Why is the business doing this?
✓	Who are the major players (stakeholders)?
✓	What are the business and technical constraints?
✓	What architecture characteristics must be supported?
✓	What are the unique challenges of this business problem?
☐	What are the alternatives and corresponding tradeoffs?
✓	What is the proposed architectural solution?
✓	What artifacts are used to describe the architecture?
✓	What are the major architectural decisions?
☐	What are the risks associated with the architecture?

https://github.com/lookfwd/archkata

Jiakaturi



Architecture Narrative Checklist	
✓	What is the business problem?
✓	Why is the business doing this?
✓	Who are the major players (stakeholders)?
✓	What are the business and technical constraints?
✓	What architecture characteristics must be supported?
✓	What are the unique challenges of this business problem?
☐	What are the alternatives and corresponding tradeoffs?
✓	What is the proposed architectural solution?
✓	What artifacts are used to describe the architecture?
✓	What are the major architectural decisions?
☐	What are the risks associated with the architecture?

https://github.com/lookfwd/archkata

Jiakaturi

Act 1	Overview Vision Goals and Opportunities Use Cases
Act 2	Architecture Characteristics Design Constraints
Act 3	High-Level Architecture Mid-Level Architecture Milestones ADRs

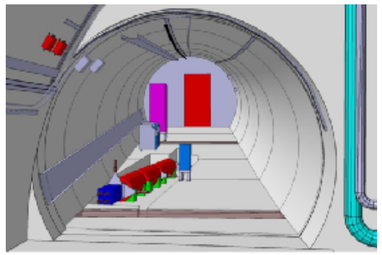
architecture narrative

YOU BE THE JUDGE!

Which of these architecture solution narratives is more effective and why?

Myagi's Little Forests

https://github.com/miyagis-forests/farmacy-food-kata



Arcolider

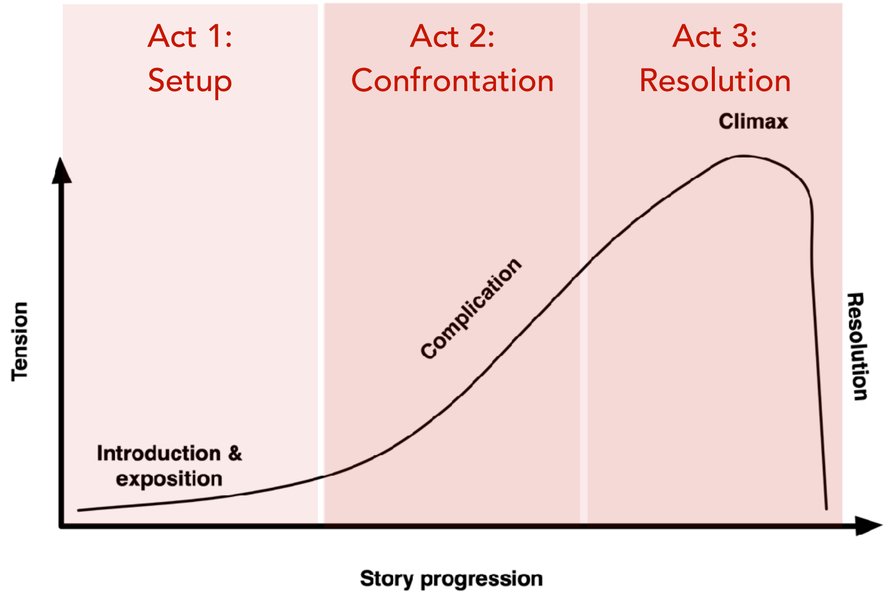
https://github.com/ldynia/archcolider

Poll question:

Based on your analysis, which architecture narrative did you find most effective for describing the architecture solution?

Myagi's Little Forests

Arcolider



https://github.com/miyagis-forests/farmacy-food-kata

Act 1

Requirements

This section contains the requirements, distilled from the provi
the interview with the PO, Kwaku Osei, but also with some assu
main drivers for the design decisions in this proposal.

- Functional requirements
- Quality attribute requirements, aka architecture characteris

Act 2

Architecture

Here you find the documentation of the software architecture th
requirements.

As a starting point, there's a context diagram that gives an over
we called the *Farmacy Food System*, which is the scope of this

ADRs

The linked ADRs below record the main architecture decisions re
context and rationale.

- ADR 001 - Microservice style
- ADR 002 - Payment gateway

Architecture Narrative Checklist
☐ What is the business problem?
☐ Why is the business doing this?
☐ Who are the major players (stakeholders)?
☐ What are the business and technical constraints?
☐ What architecture characteristics must be supported?
☐ What are the unique challenges of this business problem?
☐ What are the alternatives and corresponding tradeoffs?
☐ What is the proposed architectural solution?
☐ What artifacts are used to describe the architecture?
☐ What are the major architectural decisions?
☐ What are the risks associated with the architecture?

https://github.com/miyagis-forests/farmacy-food-kata

Requirements

This section contains the requirements, distilled from the provi
the interview with the PO, Kwaku Osei, but also with some assu
main drivers for the design decisions in this proposal.

- Functional requirements
- Quality attribute requirements, aka architecture characteris

Architecture

Here you find the documentation of the software architecture th
requirements.

As a starting point, there's a context diagram that gives an over
we called the *Farmacy Food System*, which is the scope of this

ADRs

The linked ADRs below record the main architecture decisions re
context and rationale.

- ADR 001 - Microservice style
- ADR 002 - Payment gateway

Architecture Narrative Checklist	
✓	What is the business problem?
	Why is the business doing this?
✓	Who are the major players (stakeholders)?
	What are the business and technical constraints?
✓	What architecture characteristics must be supported?
	What are the unique challenges of this business problem?
	What are the alternatives and corresponding tradeoffs?
✓	What is the proposed architectural solution?
✓	What artifacts are used to describe the architecture?
✓	What are the major architectural decisions?
	What are the risks associated with the architecture?

https://github.com/miyagis-forests/farmacy-food-kata

Requirements

This section contains the requirements, distilled from the provi
the interview with the PO, Kwaku Osei, but also with some assu
main drivers for the design decisions in this proposal.

- Functional requirements
- Quality attribute requirements, aka architecture characteris

Architecture

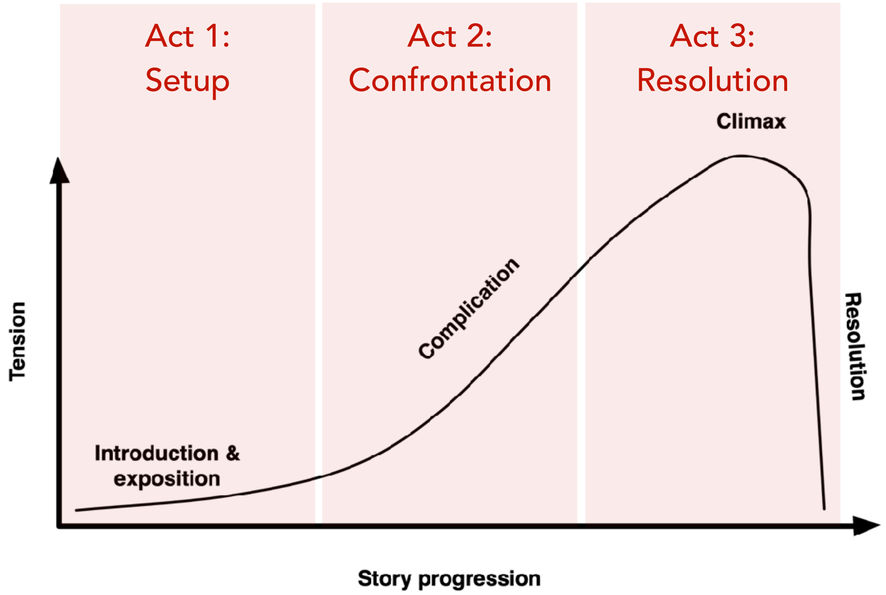
Here you find the documentation of the software architecture th
requirements.

As a starting point, there's a context diagram that gives an over
we called the *Farmacy Food System*, which is the scope of this

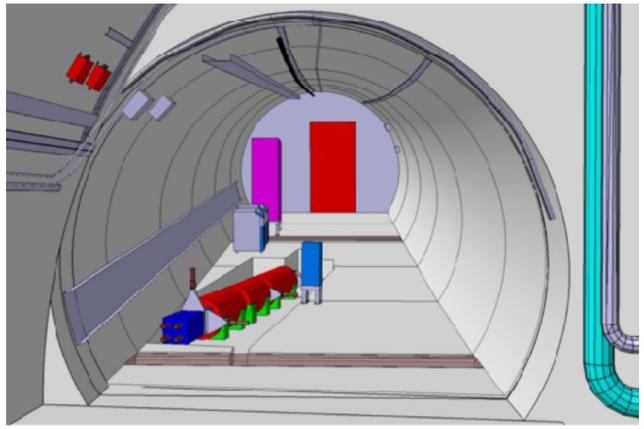
ADRs

The linked ADRs below record the main architecture decisions re
context and rationale.

- ADR 001 - Microservice style
- ADR 002 - Payment gateway



https://github.com/ldynia/archcolider



Act 1

- Problem Background
  - System overview
  - Goals and Context
  - Constraints
  - Stakeholder
  - Significant Driving Requirements

Act 2

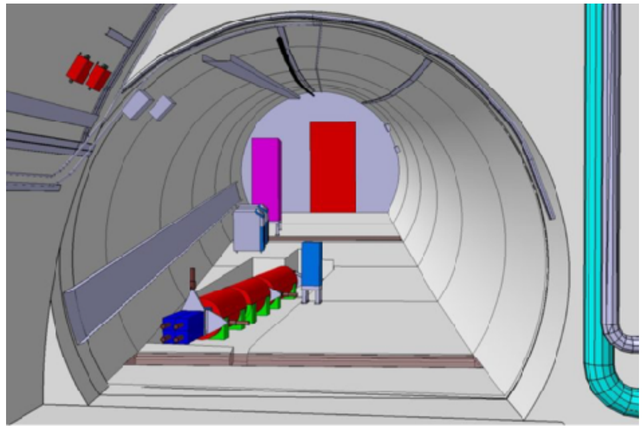
- Solution Background
  - Solution Overview
  - Assumptions
  - Approach Summary
  - Tradeoffs
  - Risks and Sensitive points
  - ADRs

Act 3

- Views and Perspectives
  - User Scenarios
  - Informational
  - Concurrency
  - Deployment
  - Cost Analysis
  - Security

Architecture Narrative Checklist
☐ What is the business problem?
☐ Why is the business doing this?
☐ Who are the major players (stakeholders)?
☐ What are the business and technical constraints?
☐ What architecture characteristics must be supported?
☐ What are the unique challenges of this business problem?
☐ What are the alternatives and corresponding tradeoffs?
☐ What is the proposed architectural solution?
☐ What artifacts are used to describe the architecture?
☐ What are the major architectural decisions?
☐ What are the risks associated with the architecture?

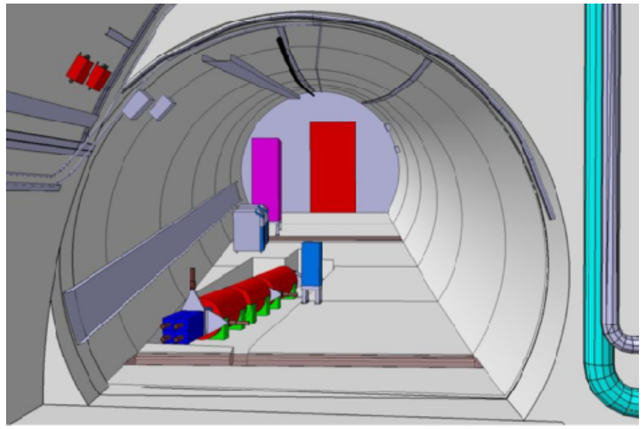
https://github.com/ldynia/archcolider



- Problem Background
  - System overview
  - Goals and Context
  - Constraints
  - Stakeholder
  - Significant Driving Requirements
- Solution Background
  - Solution Overview
  - Assumptions
  - Approach Summary
  - Tradeoffs
  - Risks and Sensitive points
  - ADRs
- Views and Perspectives
  - User Scenarios
  - Informational
  - Concurrency
  - Deployment
  - Cost Analysis
  - Security

Architecture Narrative Checklist
✓ What is the business problem?
✓ Why is the business doing this?
✓ Who are the major players (stakeholders)?
✓ What are the business and technical constraints?
✓ What architecture characteristics must be supported?
✓ What are the unique challenges of this business problem?
✓ What are the alternatives and corresponding tradeoffs?
✓ What is the proposed architectural solution?
✓ What artifacts are used to describe the architecture?
✓ What are the major architectural decisions?
✓ What are the risks associated with the architecture?

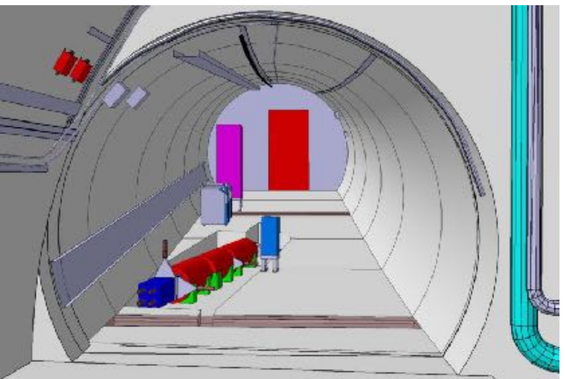
https://github.com/ldynia/archcolider



- Problem Background
  - System overview
  - Goals and Context
  - Constraints
  - Stakeholder
  - Significant Driving Requirements
- Solution Background
  - Solution Overview
  - Assumptions
  - Approach Summary
  - Tradeoffs
  - Risks and Sensitive points
  - ADRs
- Views and Perspectives
  - User Scenarios
  - Informational
  - Concurrency
  - Deployment
  - Cost Analysis
  - Security

Organization + Completeness

Ubiquitous language

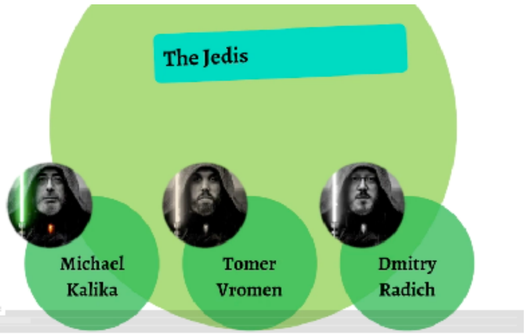


Solution Structure

Table of content:

evolutionary architecture

Awareness of the domain's likely evolution tied to the architecture plan.



The Architecture

- General Architecture - the general architectural idea.

Stage 1 Capabilities

- Fridge Capability
- Card and Payment
- Identity and Profile
- Kitchen Capability
- Meal Inventory
- Pricing

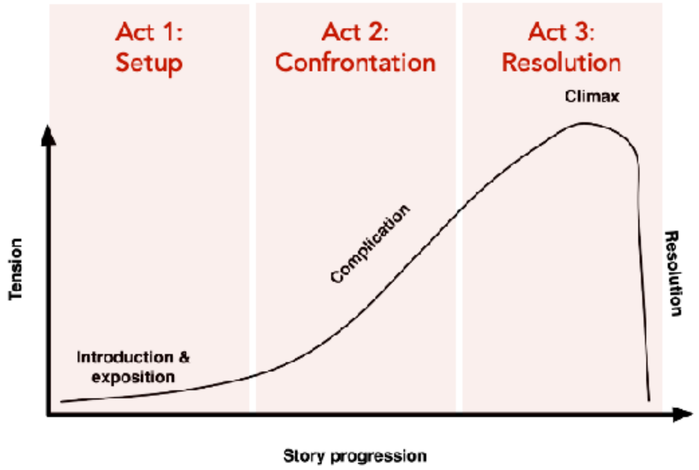
Stage 2 Capabilities

- Customer Subscriptions
- Notifications
- Feedback and Ranking
- Referrals and Rewards

Stage 3 Capabilities

- Data Platform
- Expert Platform and CMS

the bottom line…



Think about and focus on how you describe and present your architectural solution in the most effective way possible

Architecture Narrative Checklist
☐ What is the business problem?
☐ Why is the business doing this?
☐ Who are the major players (stakeholders)?
☐ What are the business and technical constraints?
☐ What architecture characteristics must be supported?
☐ What are the unique challenges of this business problem?
☐ What are the alternatives and corresponding tradeoffs?
☐ What is the proposed architectural solution?
☐ What artifacts are used to describe the architecture?
☐ What are the major architectural decisions?
☐ What are the risks associated with the architecture?

Driving Characteristics

Are these the characteristics the Farmacy Foods architecture must support?

Simplicity

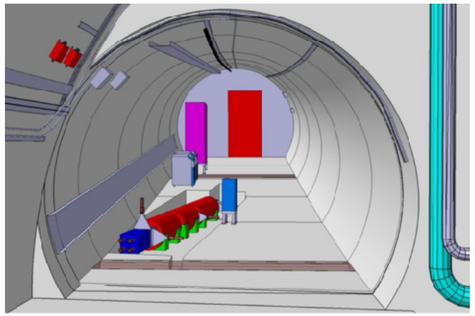
Modifiability

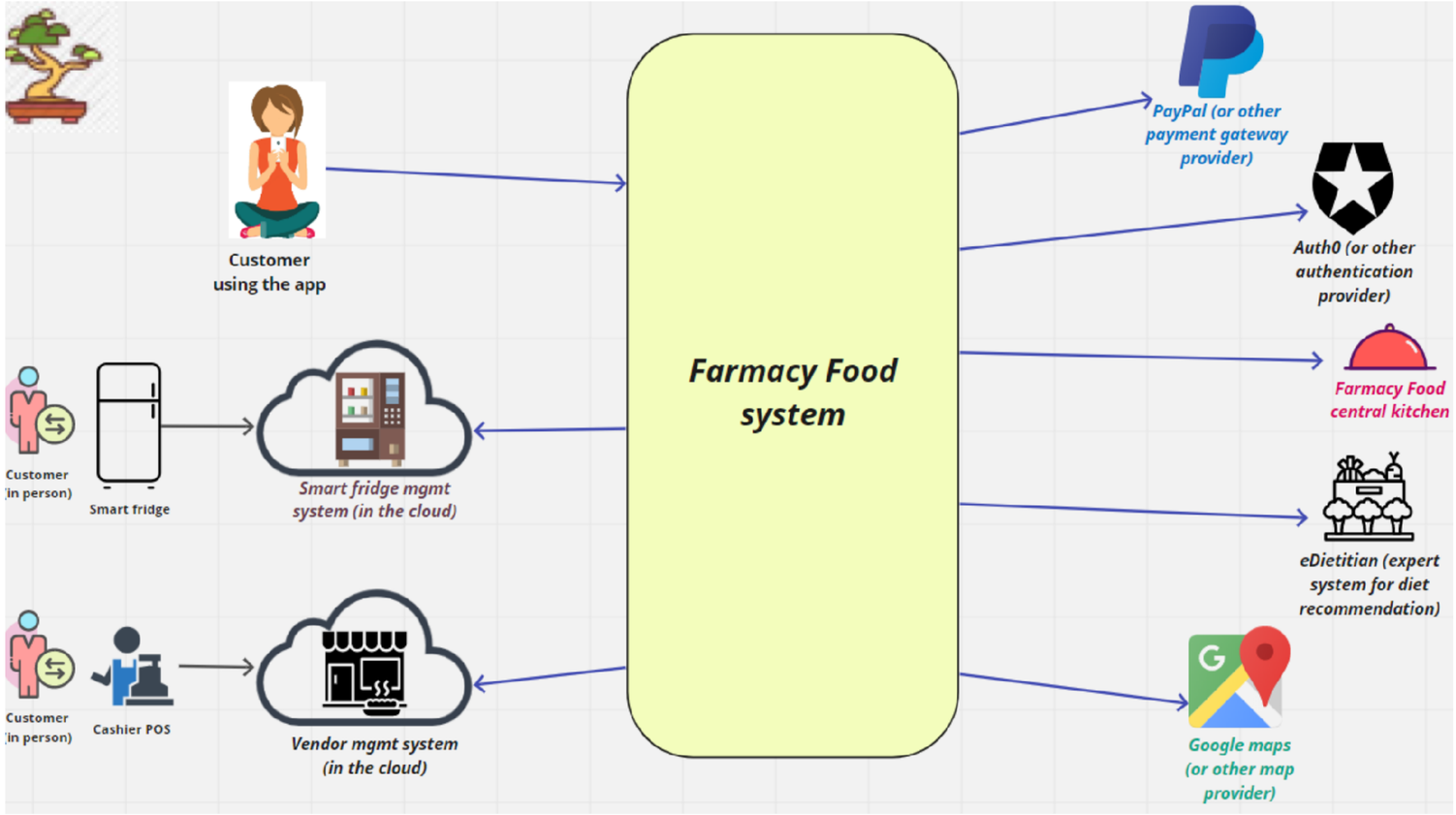
Availability

Reliability

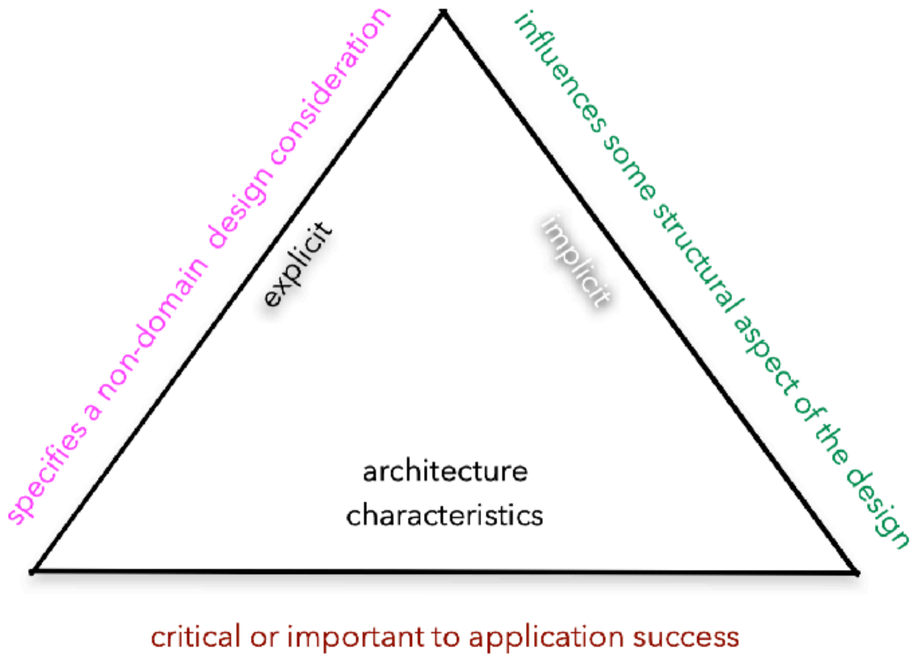
Data Integrity

Security





Architecture characteristics form the foundational aspects of the architecture and drive trade-off analysis and decision making



- [x] What is the scope of each characteristic?
- [x] Can you justify the identified characteristics?
- [x] Can you tie them back to business drivers and requirements?
- [x] Are all the characteristics you identified critical to the application success?

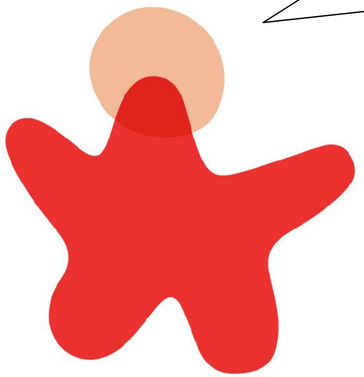
Architecture Characteristics Checklist	
☐	What is the scope of the identified characteristics?
☐	Can you justify the identified characteristics?
☐	Can you tie them back to business needs or requirements?
☐	Are all the characteristics critical or important to success?

architecture characteristics

accessibility
accountability
accuracy
adaptability
administrability
affordability
agility
auditability
autonomy
availability
compatibility
composability
configurability
correctness
credibility
customizability
debugability
degradability
determinability
demonstrability
dependability
deployability
discoverability
distributability
durability
effectiveness
efficiency
evolvability
extensibility
failure transparency
fault-tolerance
fidelity
flexibility
inspectability
installability
integrity
interchangeability
interoperability
learnability
maintainability
manageability
mobility
modifiability
modularity
operability
orthogonality
portability
precision
predictability
process capabilities
producibility
provability
recoverability
relevance
repeatability
reproducibility
resilience
responsiveness
reusability
robustness
safety
scalability
seamlessness
self-sustainability
serviceability
supportability
securability
simplicity
stability
standards compliance
survivability
sustainability
tailorability
testability
timeliness
traceability
transparency
ubiquity
understandability
upgradability
usability

architecture characteristics

"How do I know which architecture characteristics I need for this system?"



software architect

"We are building a new stock trading system"

"We must support 20 to 200,000 concurrent customers"

"We are aggressively expanding our business"

https://www.developertoarchitect.com/downloads/worksheets.html

https://www.developertoarchitect.com/lessons/lesson112.html

Architecture Characteristics Worksheet

System/Project: _____

Architect/Team: _____ **Date:** _____

Candidate Architecture Characteristics

performance	data integrity	deployability
responsiveness	data consistency	testability
availability	adaptability	abstraction
fault tolerance	extensibility	workflow
scalability	interoperability	configurability
elasticity	concurrency	recoverability

others: _____

_____

_____

a / b denotes characteristics that are related; some systems only need one of these, other systems may need both

Top 3 Driving Characteristics

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____

Implicit Characteristics

feasibility (cost/time)

security

maintainability

simplicity

Others Considered

_____

_____

_____

_____

Instructions

- Identify no more than 7 driving characteristics.
- Pick the top 3 characteristics (in any order).
- Implicit characteristics can become driving characteristics if they are deemed *structural* concerns.
- Add additional characteristics identified that weren't deemed as important as the list of 7 to the *Others Considered* list.

Created by Mark Richards, DeveloperToArchitect.com

justifying architecture characteristics

1. Enable Discovery - Agility

The customer experience in acquisition channels (mobile, web and even SMS) must be seamless. This requires custom experimentation and optimization. The architecture must provide ways to capture customer behaviour with comprehensive analytics and support A/B testing. It's a plus if it can also provide rich experiences like smart recommendations powered by AI/Machine Learning. Those features must be immediatelly available on pay-as-you go basis, instead of requiring significant upfront investments in development or technology.

tying back to requirements

Requirement: Must integrate with 3rd party smart fridges to obtain inventory and purchase activity

Characteristics: ***Reliability*** - If smart fridges fail to communicate with the Farmacy Food system on item inventory levels and purchases, it will impact the reliability of the Farmacy Food system.

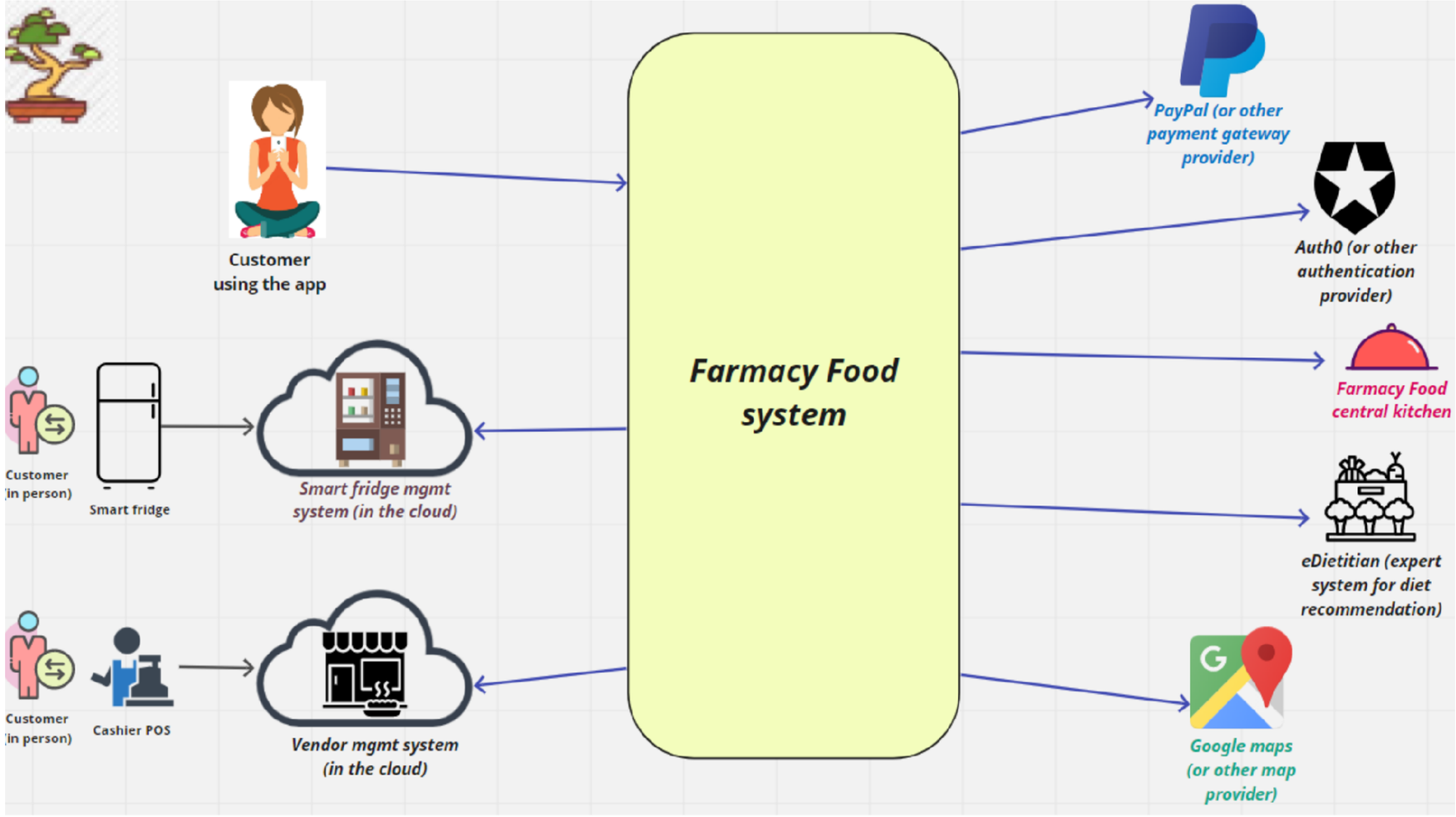
Availability: Farmacy food system plays the middle man role between cheftec and smart fridge.

SuperKings

https://github.com/lastlegion/arch-katas

Jiakaturi

- Agility
- Viability
- Availability
- Security
- Scalability
- Performance



architecture characteristics

YOU BE THE JUDGE!

Review the architecture characteristics that Jaikaturi identified. Select all the characteristics that you feel are properly justified and that you agree with.

Agility	Availability	Performance
Viability	Security	
Flexibility	Scalability	

Jiakaturi

https://github.com/lookfwd/archkata

Poll question:

Select all the characteristics from Jaikaturi that you feel are properly justified and that you agree with (MULTPLE CHOICE)

Agility

Viability

Flexibility

Availability

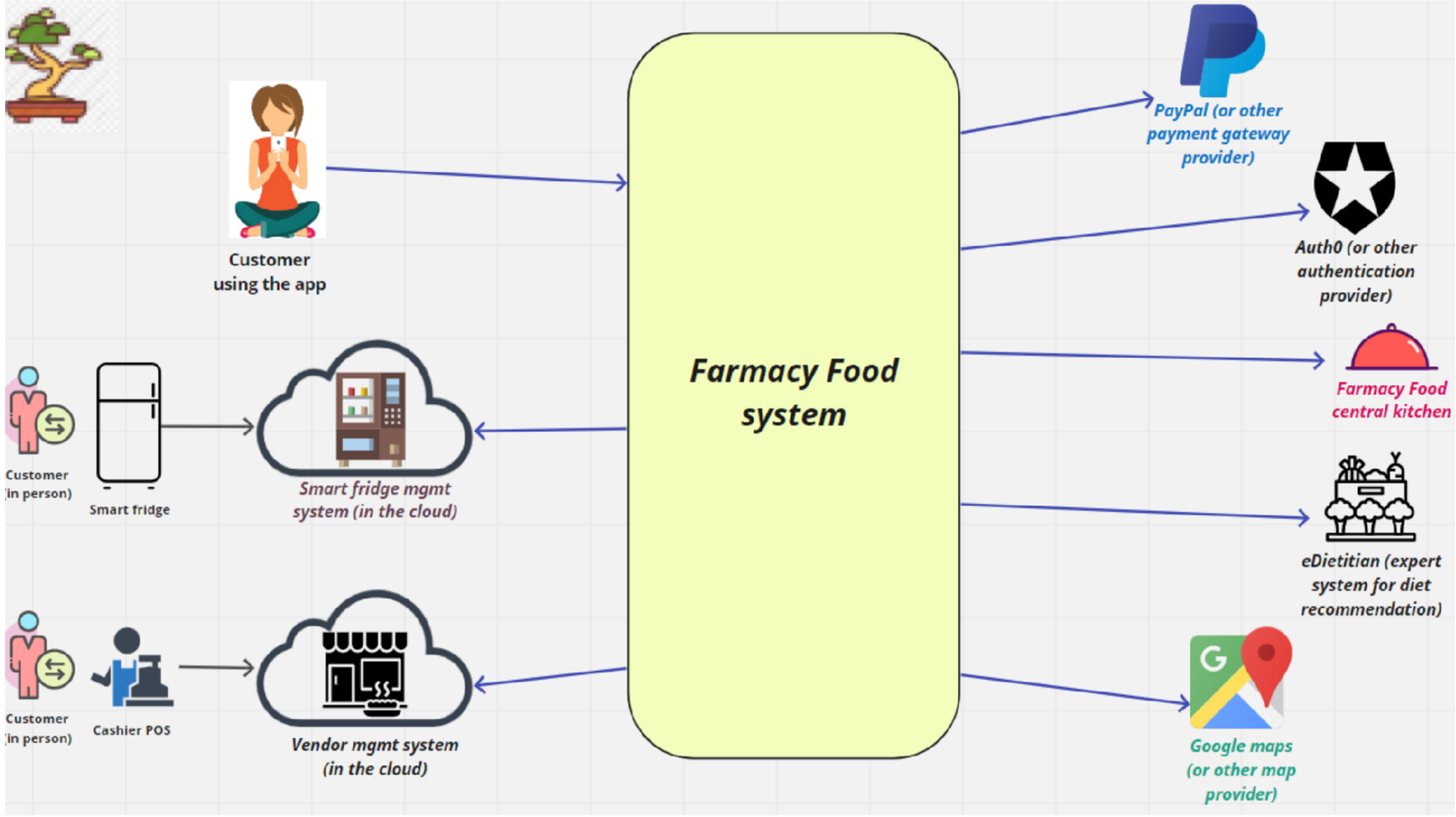
Security

Scalability

Performance

Jiakaturi

- Agility
- Viability
- Availability
- Security
- Scalability
- Performance

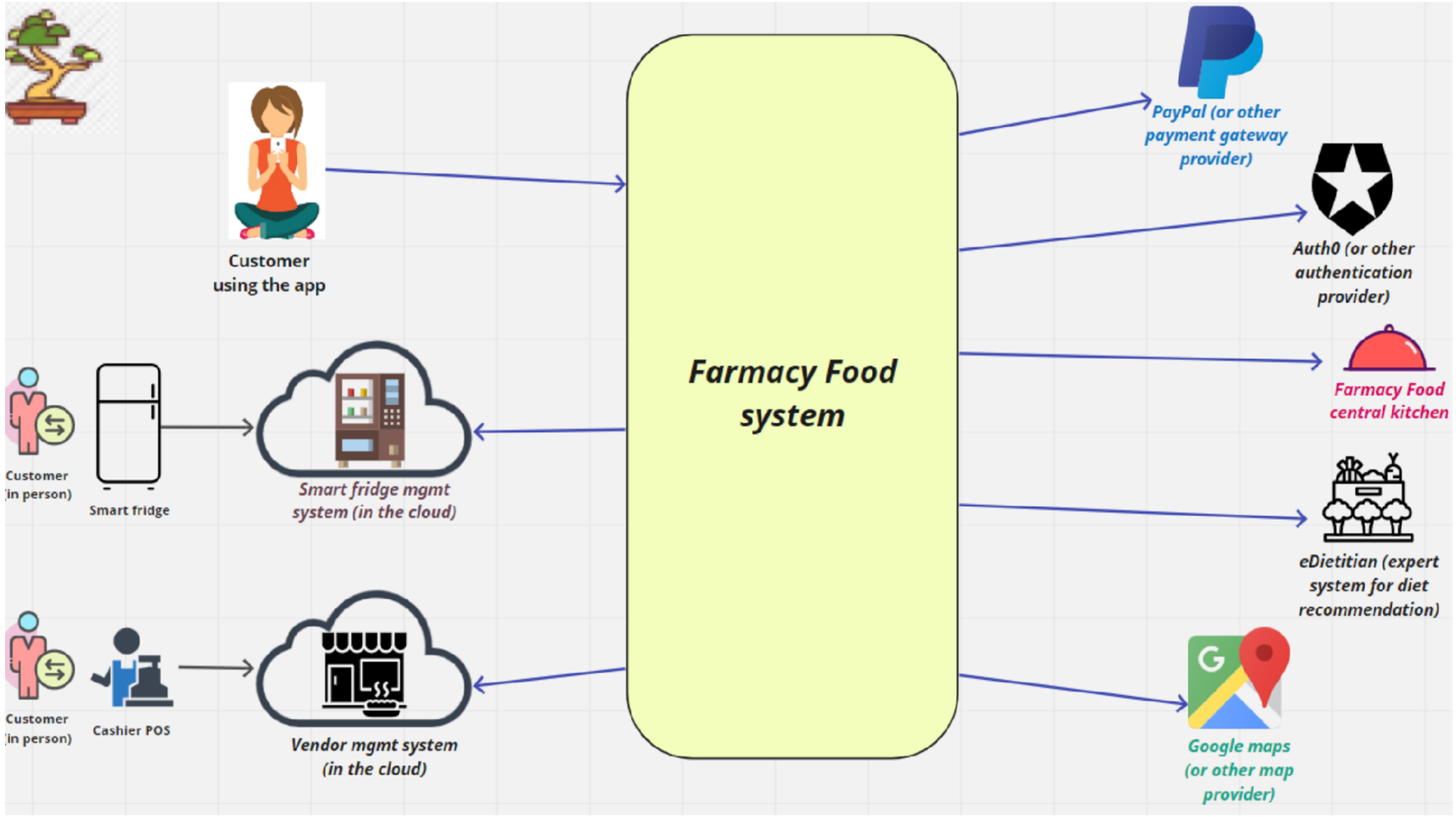


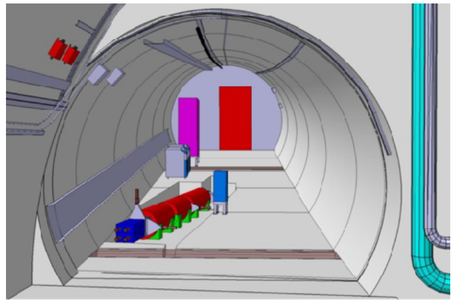
Interoperability

Scalability

Performance

Security





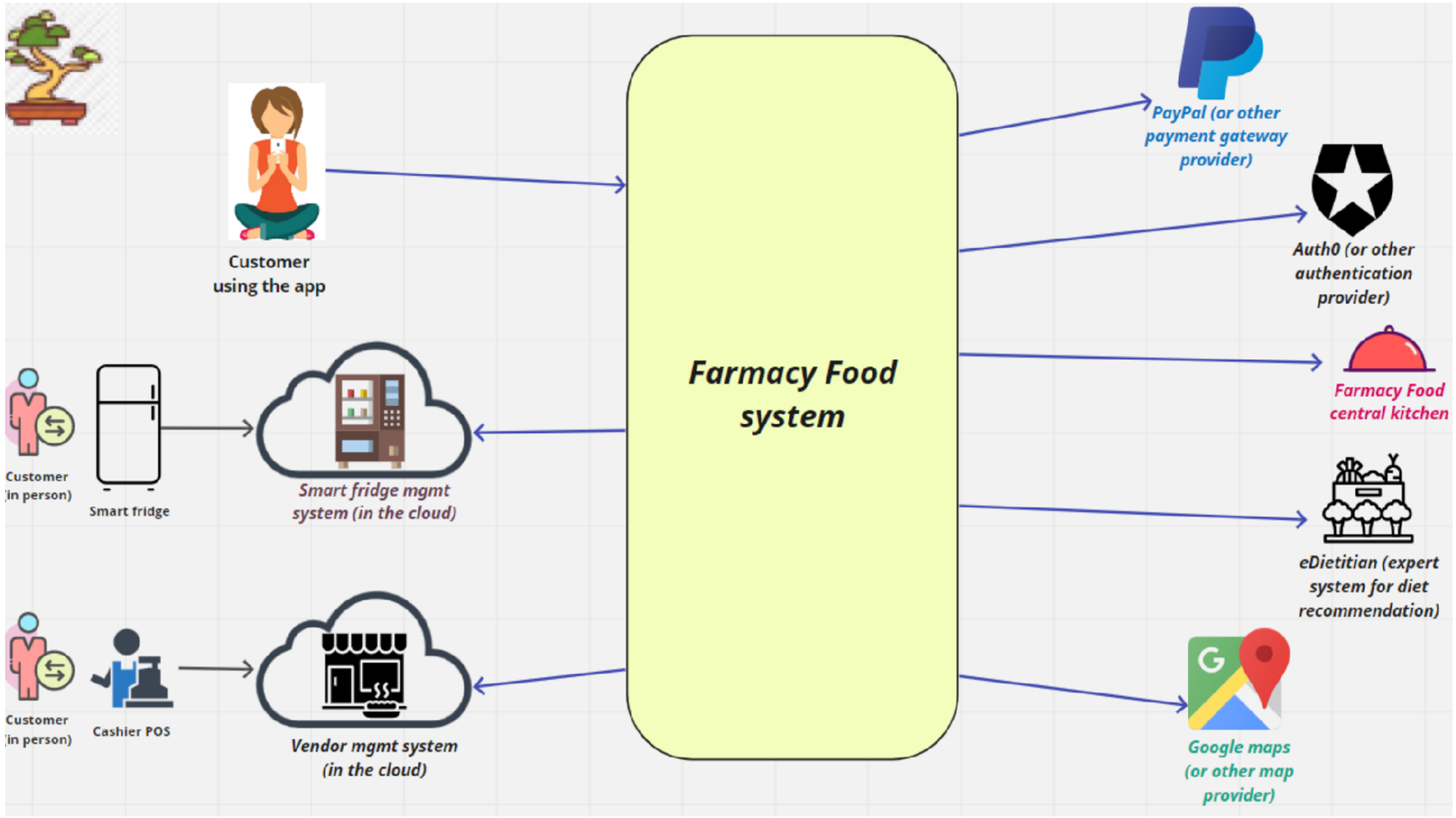
Simplicity

Modifiability

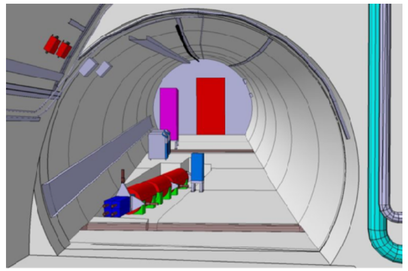
Availability

Data Integrity

Security



Jiakaturi



Agility
Viability
Availability
Security
Scalability
Performance

Security
Scalability
Performance
Interoperability

Modifiability

Availability
Security

Data Integrity
Simplicity

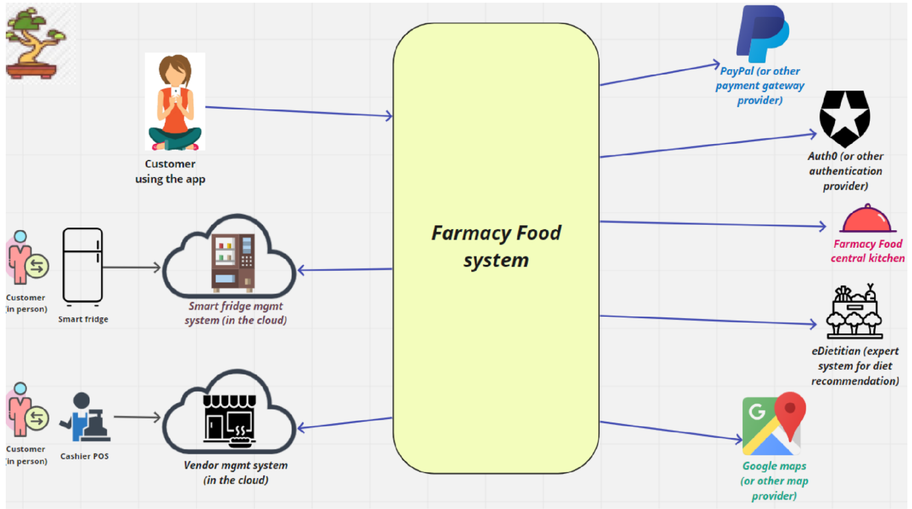
Agility
Viability
Availability
Security
Scalability
Performance
Interoperability
Data Integrity
Simplicity

architecture characteristics

YOU BE THE JUDGE!

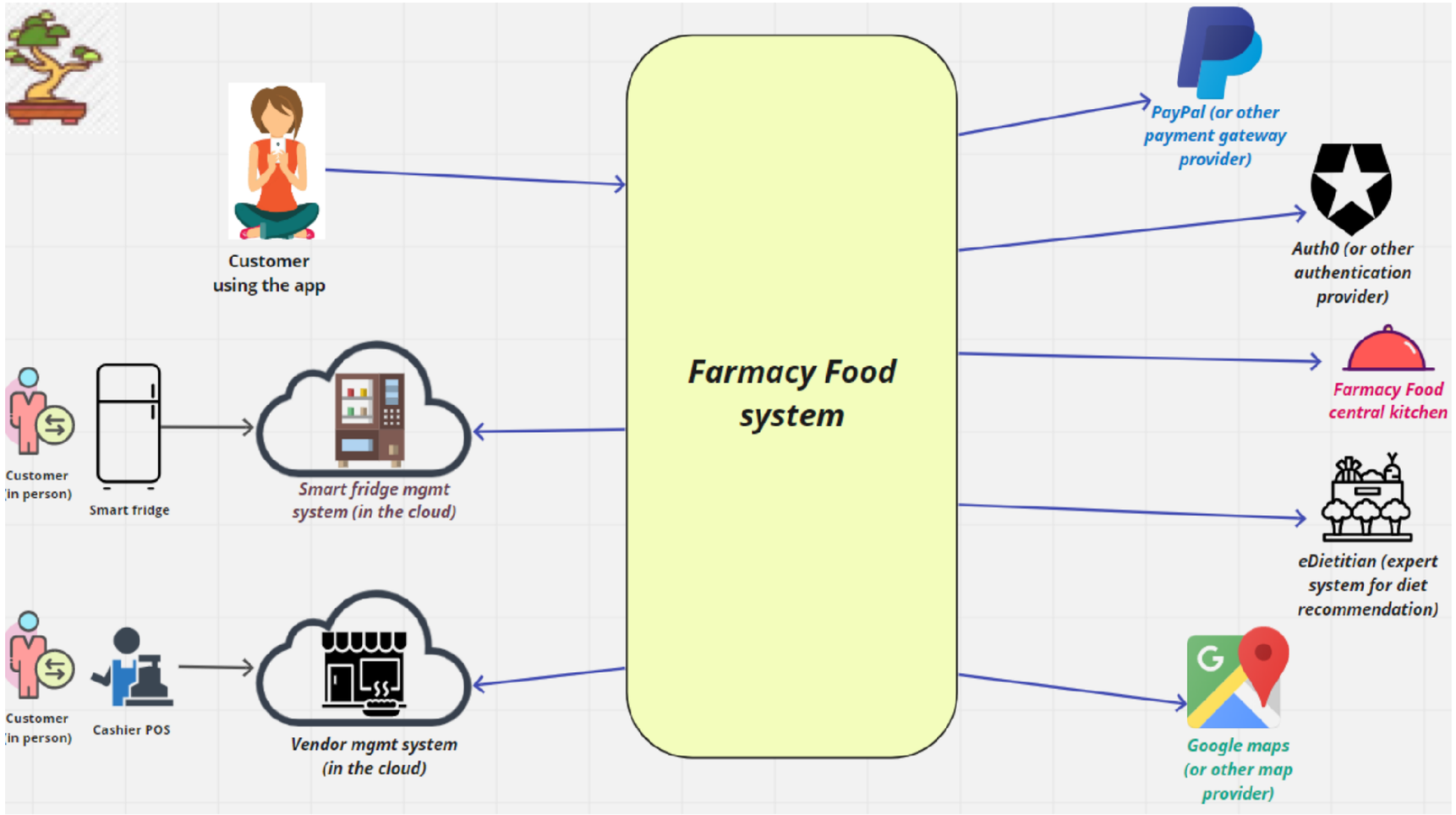
Which of the following do you feel are critical to the success of Farmacy Foods?

Agility	Security	Interoperability
Viability	Scalability	Data Integrity
Availability	Performance	Simplicity



5 min

architecture characteristics



- Agility
- Viability
- Availability
- Security
- Scalability
- Performance
- Interoperability
- Data Integrity
- Simplicity

5 min

Poll question:

Which of the following do you feel are critical to the success of Farmacy Foods?
(MULTPLE CHOICE)

Agility

Viability

Availability

Security

Scalability

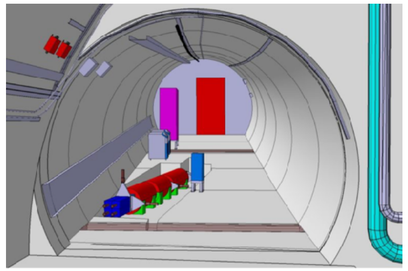
Performance

Interoperability

Data integrity

Simplicity

Jiakaturi



Agility

Viability

Availability

Security

Scalability

Performance

Security

Scalability

Performance

Interoperability

Modifiability

Availability

Security

Data Integrity

Simplicity

Critical or important for success?

Agility

Viability

Availability

Security

Scalability

Performance

Interoperability

Data Integrity

Simplicity

Poll question:

Which of the following do you feel are the **top 3** most important (in any order)
(MULTPLE CHOICE)

Viability

Availability

Security

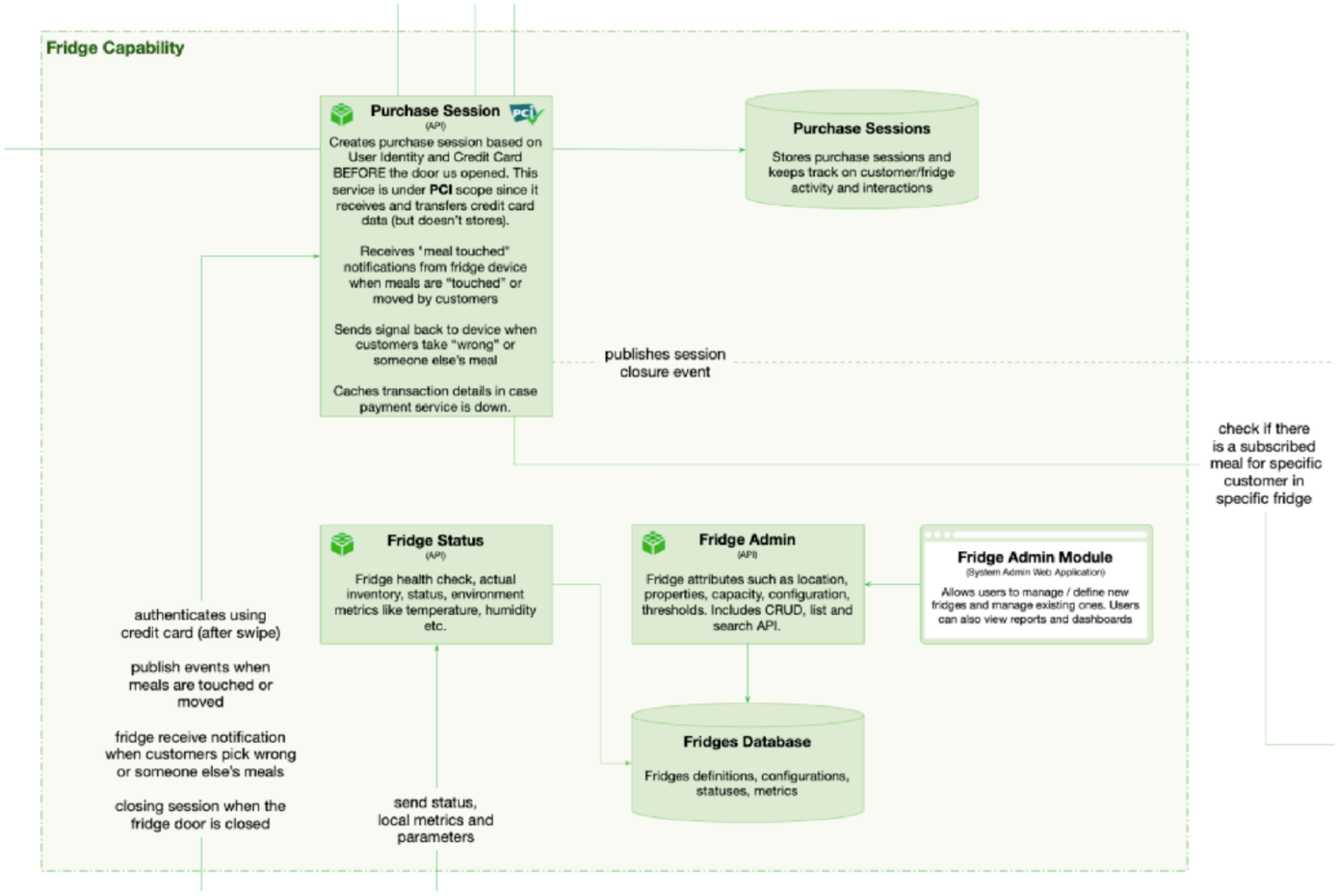
Scalability

Interoperability

Simplicity

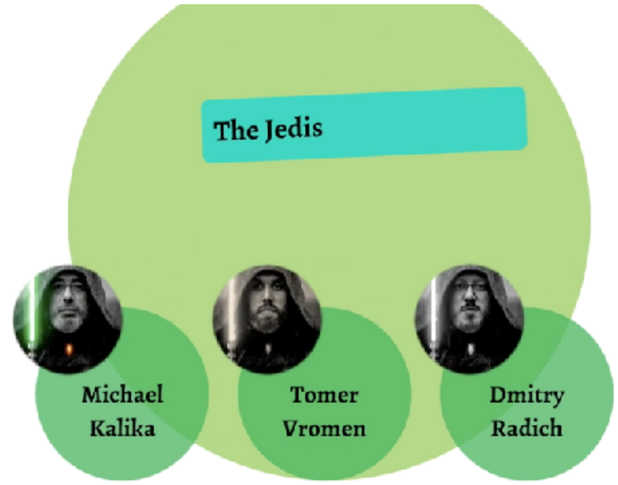
scope of characteristics

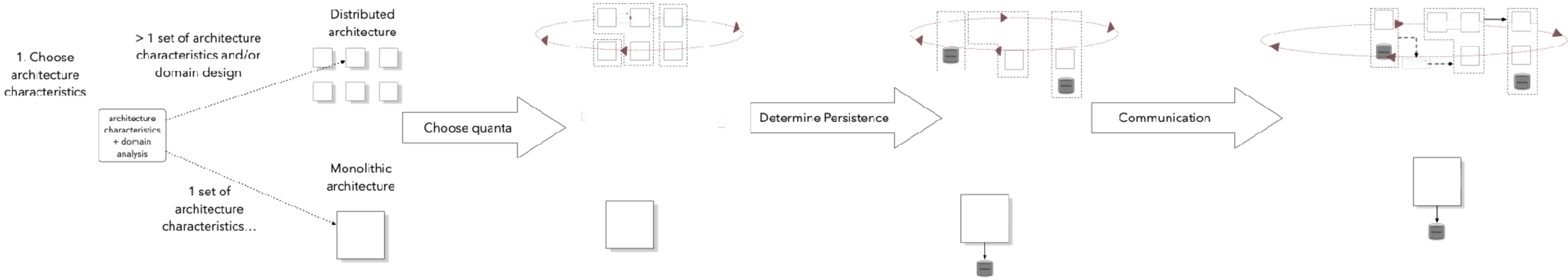
Fridge Capability

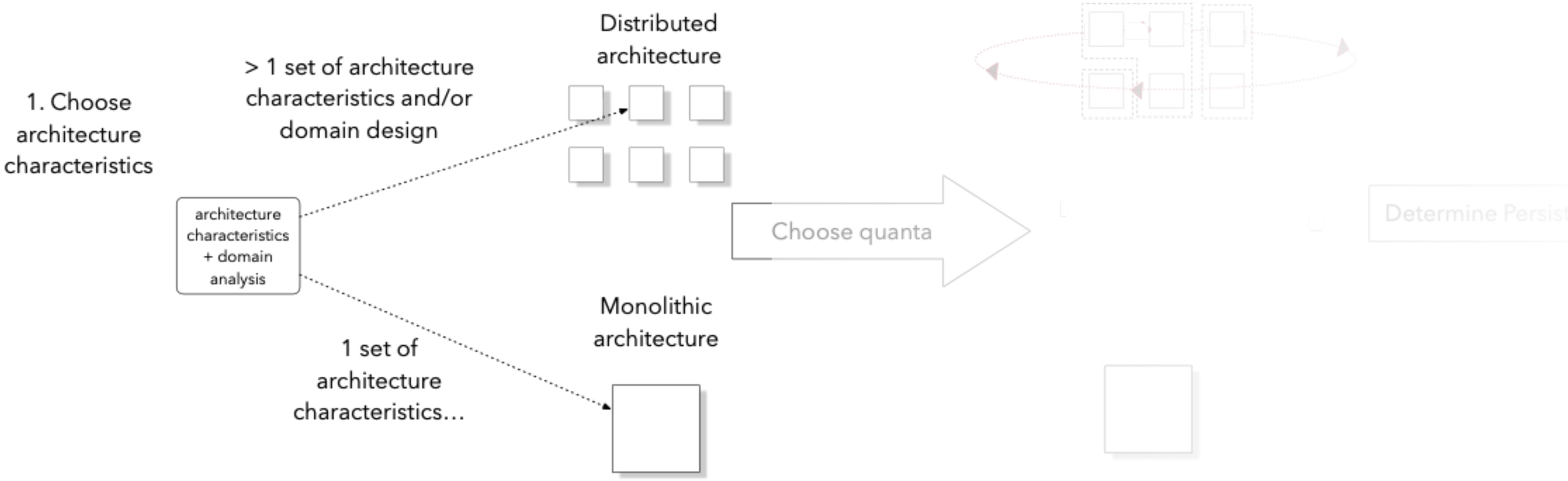


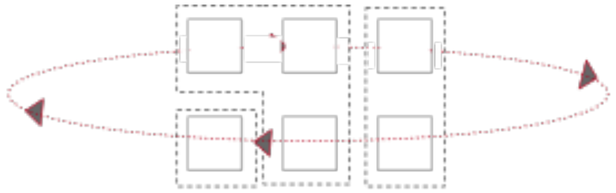
Architectural characteristics

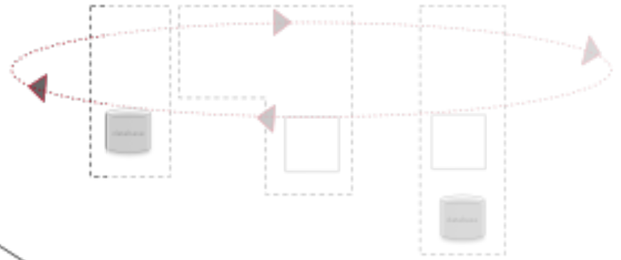
- Fault Tolerance
- Elasticity
- Scalability
- Testability




1. Choose architecture characteristics
architecture characteristics + domain analysis
> 1 set of architecture characteristics and/or domain design
Distributed architecture
1 set of architecture characteristics...
Monolithic architecture
Choose quanta
Determine Persistence
Communication


1. Choose architecture characteristics
> 1 set of architecture characteristics and/or domain design
architecture characteristics + domain analysis
Distributed architecture
1 set of architecture characteristics…
Monolithic architecture
Choose quanta

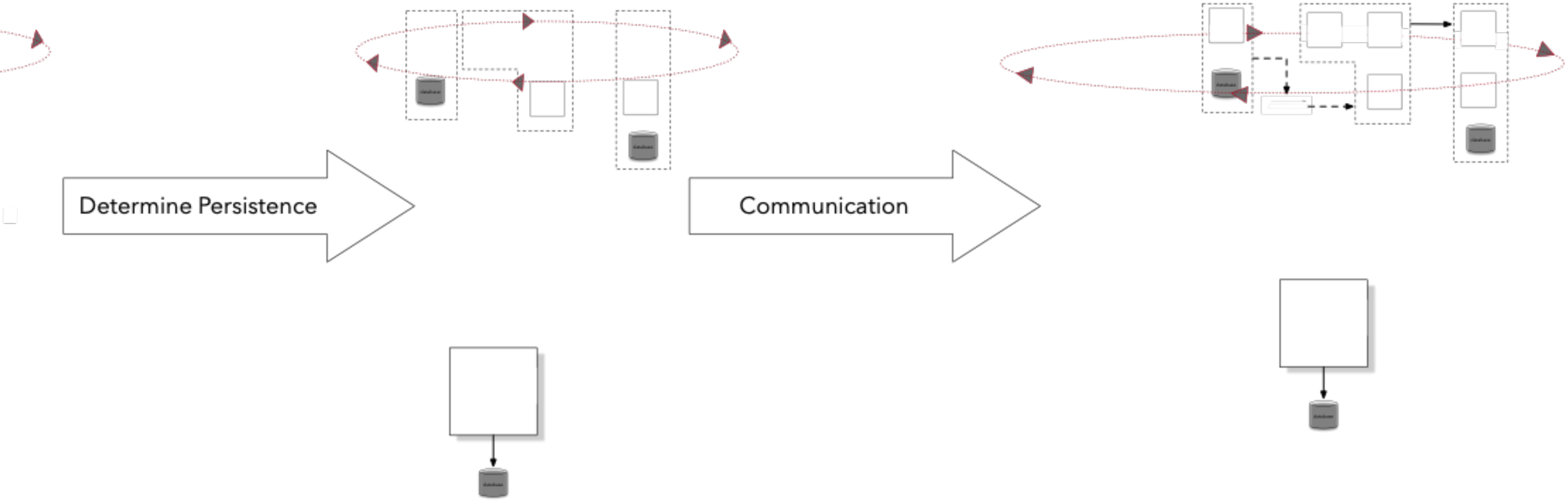




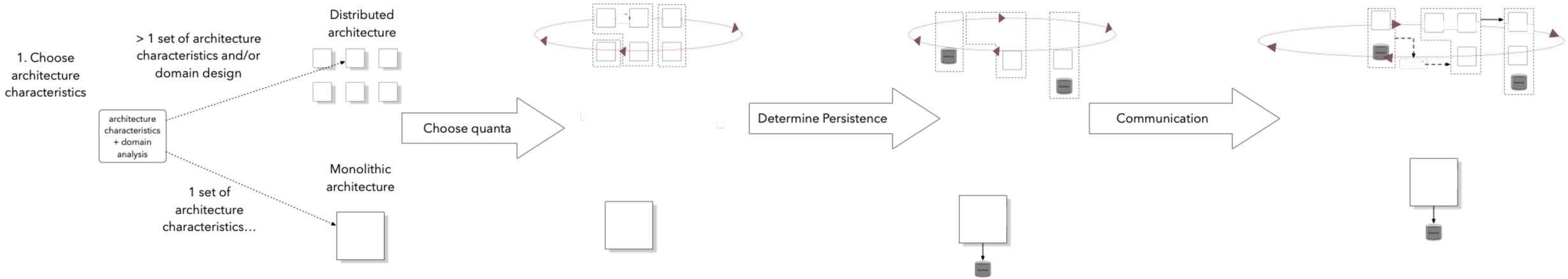
uanta

Determine Persistence

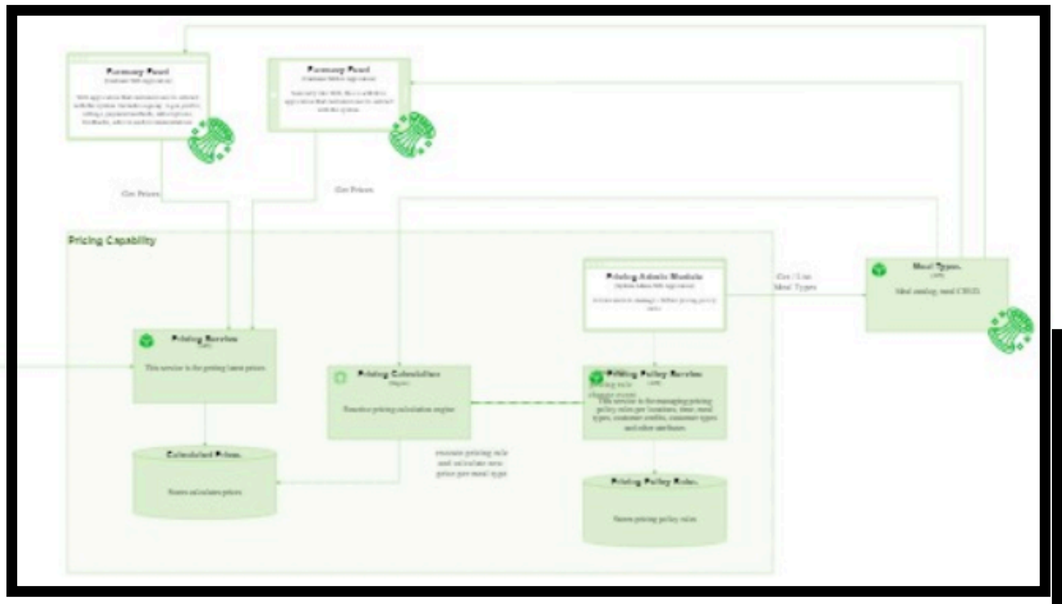
Communication

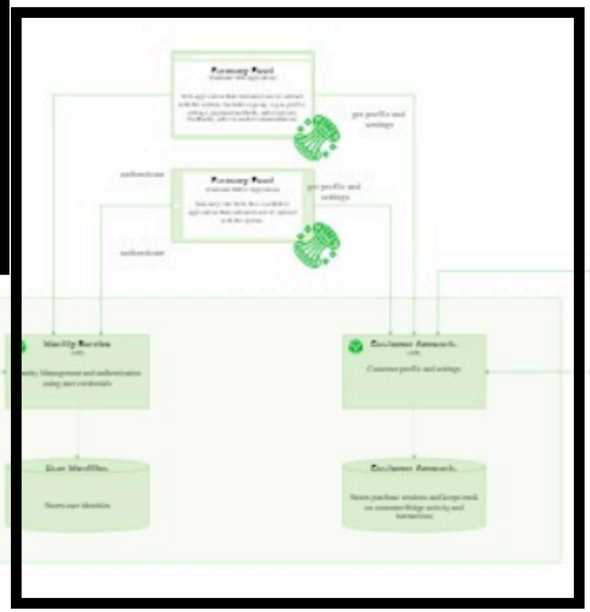

Determine Persistence
Communication

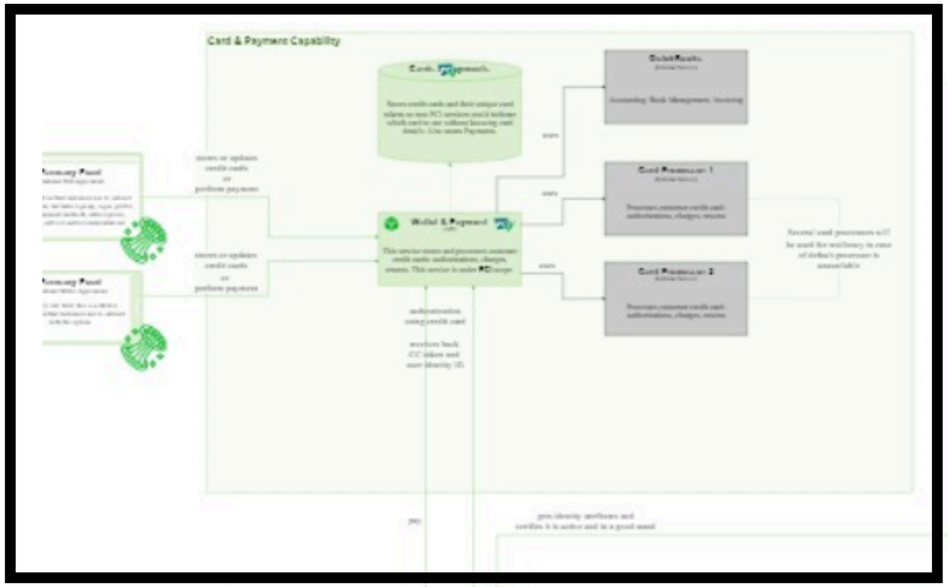
Monolith | Distributed ?

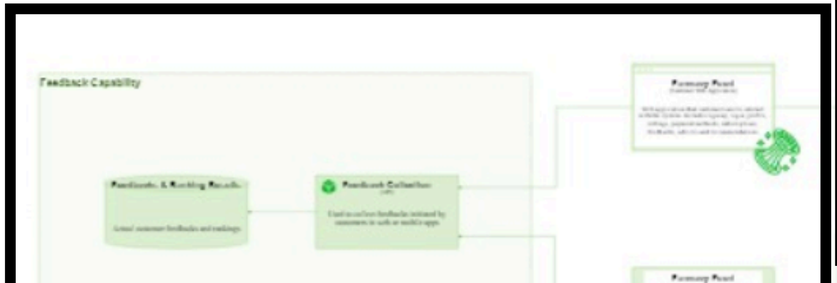


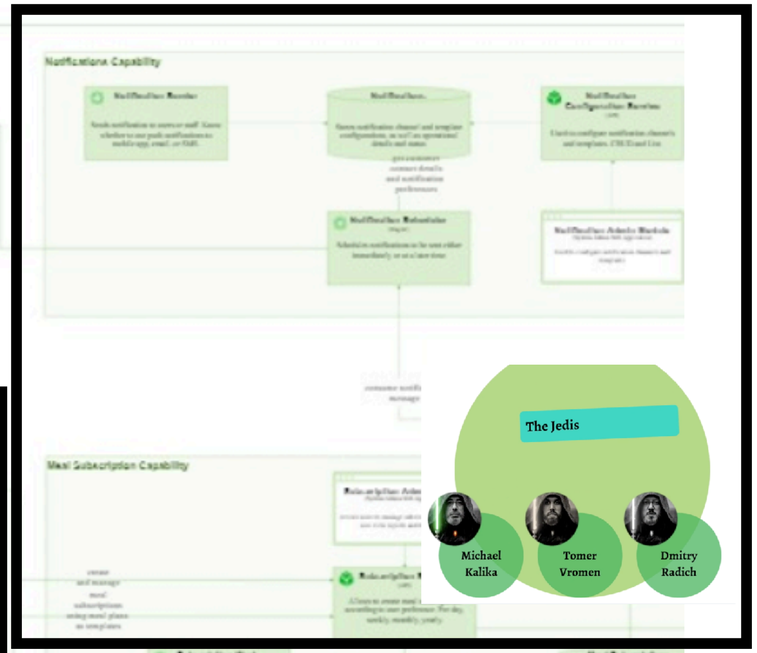
scope of architecture characteristics











scope of architecture characteristics

Stage 1 Capabilities

- Fridge Capability
- Card and Payment
- Identity and Profile
- Kitchen Capability
- Meal Inventory
- Pricing

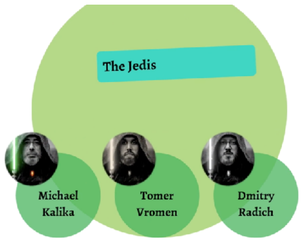
Stage 2 Capabilities

- Customer Subscriptions
- Notifications
- Feedback and Ranking
- Referrals and Rewards

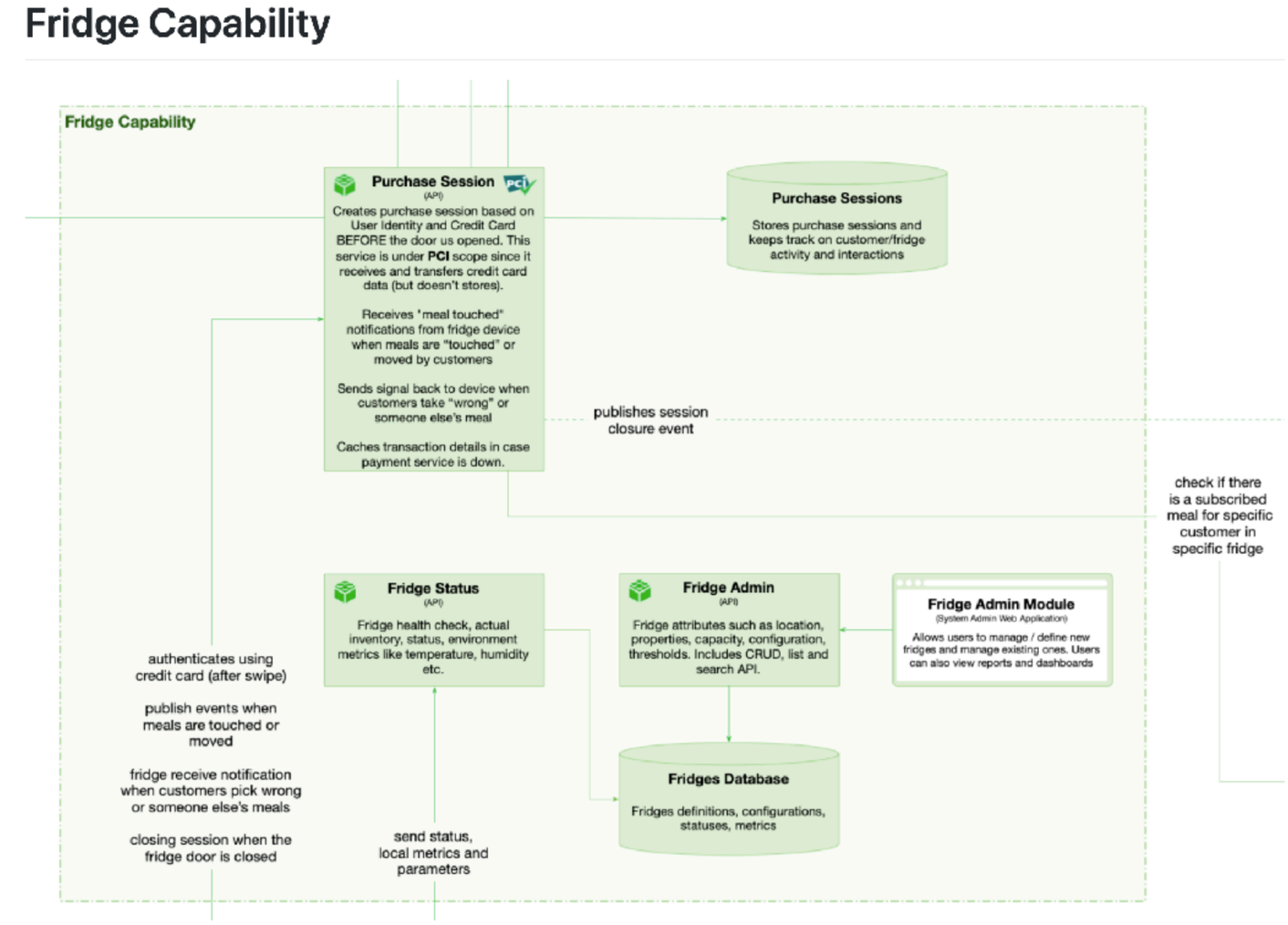
Stage 3 Capabilities

- Data Platform
- Expert Platform and CMS

each capability of the architecture has it’s own set of architecture characteristics

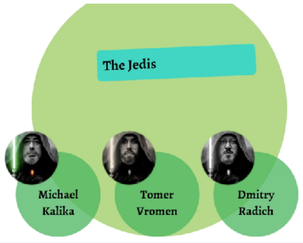


scope of architecture characteristics



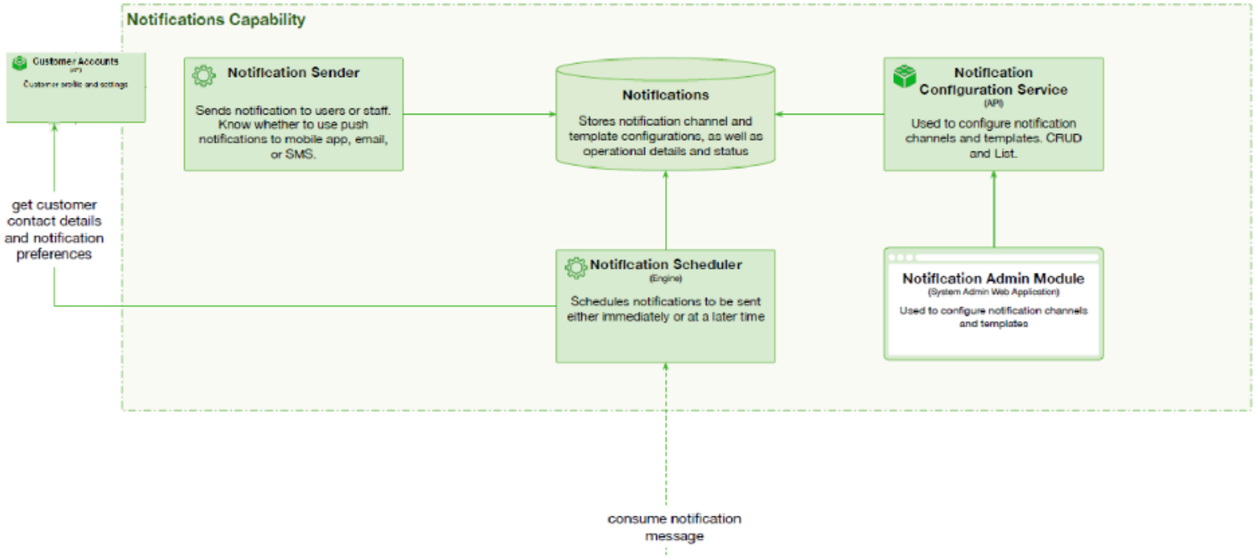
Architectural characteristics

- Fault Tolerance
- Elasticity
- Scalability
- Testability



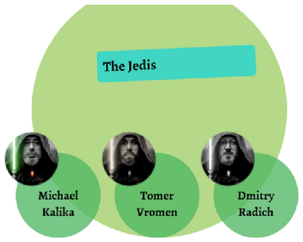
scope of architecture characteristics

Notifications Capability



Architectural characteristics

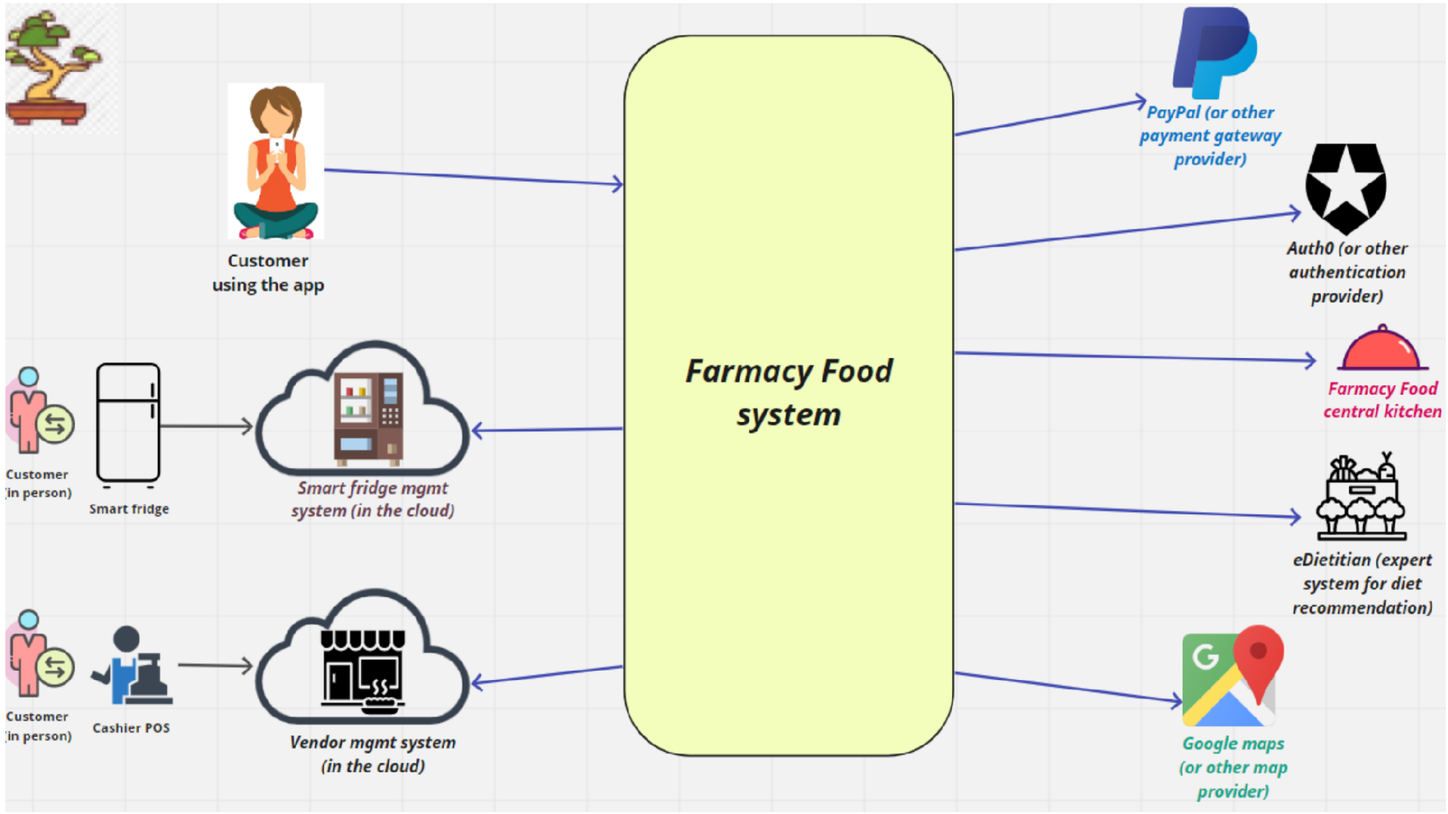
- Elasticity.
- Fault tolerance.
- Plugin support.



architecture characteristics

YOU BE THE JUDGE!

Should Farmacy Foods be a monolithic or distributed architecture?



Poll question:

Should Farmacy Foods be a monolithic or distributed architecture?

Monolithic

Distributed

Architecture Decisions

Is this an architecture decision or a design decision?

ADR 005: Use a CQRS Pattern

We have a microservice architecture with several REST services that would need to read and write data. Sometimes it's difficult to design those services to do both tasks efficiently. In the Farmacy Food system, the design of writes (commands) is primarily concerned with the integrity of data and business rules, whereas the design of reads (queries) has performance (response time) as a key requirement.

Decision

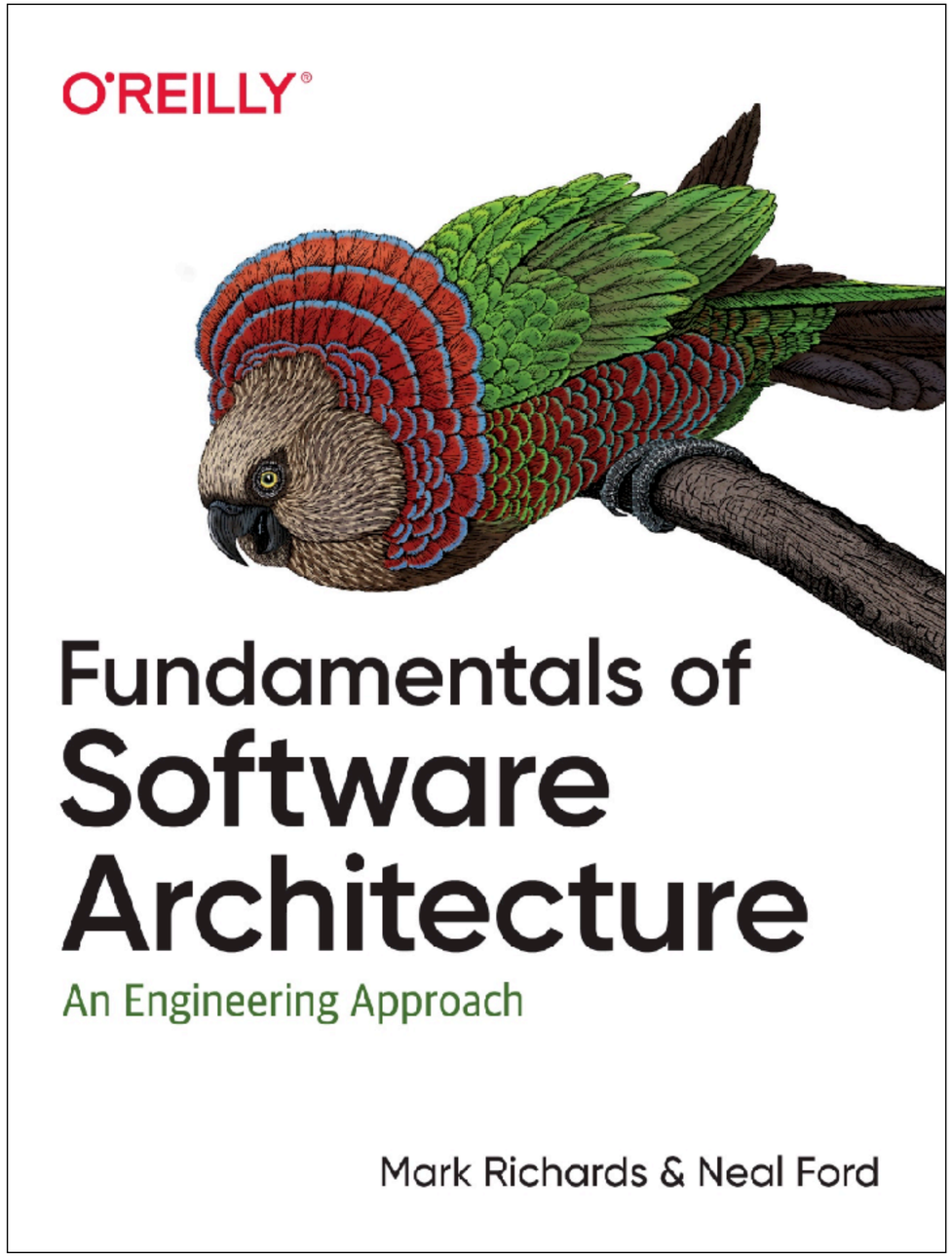
We will use the CQRS pattern applied to microservices with this design:

- an independently deployed service takes care queries. It reads data from a query view that is optimized for data reads. ("Optimized" means using DB technologies and techniques that favor performance, such as: NoSQL, desnormalized view, memory cache, sharding, replication.)
- an independently deployed service takes care of commands. It may be a REST service that gets POST commands or a reactive service that subscribes to a pub-sub topic and gets messages that updates the master database for the scope of that service.
- a batch component (example: K8S cron job or logic activated by database-triggered events) synchronizes the data between the master database and the query views. The frequency of data synchronization varies--it can be trigger-based, every 5 minuts, every hour, ..., once a day, etc.

Architecture decisions answer the “why” question and provide justification and trade-off analysis for decisions made about the architecture

- ☑ Is it an architecture decision or a design decision?
- ☑ Is the decision properly justified?
- ☑ What is the scope of the decision?

Architecture Decisions Checklist	
☐	Is this an architecture or design decision?
☐	Is the decision properly justified?
☐	What is the scope of the decision?



Second Law of Software Architecture

"Why is more important than how"

architecture vs. design decision

Definition of Software Architecture:

Decisions where each possibility features significant tradeoffs.

architecture vs. design decision

architecture vs. design decision

A B C D

Long-term capabilities/restrictions

“Stuff that’s hard to change later.”

Significant trade-offs on for each option.

architecture vs. design decision

A B C D

Medium-term capabilities/restrictions

architecture vs. design decision

A B C D

Collaboration is key.

Shared understanding & justification.

Where strategy & tactics meet.

architecture vs. design decision

A B C D

Decisions of consequence…

…with safety nets (fitness functions)

Next generation of architects.

architecture decision records

short text file; 1-2 pages long, one file per decision
markdown, textile, asciidoc, plaintext, etc.

short noun phrase

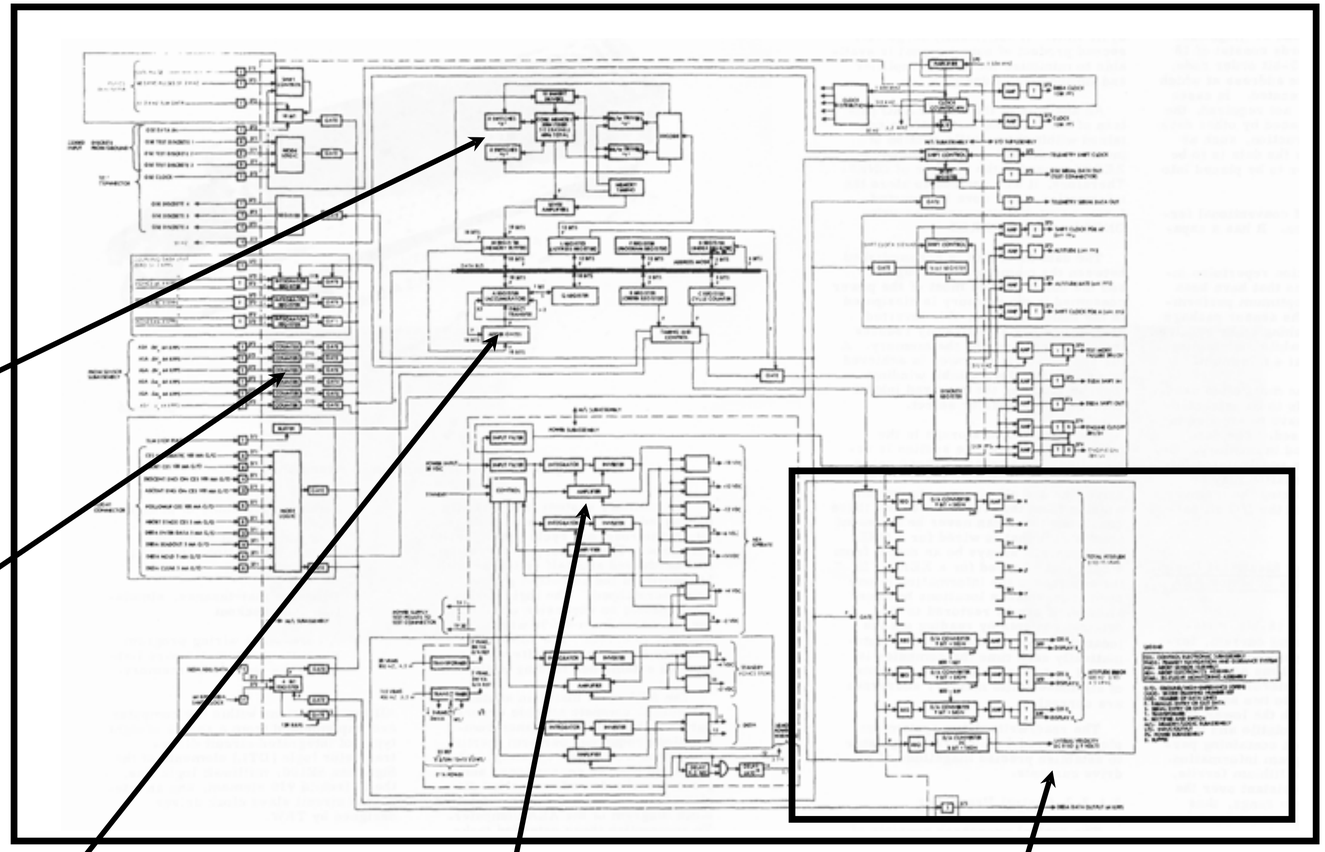
proposed, accepted, superseded

description of the problem and alternative solutions available (documentation)

decision and justification (the “why”)

trade-offs and impact of decision

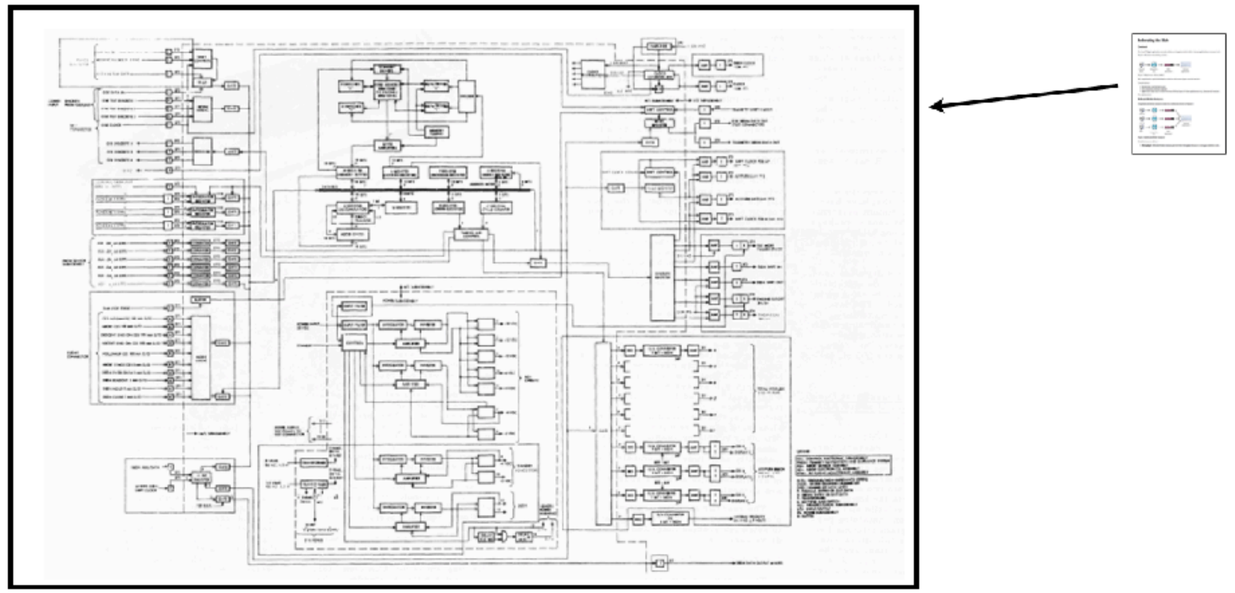
architecture decision scope



architecture decision scope

ADR 001: Use the microservice architecture style with containerization

Farmacy Food is a start up company and does not have a sizeable team of experienced developers available. The overarching architecture style for the Farmacy Food system should be simple, easy to create, maintain and **evolve**. Finding developers that can create and evolve the system, as well as tools and frameworks that support the system should not require heaps of money. In other words, Farmacy Food is not in a position to be an *early adopter*, and should hence adopt an established architecture style that supports evolution.



architecture decision scope

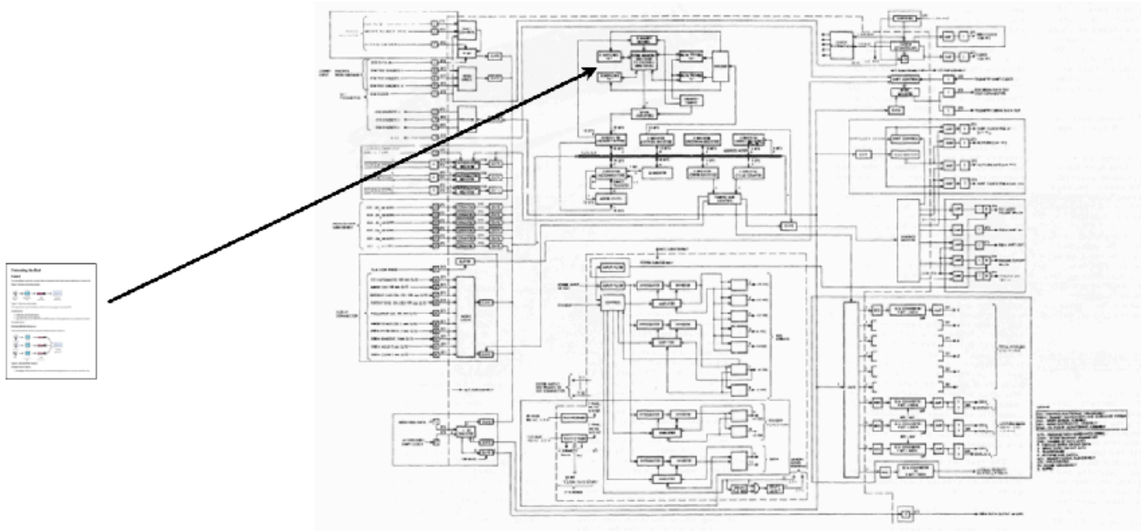
ADR_004 Use a centralized notification for external communication

Context

There was some confusion around the purpose of the notification component. Specifically, is this component an event bus for **all** communication or is it a shared component for communicating externally.

Decision

We decided to have a dedicated notification system responsible of sending external notification. The reasons include:



selfdriventeam

architecture decision scope

ADR 1. Hosting Platform

Context: The architecture needs to be hosted either on dedicated servers or in the cloud. The choice of cloud provider affects th implementation. Jiakaturi

context-based title

ADR 001: Use the microservice architecture style with containerization

Farmacy Food is a start up company and does not have a sizeable team of experienced developers available. The overarching architecture style for the Farmacy Food system should be simple, easy to create, maintain and **evolve**.

decision-based title

architecture decisions

YOU BE THE JUDGE!

Which ADR format is more effective?

Context-Based Title

ADR001: Architecture Style

Decision-Based Title

ADR001: We will use Microservices

Poll question:

Which ADR title is more effective?

Context-Based - ADR001: Architecture Style

Decision-Based - ADR001: We will use Microservices

architecture decision scope

ADR 1. Hosting Platform

Context: The architecture needs to be hosted either on dedicated servers or in the cloud. The choice of cloud provider affects the cost of the solution and the time of the implementation. Jiakaturi

ADR 001: Use the microservice architecture style with containerization

Farmacy Food is a start up company and does not have a sizeable team of experienced developers available. The overarching architecture style for the Farmacy Food system should be simple, easy to create, maintain and **evolve**.

decision-based title

architecture decisions

YOU BE THE JUDGE!

Where in the spectrum are each of the following architecture decisions?

(who has the <u>responsibility</u> to make the decision?)

architecture ***design***

A B C D

Jiakaturi

architecture vs. design decision

ADR 1. Hosting Platform

Context: The architecture needs to be hosted either on dedicated servers or in the cloud. The choice of cloud provider affects the cost of the solution and the time of the implementation.

Decision: The platform of choice is the Google Cloud Platform (GCP) on the basis of ease of development, security, versatility and price. To avoid vendor lock-in, no components that don't have an AWS-equivalent should be used without explicit permission.

Consequences: The implementation is expected to scale elastically on demand. We can run experiments and develop staging and test environments easily. AWS remains a migration option if factors like host dictate it at some point. Notably, Google App Engine and BigQuery can *not* be used because of vendor lock-in concerns.

Poll question:

We will use the Google Cloud Platform as the hosting platform

A – All about architecture
B – Somewhat architecture
C – Both architecture and design
D – All about design

architecture vs. design decision

ADR 003: Use the BFF pattern

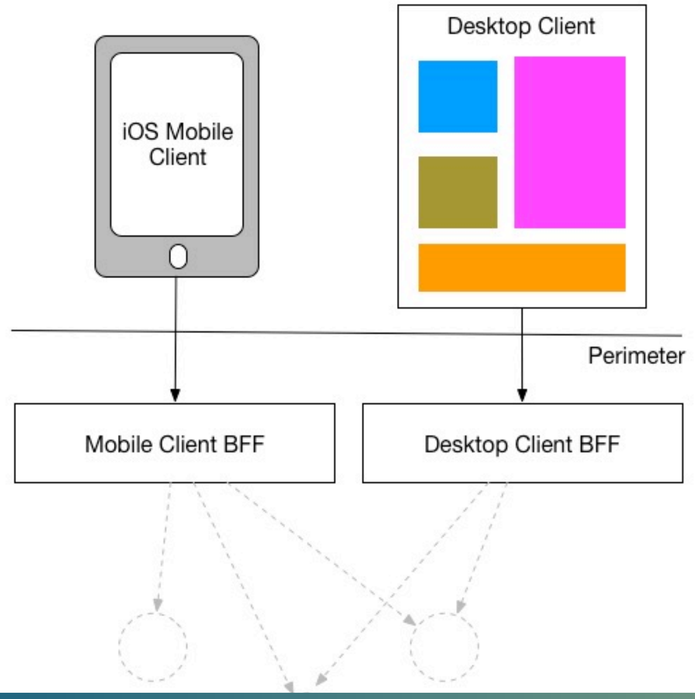
We have a microservice architecture with several REST services and different types of frontends: Web application, iOS application, Android application, public API clients (for the future), chatbot (also for the future). Different frontends may require slightly different message formats, message structures, headers, etc. Farmacy Food is a start-up with limited resources to configure, deploy, and govern more sophisticated middleware solutions, such as a full-fledged API gateway product.

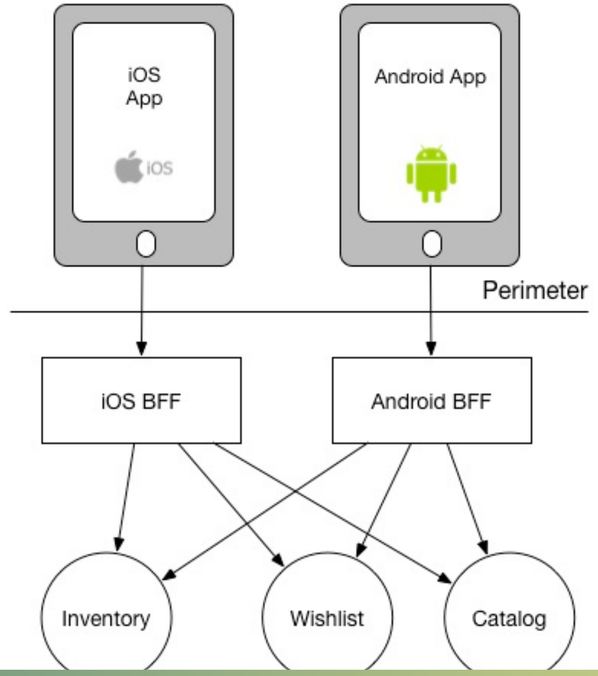
Decision

We will use the BFF pattern and have BFF services for each type of frontend as the central point of interaction with the Farmacy Food frontend apps. Moreover, instead of a single BFF service (for each frontend type) that interacts with *all* backend services, we will create separate BFF services per subdomain.

architecture vs. design decision

ADR 003: Use the BFF pattern





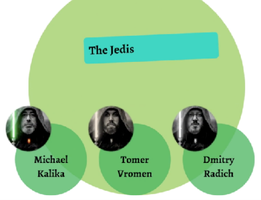
architecture — design

A B C D

Poll question:

We will use the BFF pattern for each user interface

A – All about architecture
B – Somewhat architecture
C – Both architecture and design
D – All about design



architecture vs. design decision

Customer IDs as part of meal status inventory messages

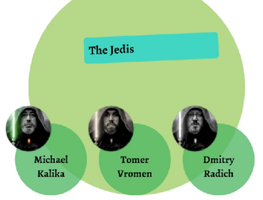
Context and Problem Statement

The Subscription Notification Engine consumes meal status update messages. It then needs to notify Notifications Scheduler to send personalized messages to customers. Should we embed customer IDs as part of meal status update messages with an option of it being empty in many cases, or should Subscription Notification engine query the Meals Inventory DB to attribute a meal to a customer?

Poll question:

Meal status inventory message payload will include only the customer ID

A – All about architecture
B – Somewhat architecture
C – Both architecture and design
D – All about design



architecture vs. design decision

Separate channels for anonymous and personal meal status messages

Context and Problem Statement

The "Farmacy Food" system maintains meal status via Meals Inventory subsystem.

The Meals Inventory subsystem consumes meal status messages from various other subsystems to maintain inventory integrity, indicating whether the meal has been ordered, produced, placed in a fridge, purchased or expired. The system also needs to notify subscribed customers of their meals status changes, but it does not send messages to anonymous customers.

Customer needs to create subscription at least once to be identified henceforth by the system. Since we assume that majority of meal purchases are anonymous, it means that a relatively small subset of meal status messages should be reflected to customers.

Poll question:

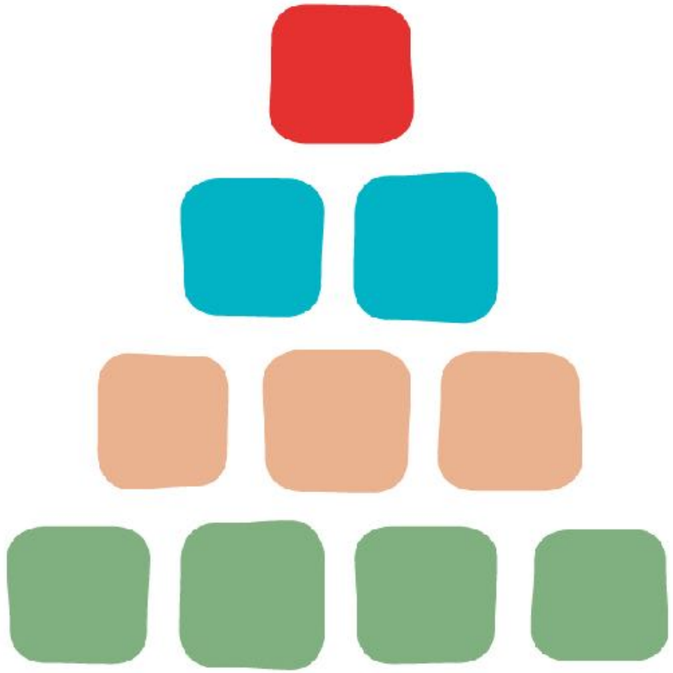
We will use separate message channels for anonymous and personal meal status messages

A – All about architecture
B – Somewhat architecture
C – Both architecture and design
D – All about design

Architecture Diagrams

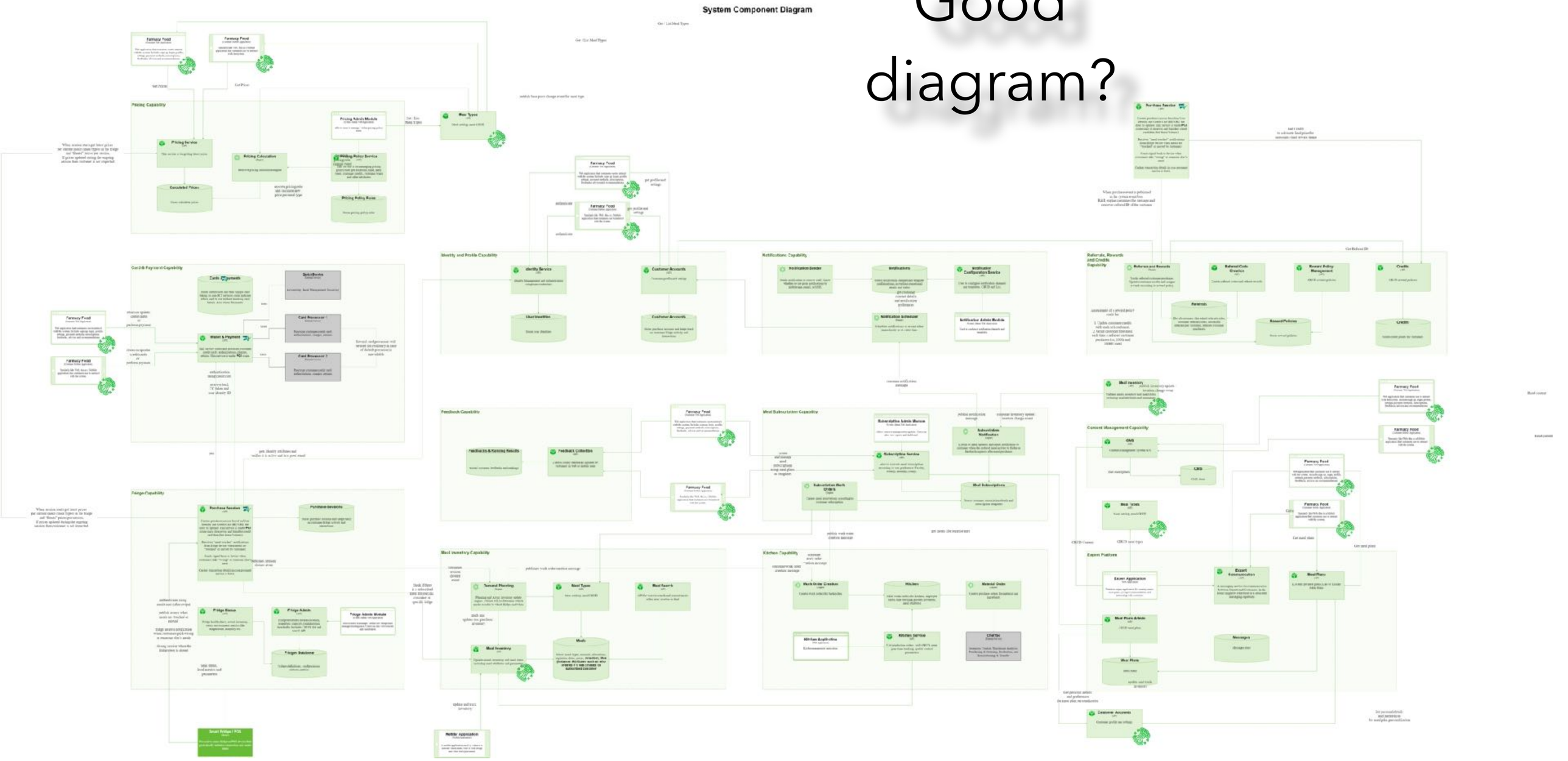
why is this criteria important?

An effective architecture picture is worth more than 1,000 words. Architecture represents topology, which benefits from visual representations.



- Short, meaningful titles
- Lines: descriptions
- Lines: unidirectional
- Lines: synchronous or asynchronous
- Lines: blocking or non-blocking
- Shapes: consistency
- Avoid acronyms
- Color check for consistency, contrast, clarity
- Most important thing(s) centered
- Includes key

Good diagram?

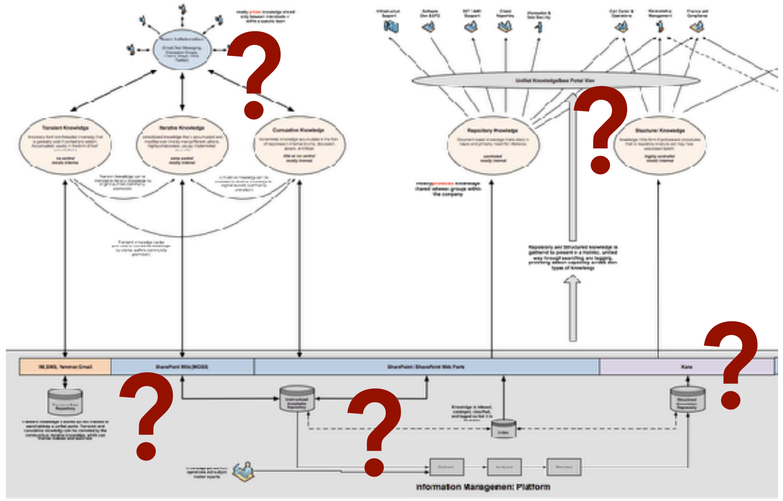


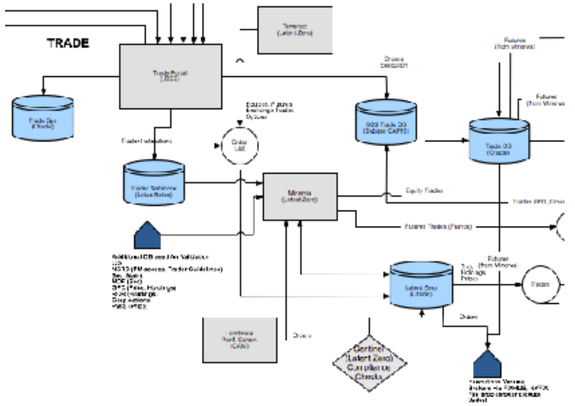
Architecture Diagram Guidelines

diagramming architecture

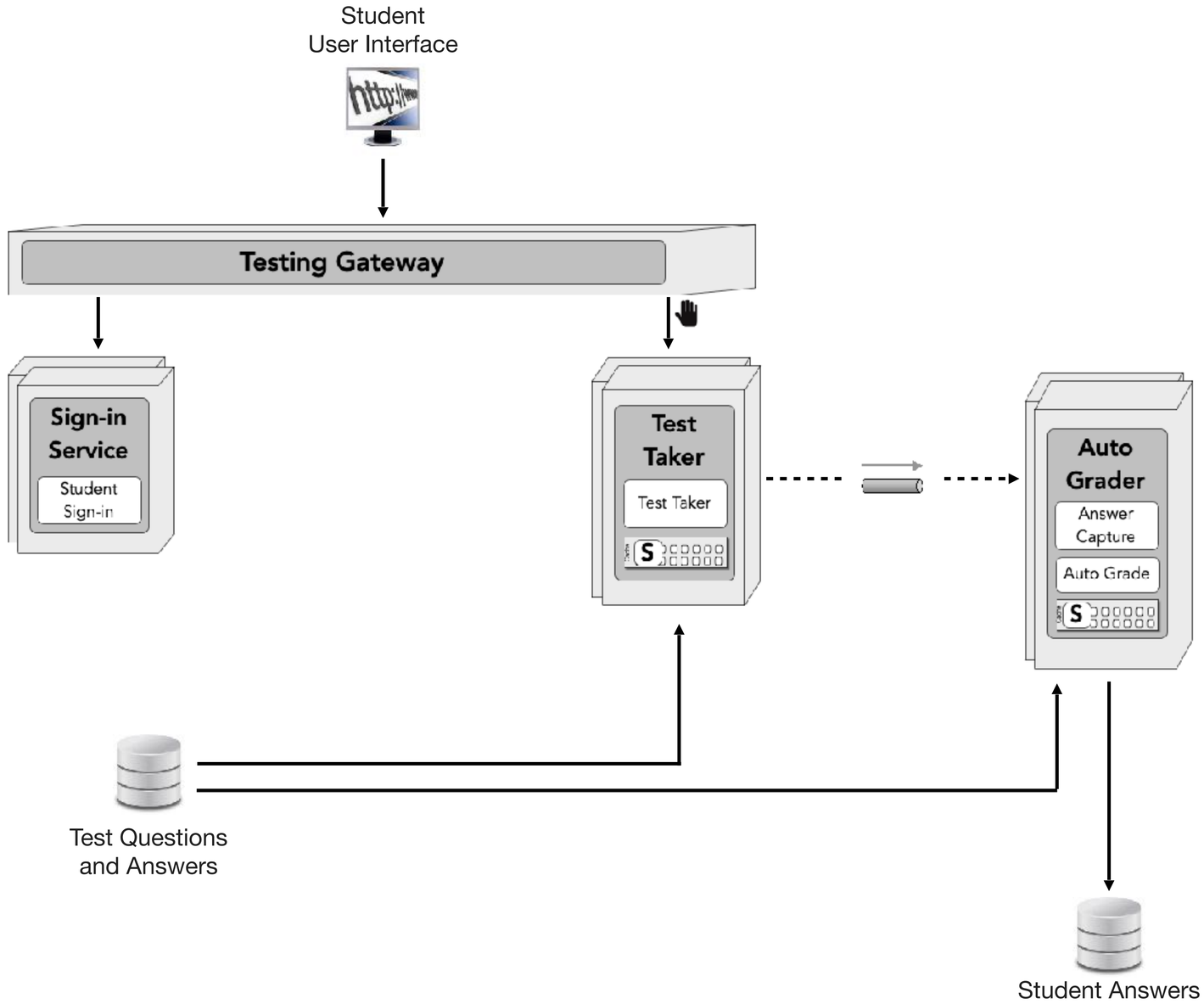
representational consistency

maintain context when moving between
models or different views of a model

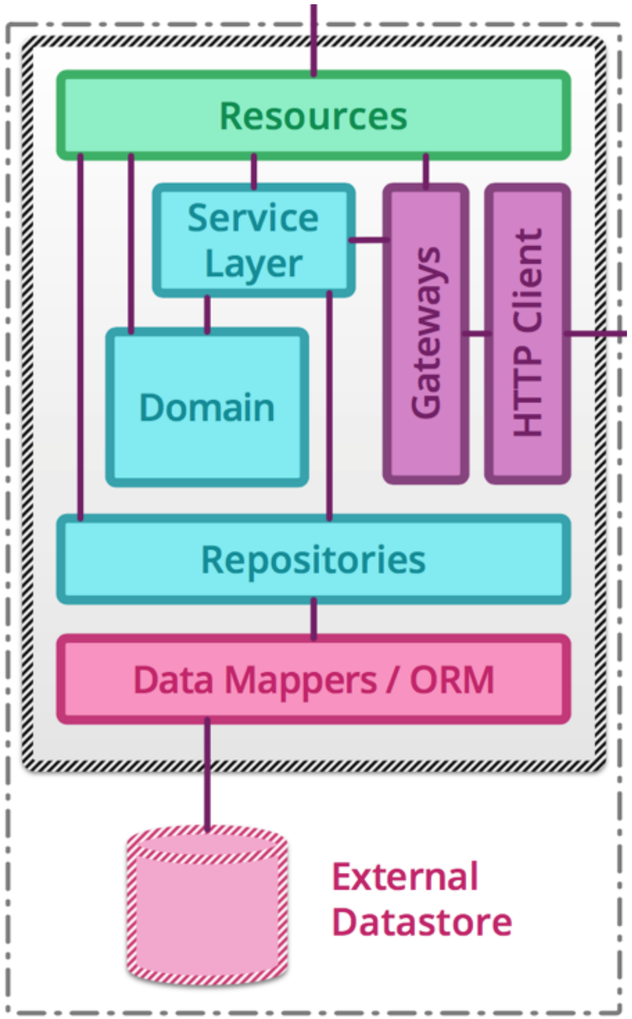




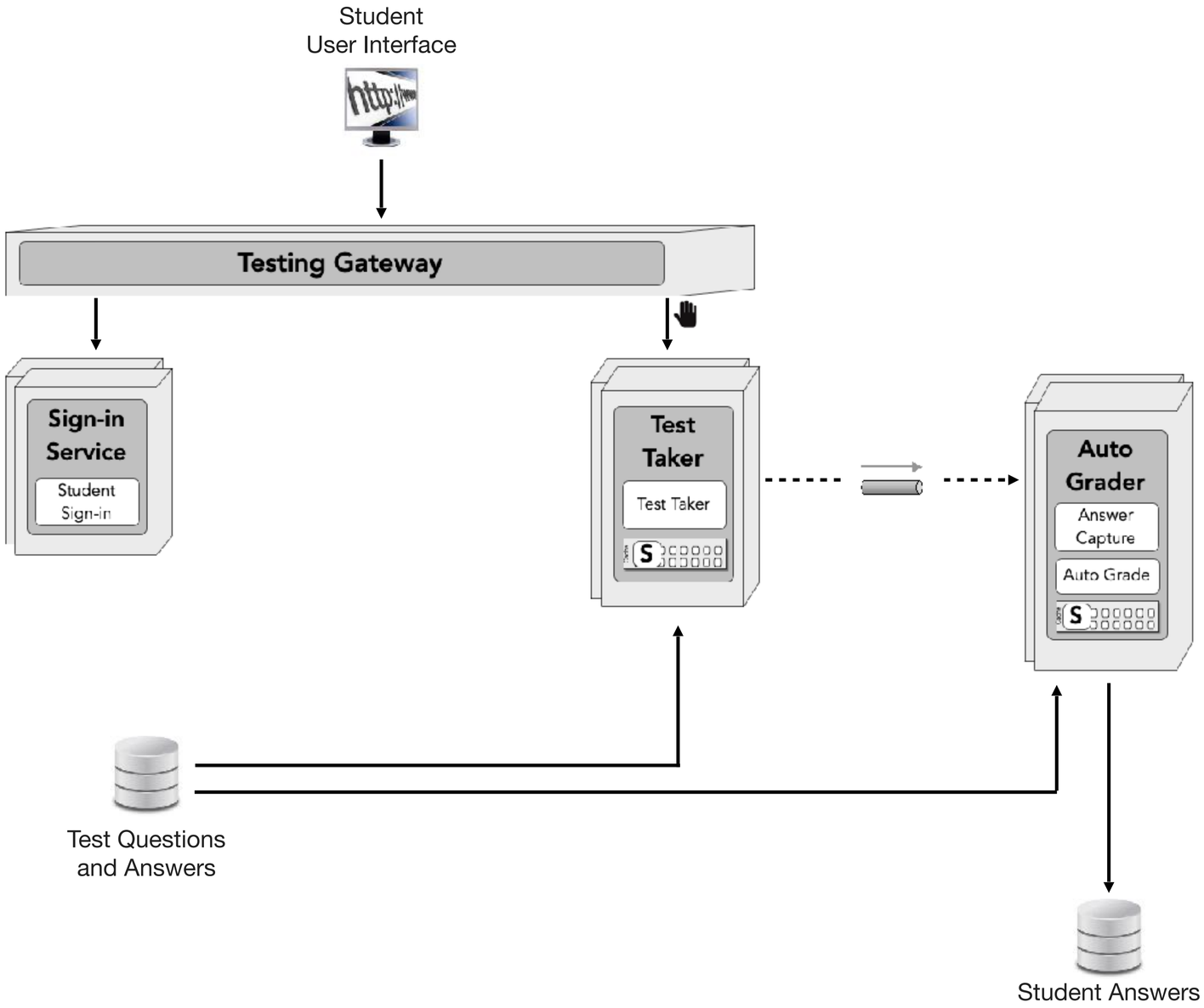
diagramming architecture



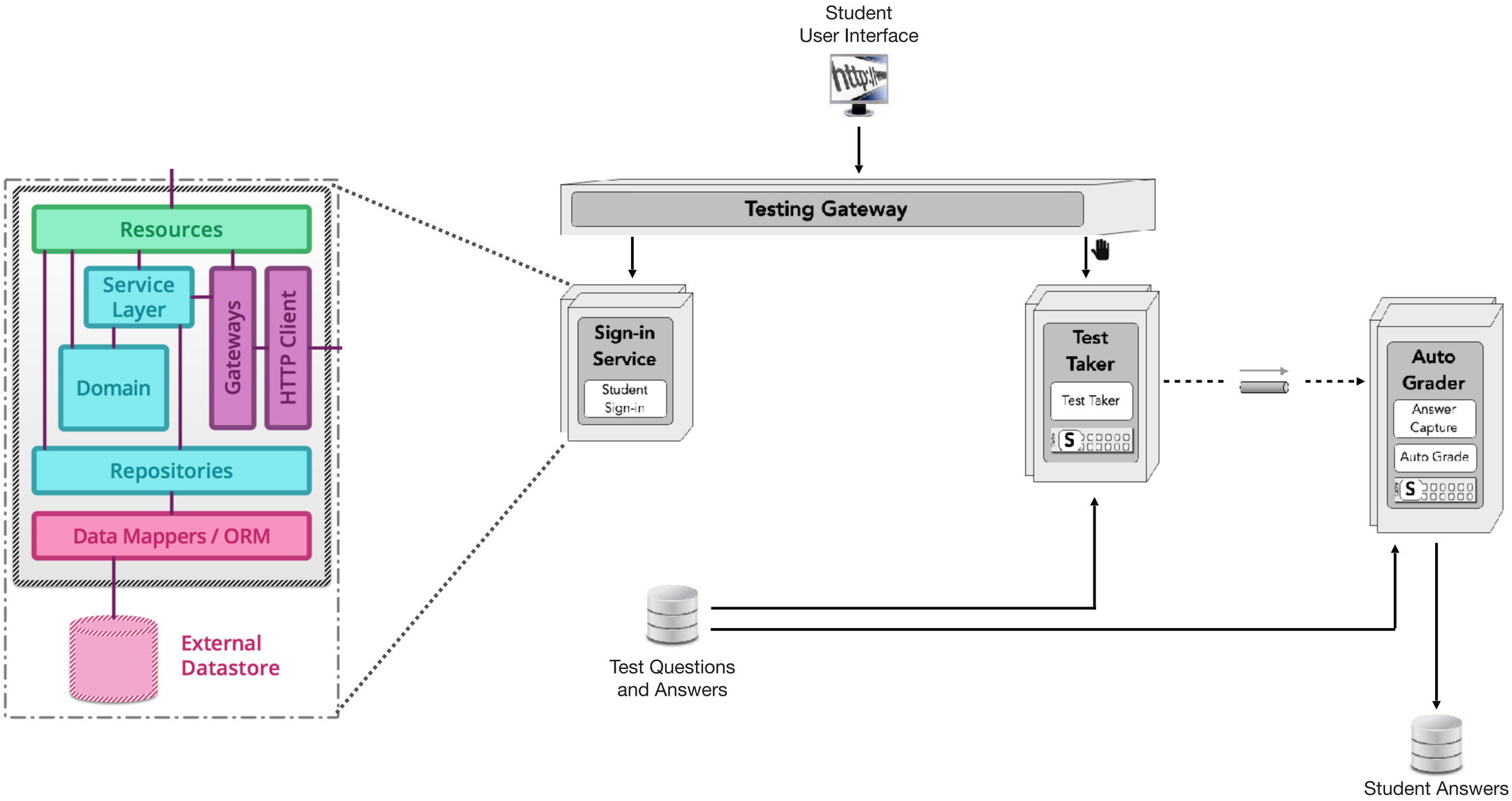
diagramming architecture



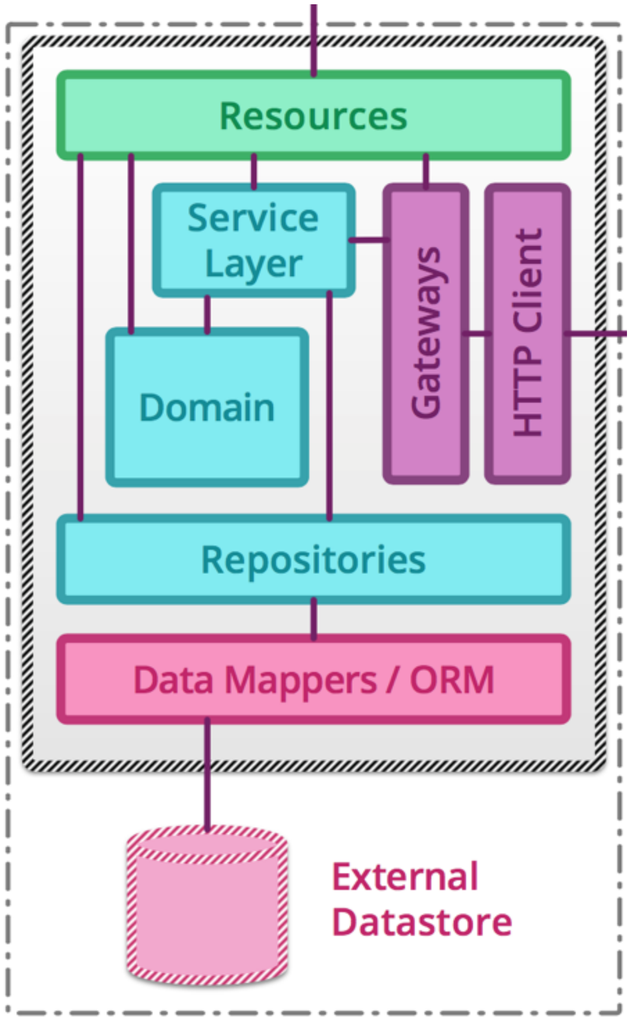
diagramming architecture



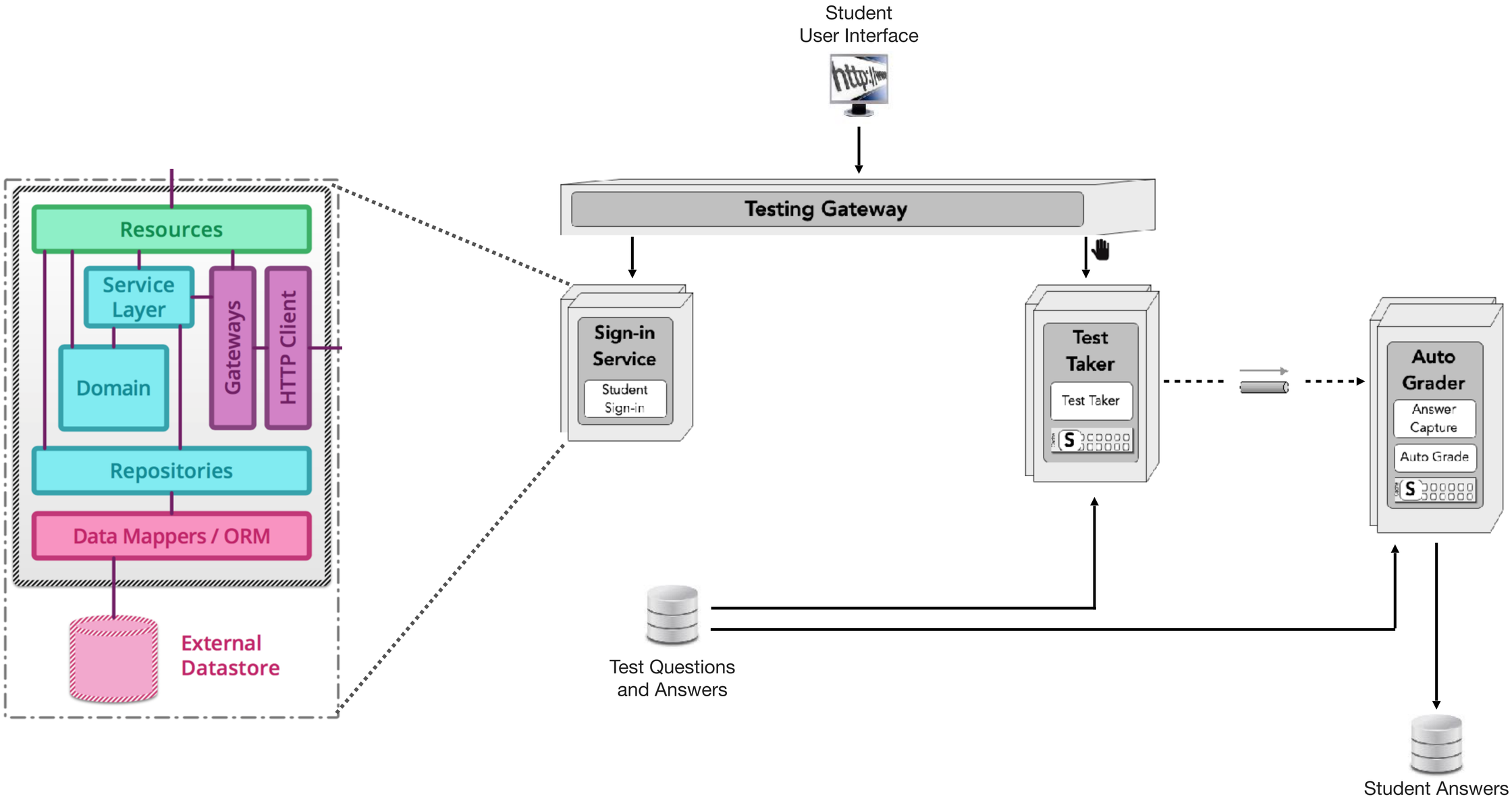
diagramming architecture



diagramming architecture



diagramming architecture



diagramming architecture

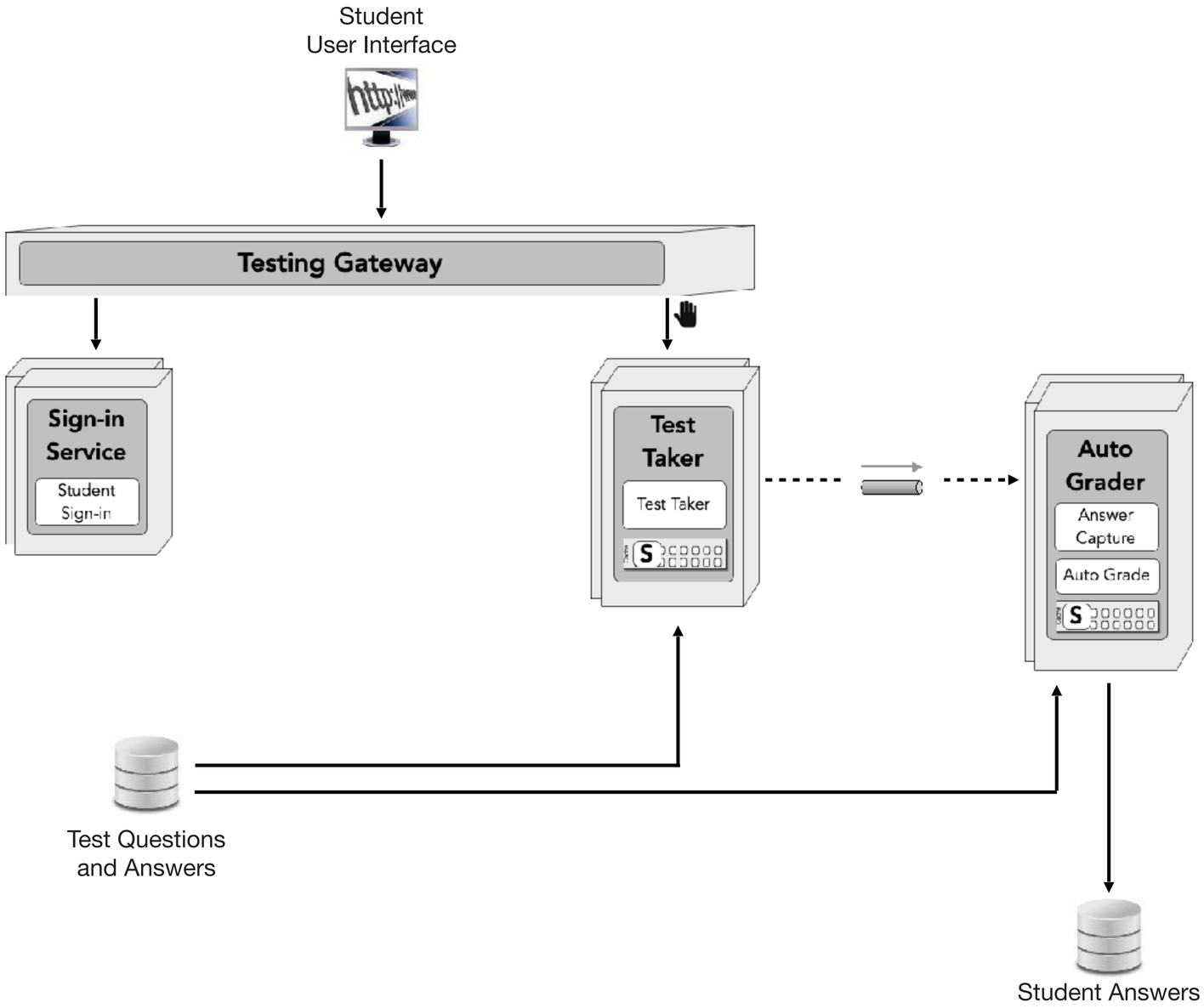
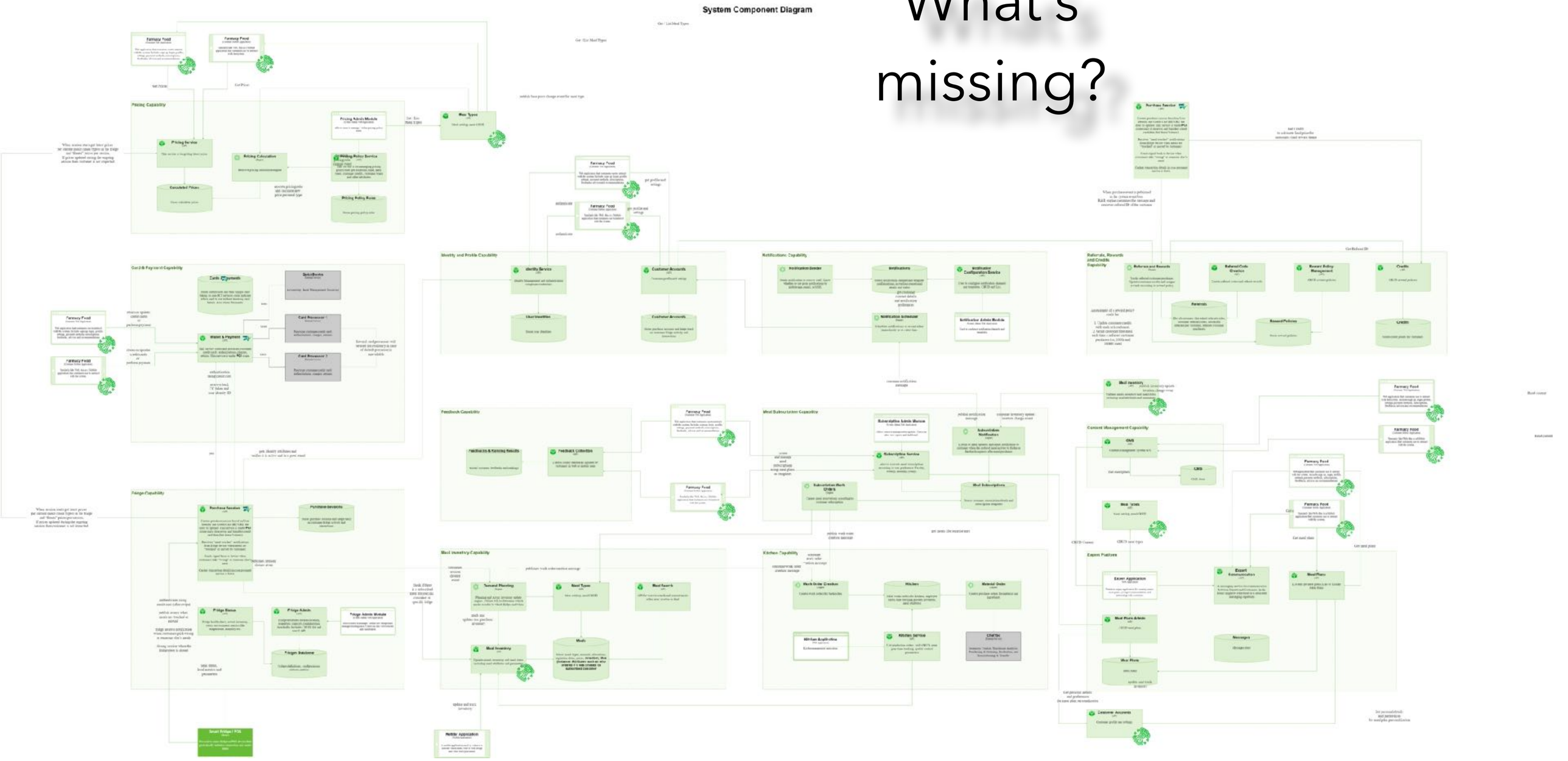


Diagram checklist

- [ ] Short, meaningful titles
- [ ] Lines: descriptions
- [ ] Lines: unidirectional
- [ ] Lines: synchronous or asynchronous
- [ ] Lines: blocking or non-blocking
- [ ] Shapes: consistency
- [ ] Avoid acronyms
- [ ] Color check for consistency, contrast, clarity
- [ ] Most important thing(s) centered
- [ ] Includes key

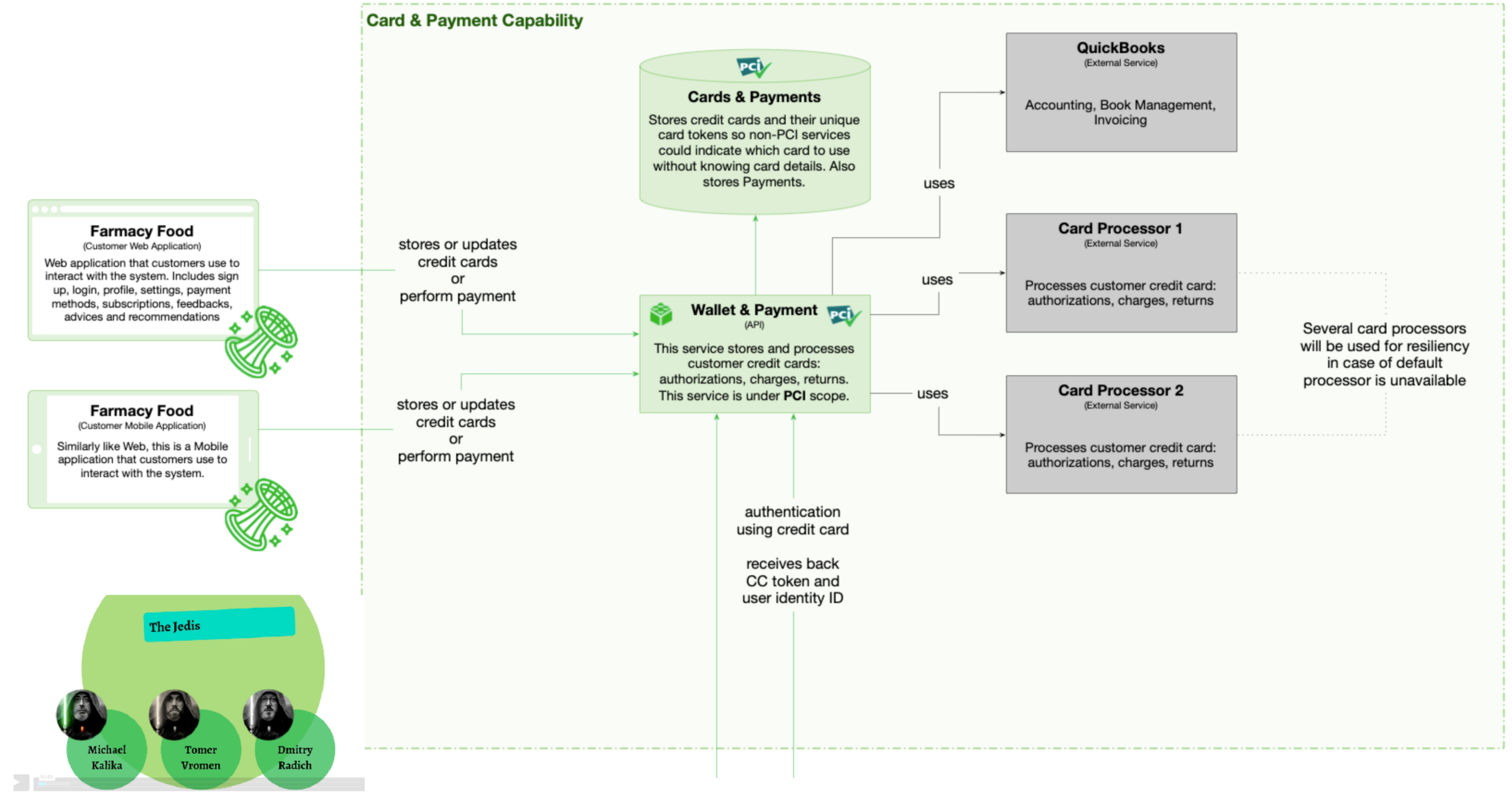
What's missing?

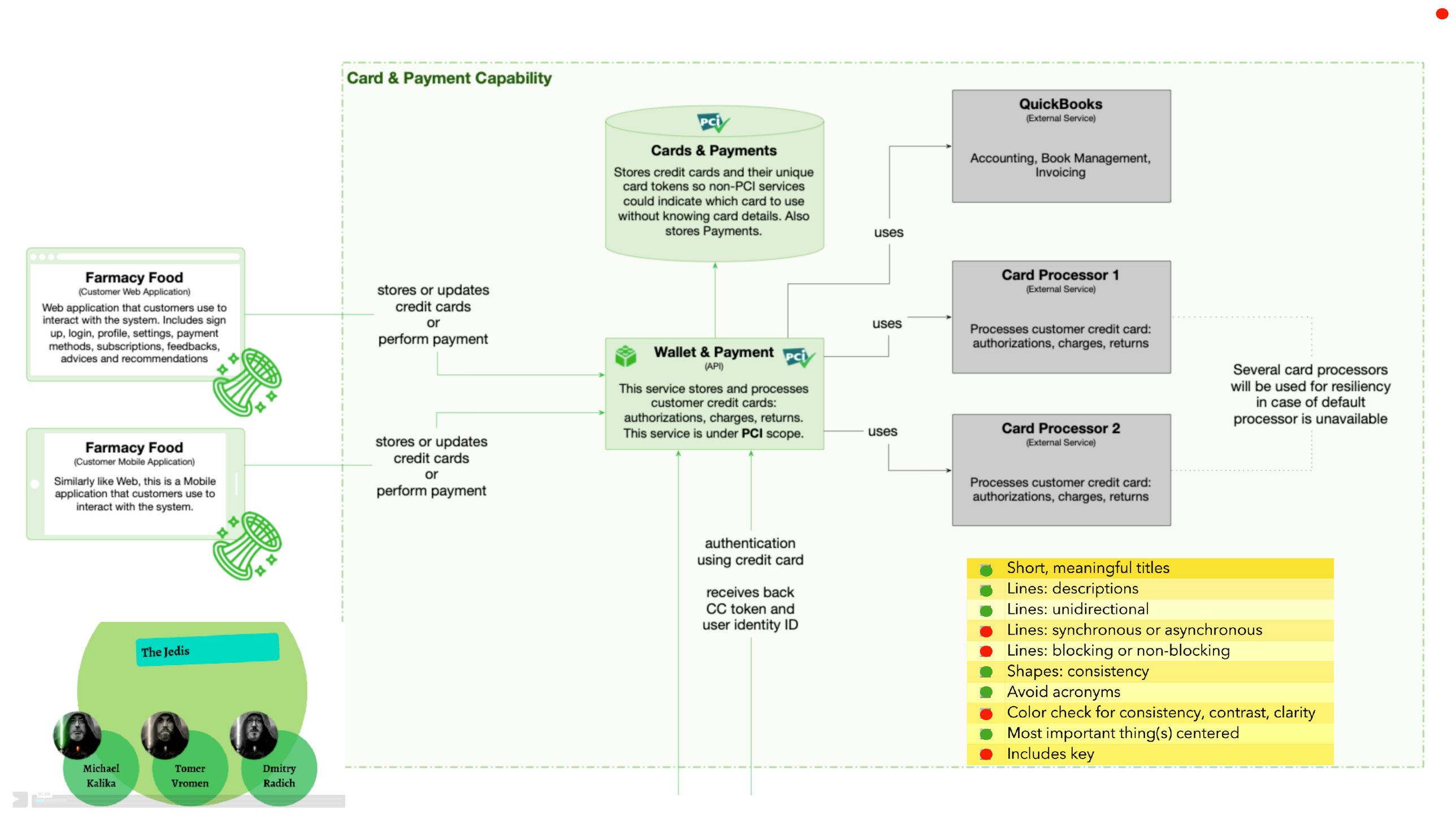


- Short, meaningful titles
- Lines: descriptions
- Lines: unidirectional
- Lines: synchronous or asynchronous
- Lines: blocking or non-blocking
- Shapes: consistency
- Avoid acronyms
- Color check for consistency, contrast, clarity
- Most important thing(s) centered
- Includes key

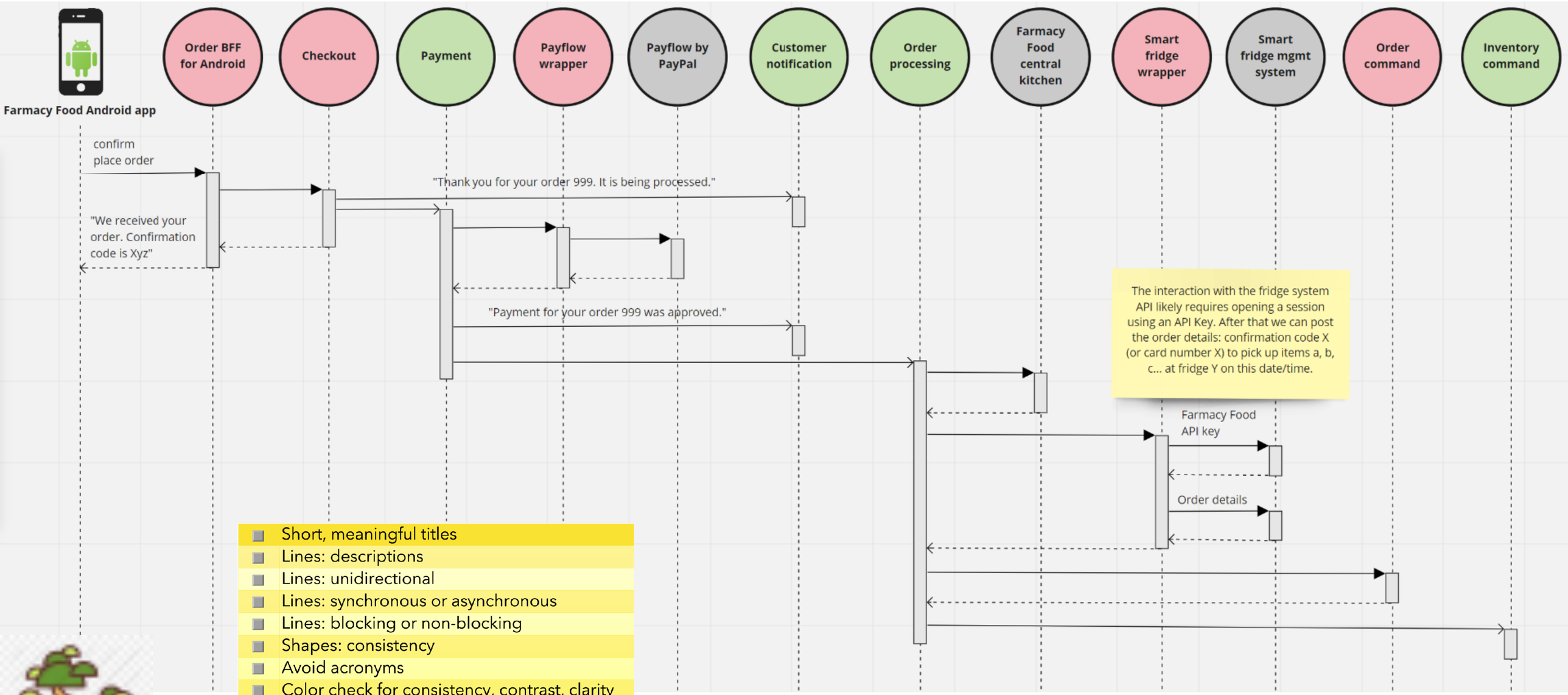
- Short, meaningful titles
- Lines: descriptions
- Lines: unidirectional
- Lines: synchronous or asynchronous
- Lines: blocking or non-blocking
- Shapes: consistency
- Avoid acronyms
- Color check for consistency, contrast, clarity
- Most important thing(s) centered
- Includes key

Architecture Diagram Types


Card & Payment Capability
Cards & Payments
Stores credit cards and their unique card tokens so non-PCI services could indicate which card to use without knowing card details. Also stores Payments.
QuickBooks
(External Service)
Accounting, Book Management, Invoicing
uses
Card Processor 1
(External Service)
Processes customer credit card: authorizations, charges, returns
uses
Several card processors will be used for resiliency in case of default processor is unavailable
Farmacy Food
(Customer Web Application)
Web application that customers use to interact with the system. Includes sign up, login, profile, settings, payment methods, subscriptions, feedbacks, advices and recommendations
stores or updates credit cards or perform payment
Wallet & Payment
(API)
This service stores and processes customer credit cards: authorizations, charges, returns. This service is under PCI scope.
uses
Card Processor 2
(External Service)
Processes customer credit card: authorizations, charges, returns
Farmacy Food
(Customer Mobile Application)
Similarly like Web, this is a Mobile application that customers use to interact with the system.
stores or updates credit cards or perform payment
authentication using credit card
receives back CC token and user identity ID
The Jedis
Michael Kalika
Tomer Vromen
Dmitry Radich

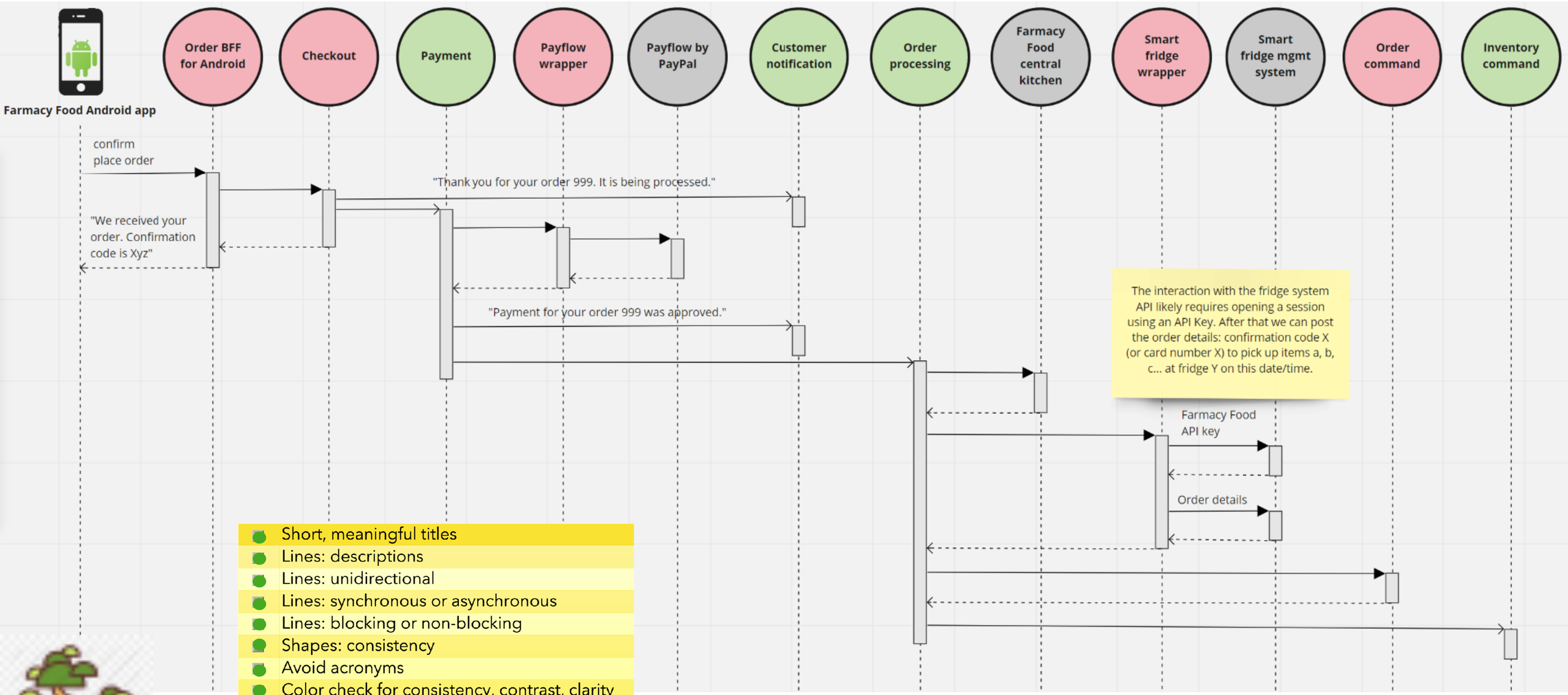

Card & Payment Capability
Cards & Payments
Stores credit cards and their unique card tokens so non-PCI services could indicate which card to use without knowing card details. Also stores Payments.
QuickBooks
(External Service)
Accounting, Book Management, Invoicing
uses
Card Processor 1
(External Service)
Processes customer credit card: authorizations, charges, returns
uses
Several card processors will be used for resiliency in case of default processor is unavailable
Farmacy Food
(Customer Web Application)
Web application that customers use to interact with the system. Includes sign up, login, profile, settings, payment methods, subscriptions, feedbacks, advices and recommendations
stores or updates credit cards or perform payment
Wallet & Payment
(API)
This service stores and processes customer credit cards: authorizations, charges, returns. This service is under PCI scope.
uses
Card Processor 2
(External Service)
Processes customer credit card: authorizations, charges, returns
Farmacy Food
(Customer Mobile Application)
Similarly like Web, this is a Mobile application that customers use to interact with the system.
stores or updates credit cards or perform payment
authentication using credit card
receives back CC token and user identity ID
Short, meaningful titles
Lines: descriptions
Lines: unidirectional
Lines: synchronous or asynchronous
Lines: blocking or non-blocking
Shapes: consistency
Avoid acronyms
Color check for consistency, contrast, clarity
Most important thing(s) centered
Includes key
The Jedis
Michael Kalika
Tomer Vromen
Dmitry Radich

Behavior



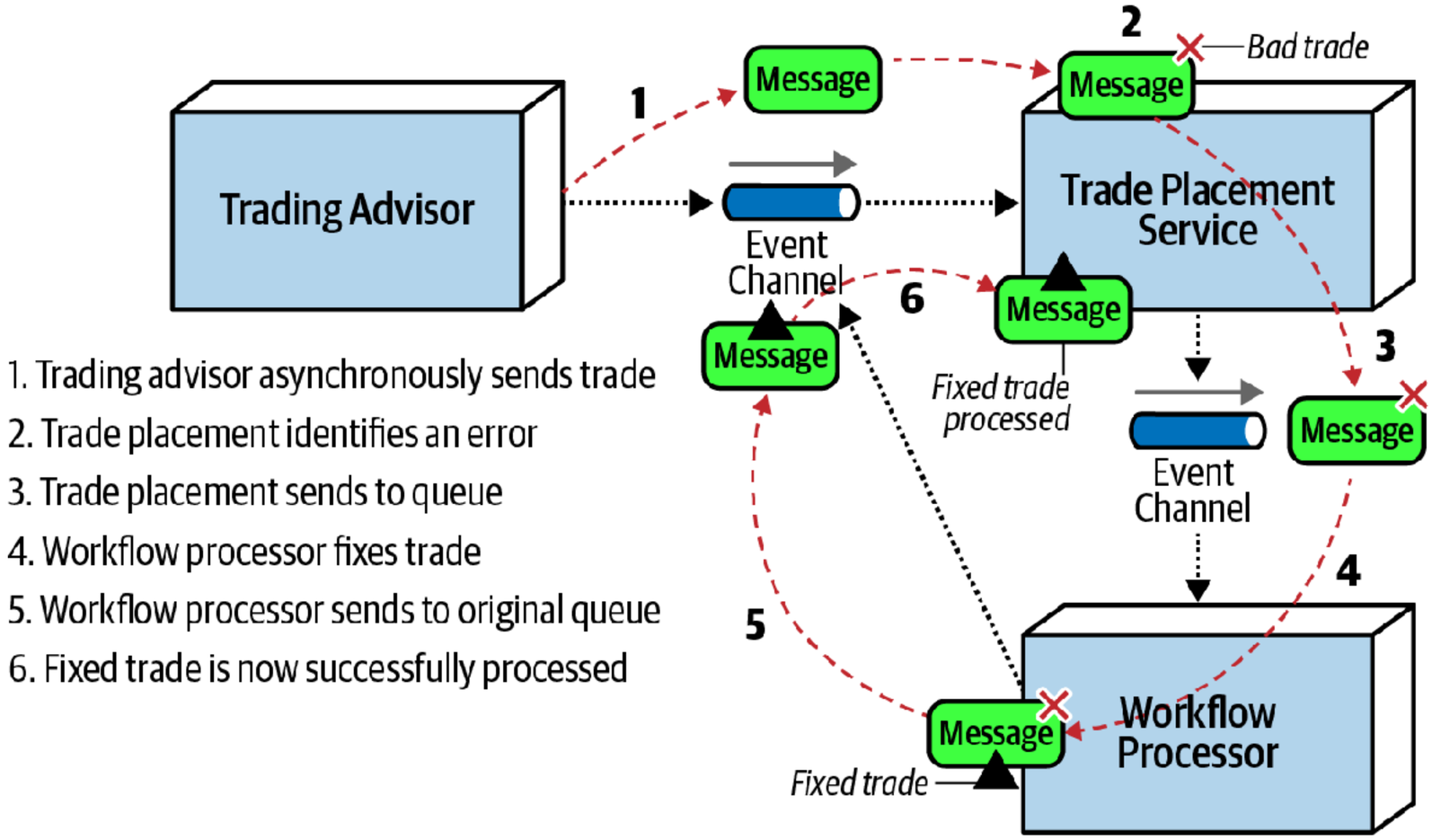
- Short, meaningful titles
- Lines: descriptions
- Lines: unidirectional
- Lines: synchronous or asynchronous
- Lines: blocking or non-blocking
- Shapes: consistency
- Avoid acronyms
- Color check for consistency, contrast, clarity
- Most important thing(s) centered
- Includes key

Behavior



- [ ] Short, meaningful titles
- [ ] Lines: descriptions
- [ ] Lines: unidirectional
- [ ] Lines: synchronous or asynchronous
- [ ] Lines: blocking or non-blocking
- [ ] Shapes: consistency
- [ ] Avoid acronyms
- [ ] Color check for consistency, contrast, clarity
- [ ] Most important thing(s) centered
- [ ] Includes key

structure + serial numbering



1. Trading advisor asynchronously sends trade
2. Trade placement identifies an error
3. Trade placement sends to queue
4. Workflow processor fixes trade
5. Workflow processor sends to original queue
6. Fixed trade is now successfully processed

Diagram Context and Flow

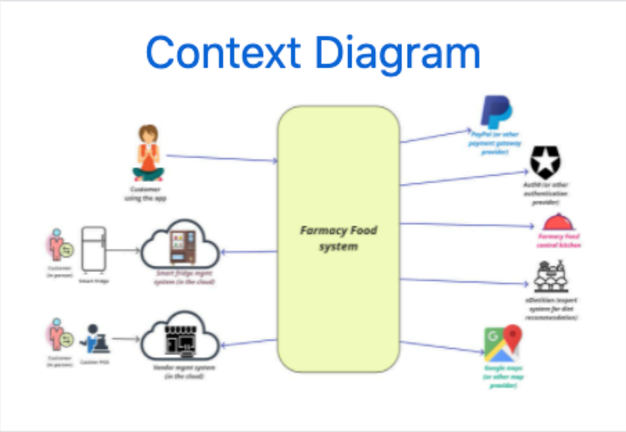
context diagram → views → bounded contexts → design → deployment

context diagram → views → bounded contexts → design → deployment

Architecture

Here you find the documentation of the software architecture that we envision to address Farmacy Food's requirements.

As a starting point, there's a context diagram that gives an overview of the external elements that interact with what we called the *Farmacy Food System*, which is the scope of this software architecture.



context diagram → **views** → bounded contexts → design → deployment

The main part of the software architecture is the set of five *architecture views* seen below. These views provide a runtime perspective of the system, that is, they show the components and connectors that have runtime presence and altogether correspond to the main capabilities provided by the Farmacy Food system. They follow the microservice and EDA architecture styles.

User Account Management - microservice view

Catalog - microservice view

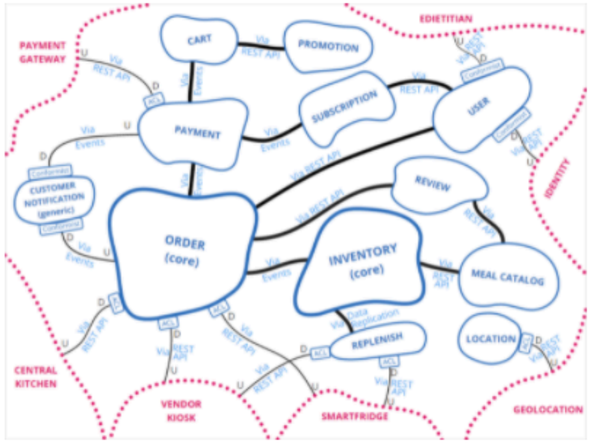
Order - microservice and EDA view

Customer at pick-up location - microservice and EDA view

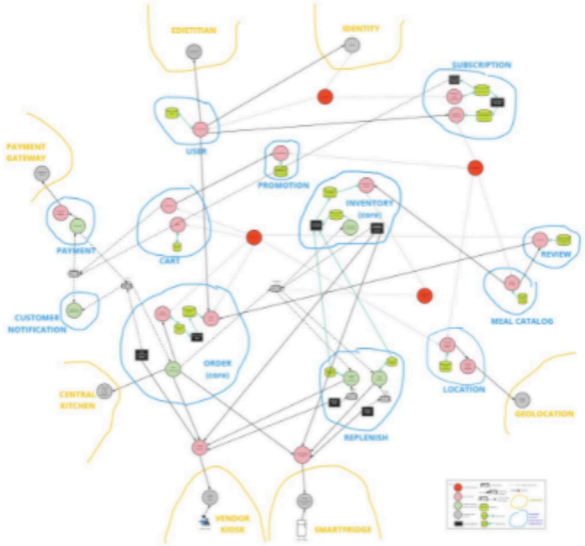
context diagram → views → **bounded contexts** → design → deployment

The following architecture view is a DDD Context Map. It shows how the Farmacy Food system is broken up into bounded contexts (BCs) and how they interact with each other.

DDD Context Map



DDD Context Map to microservices mapping



context diagram → views → bounded contexts → **design** → deployment

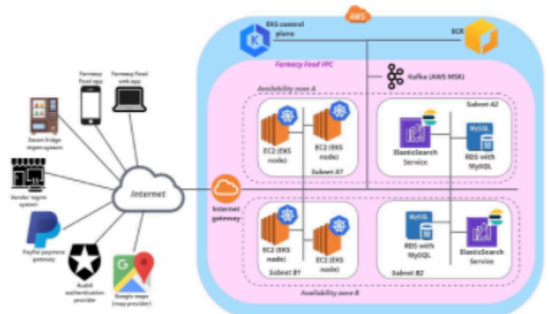
We believe the implementation of each BC can follow the hexagonal architecture. The following architecture describes how the code of each BC can be structured according to this architecture style.

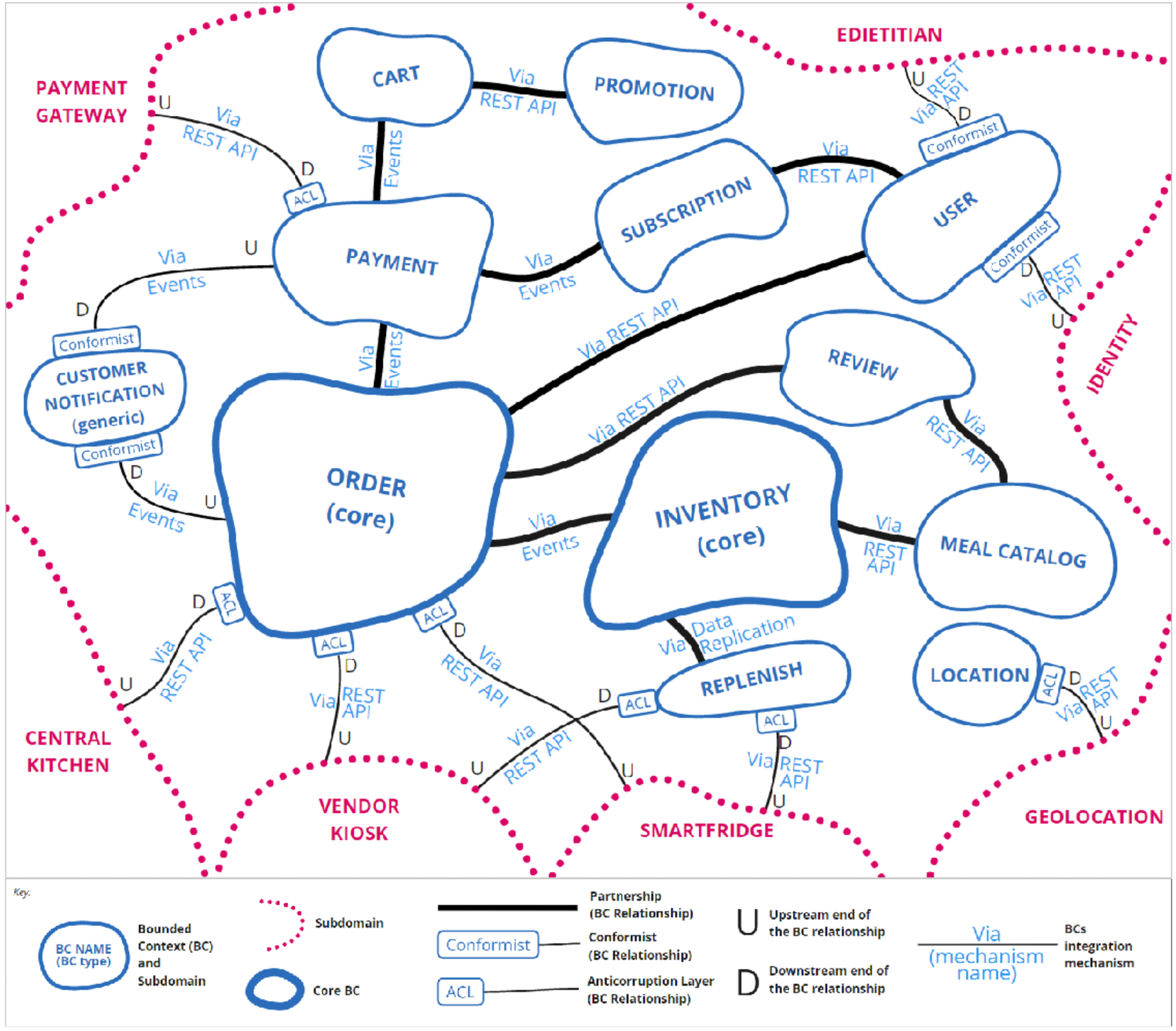
Hexagonal reference architeture view

context diagram → views → bounded contexts → design → deployment

Finally, we have a deployment view that describes the runtime infrastructure for the components seen in the five runtime architecture views.

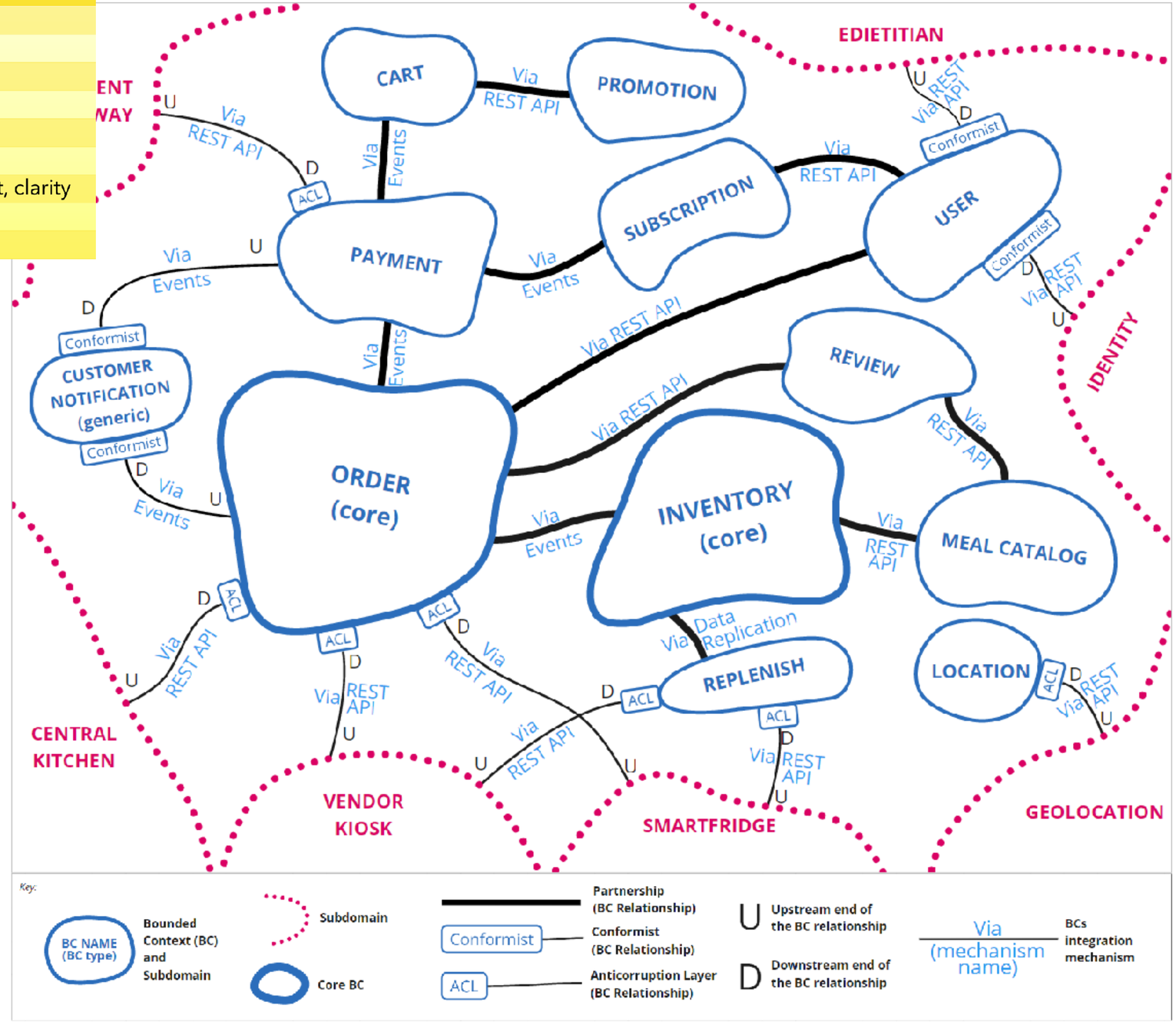
AWS deployment view




EDIETITIAN
PAYMENT GATEWAY
CART
PROMOTION
SUBSCRIPTION
USER
PAYMENT
CUSTOMER NOTIFICATION (generic)
ORDER (core)
REVIEW
INVENTORY (core)
MEAL CATALOG
REPLENISH
LOCATION
IDENTITY
CENTRAL KITCHEN
VENDOR KIOSK
SMARTFRIDGE
GEOLOCATION
Via REST API
Via Events
Via Data Replication
Via REST API
Conformist
ACL
U
D
Key:
BC NAME (BC type)
Bounded Context (BC) and Subdomain
Subdomain
Core BC
Partnership (BC Relationship)
Conformist (BC Relationship)
Anticorruption Layer (BC Relationship)
Upstream end of the BC relationship
Downstream end of the BC relationship
Via (mechanism name)
BCs integration mechanism

DDD Bounded Context

- Short, meaningful titles
- Lines: descriptions
- Lines: unidirectional
- Lines: synchronous or asynchronous
- Lines: blocking or non-blocking
- Shapes: consistency
- Avoid acronyms
- Color check for consistency, contrast, clarity
- Most important thing(s) centered
- Includes key



architecture diagrams

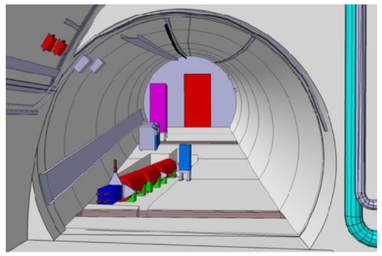
YOU BE THE JUDGE!

Which of these sets of architecture diagrams is best illustrates the architecture solution and why?

Farmacy Food System Architecture Design

Superkings

https://github.com/lastlegion/arch-katas

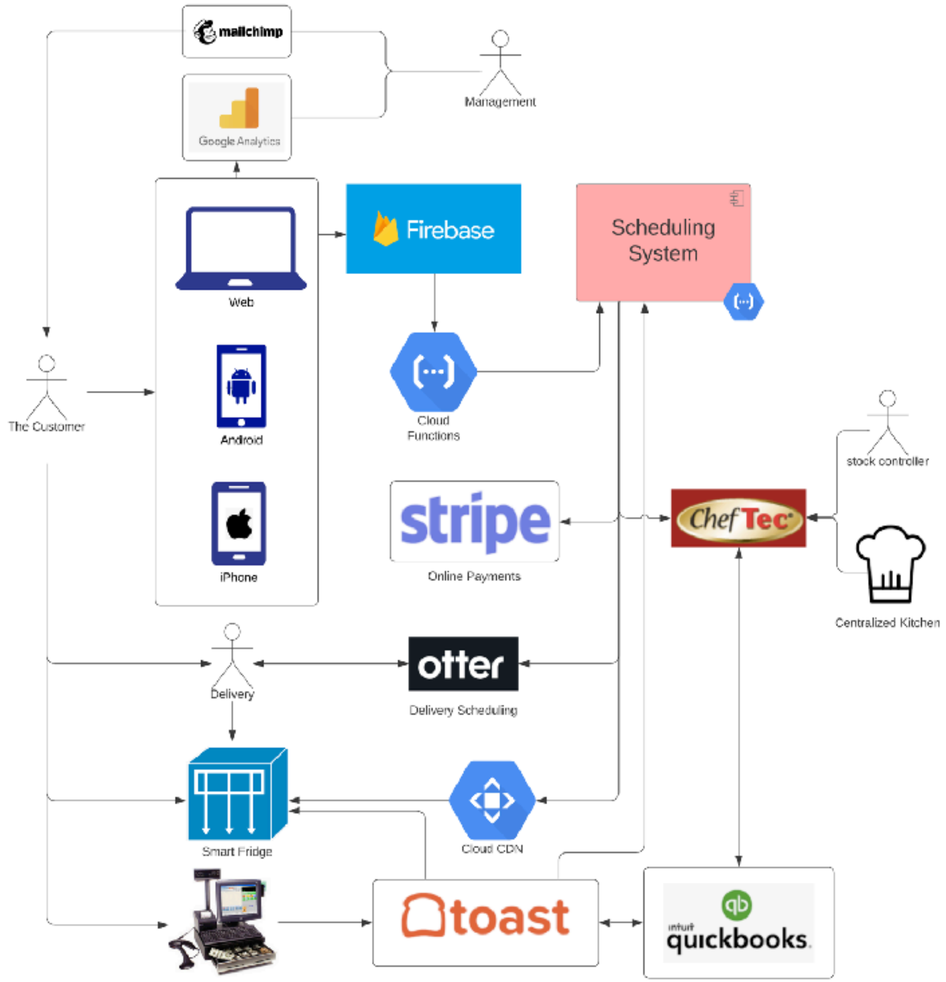


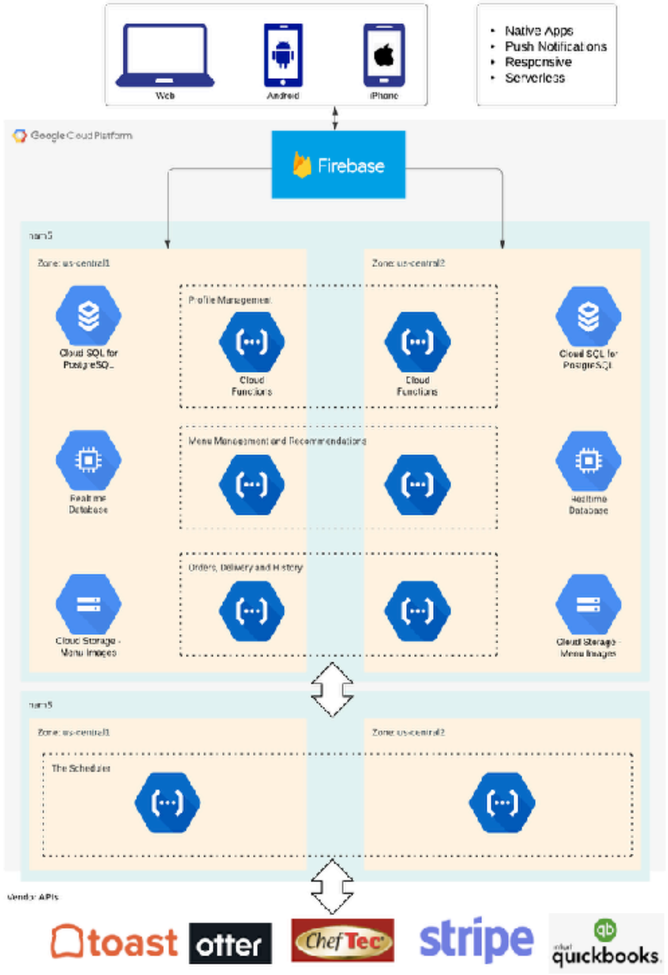
Arcolider

https://github.com/ldynia/archcolider

Architecture Solutions

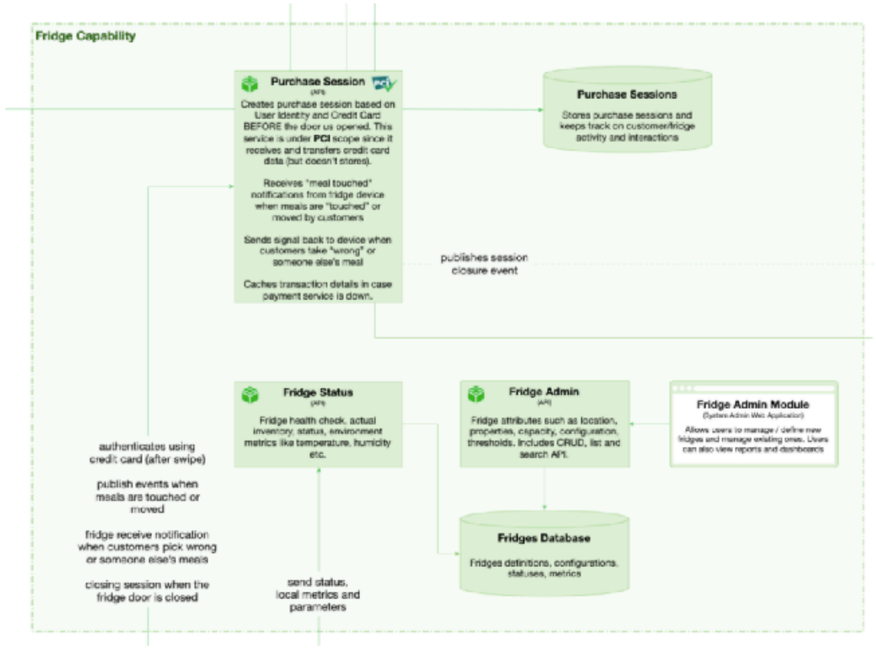
Is this a viable solution for Farmacy Foods?





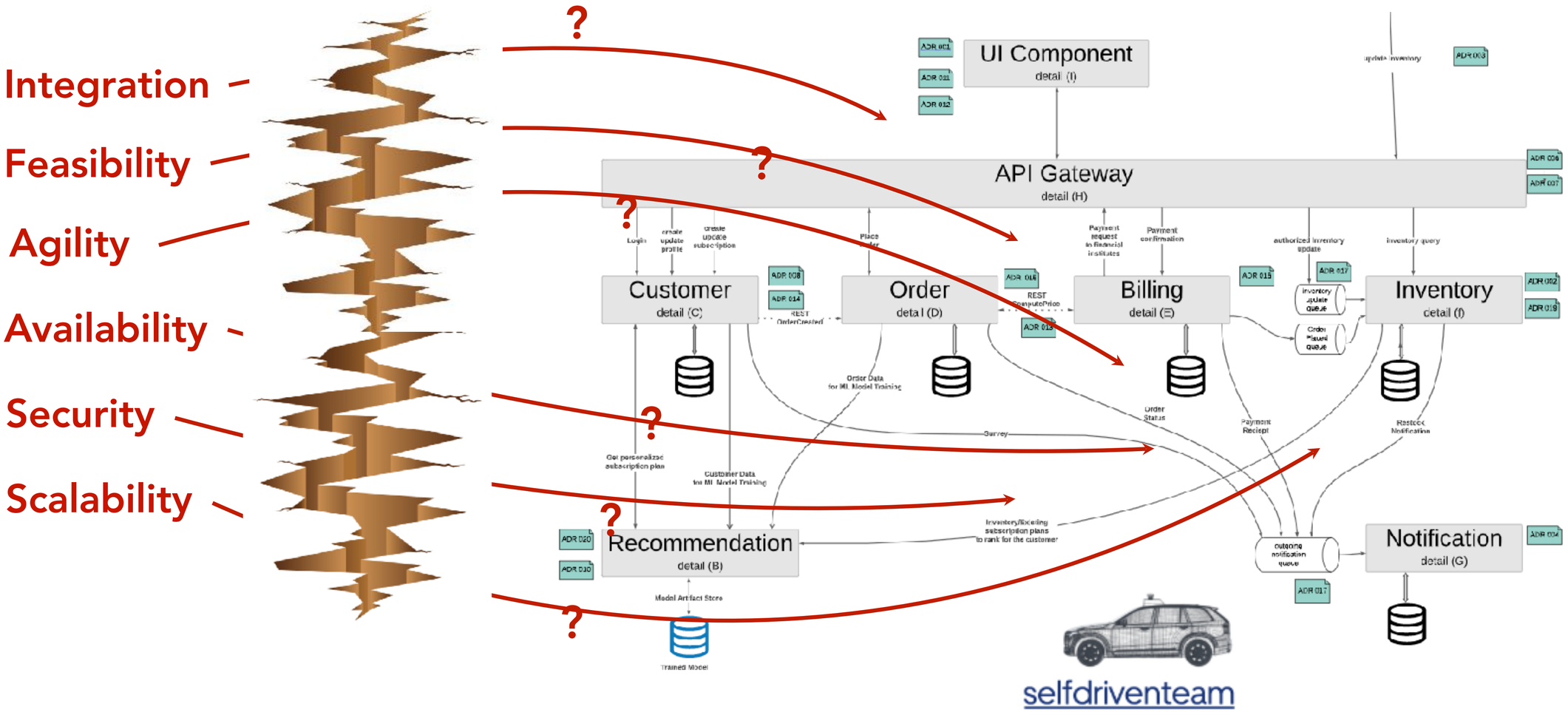
why is this criteria important?

The architecture solution describes the overall structure of the system and how it will be constructed



- [x] Are the architecture characteristics demonstrated in the solution?
- [x] Is the solution appropriate and feasible given the project constraints?
- [x] Are the architecture styles selected represented in the solution?

Are the architecture characteristics demonstrated in the solution?



Are the architecture characteristics demonstrated in the solution?

Integration

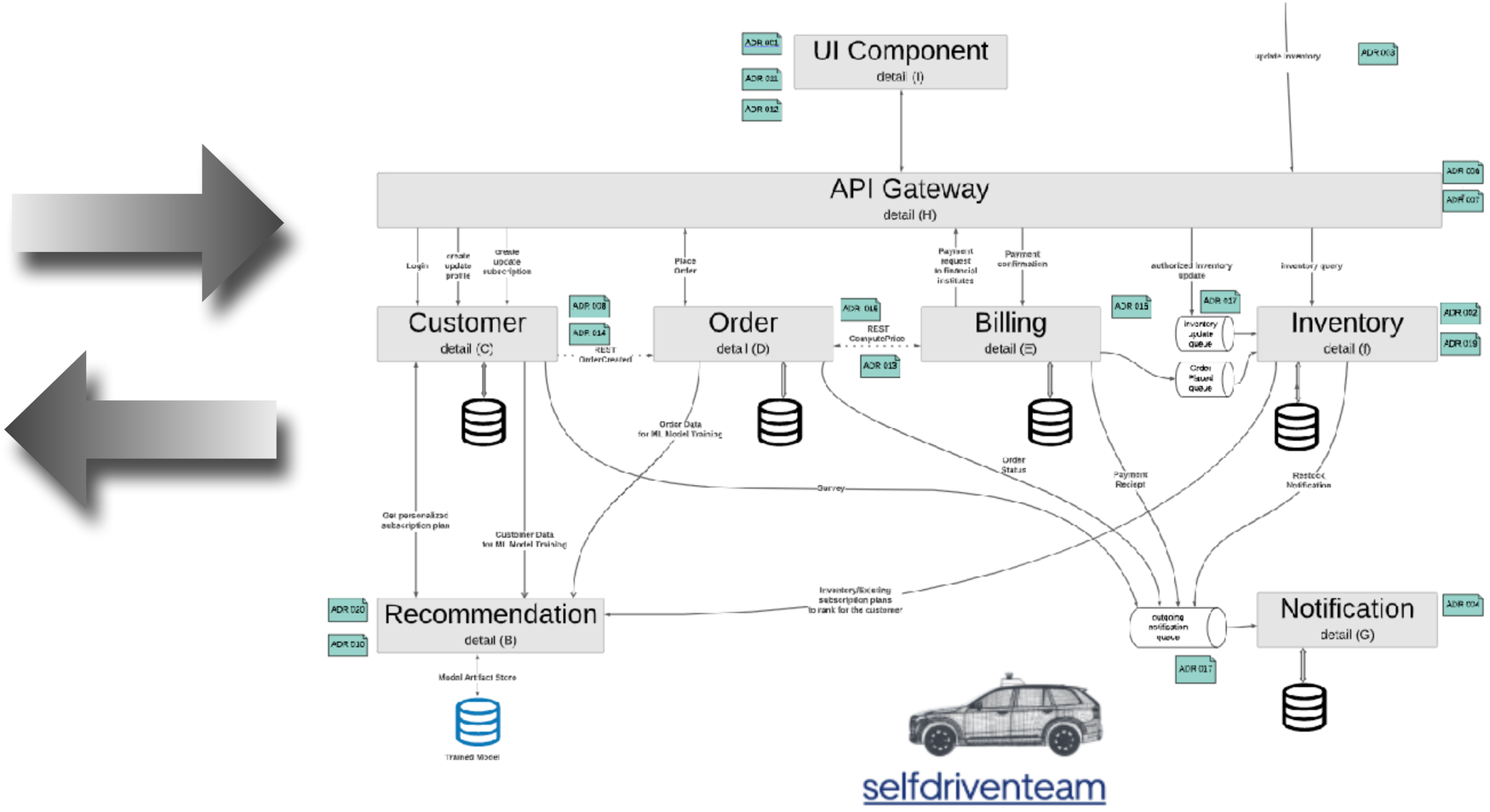
Feasibility

Agility

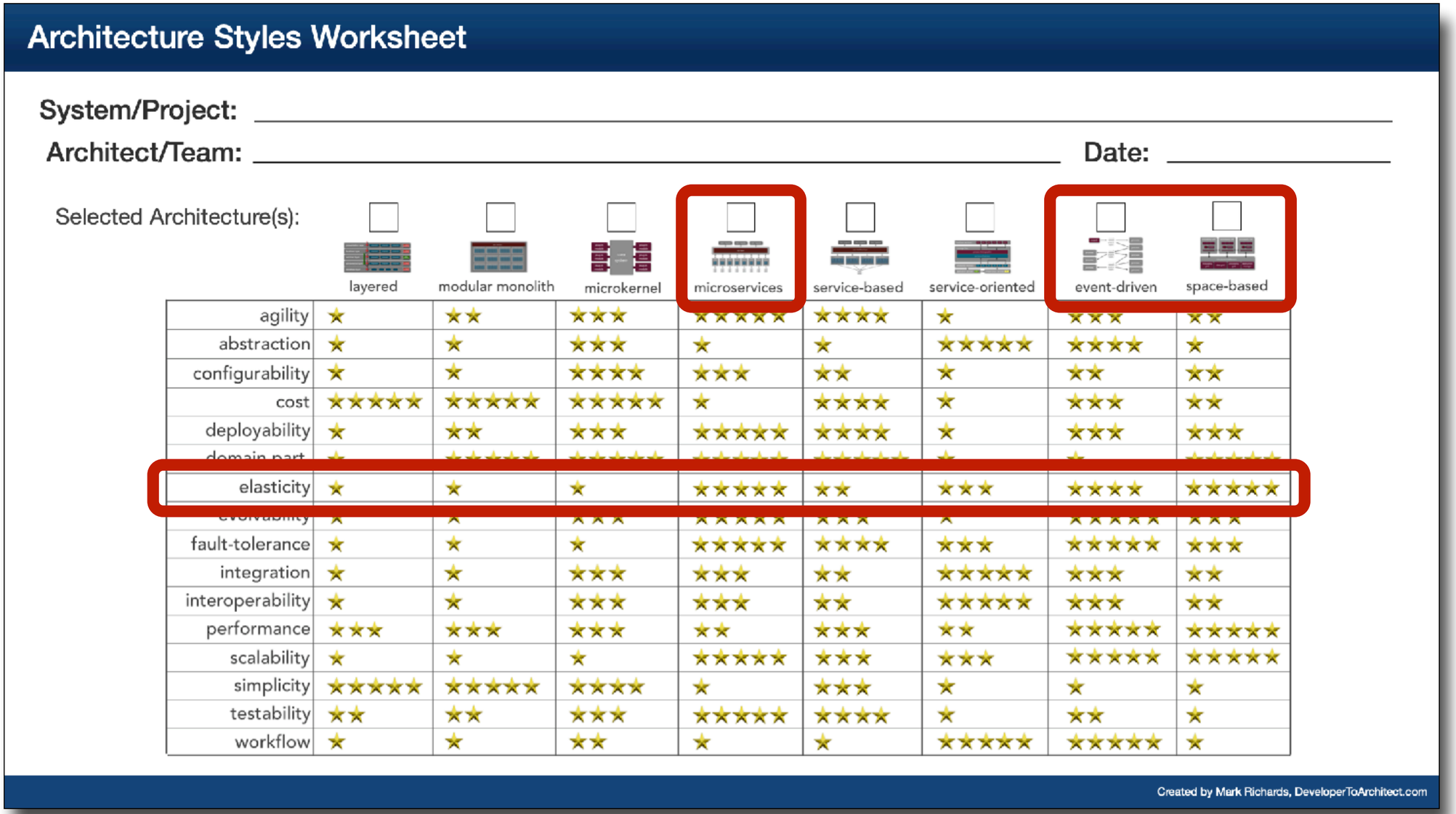
Availability

Security

Scalability



https://www.developertoarchitect.com/downloads/worksheets.html



Architecture Styles Worksheet

System/Project: _____

Architect/Team: _____ Date: _____

Selected Architecture(s):

	layered	modular monolith	microkernel	microservices	service-based	service-oriented	event-driven	space-based
agility	★	★★	★★★	★★★★★	★★★★	★	★★★	★★
abstraction	★	★	★★★	★	★	★★★★★	★★★★	★
configurability	★	★	★★★★	★★★	★★	★	★★	★★
cost	★★★★★	★★★★★	★★★★★	★	★★★★	★	★★★	★★
deployability	★	★★	★★★	★★★★★	★★★★	★	★★★	★★★
domain part	★	★★★★★	★★★★★	★★★★★	★★★★★	★	★	★★★★★
elasticity	★	★	★	★★★★★	★★	★★★	★★★★	★★★★★
evolvability	★	★	★★★	★★★★★	★★★	★	★★★★★	★★★
fault-tolerance	★	★	★	★★★★★	★★★★	★★★	★★★★★	★★★
integration	★	★	★★★	★★★	★★	★★★★★	★★★	★★
interoperability	★	★	★★★	★★★	★★	★★★★★	★★★	★★
performance	★★★	★★★	★★★	★★	★★★	★★	★★★★★	★★★★★
scalability	★	★	★	★★★★★	★★★	★★★	★★★★★	★★★★★
simplicity	★★★★★	★★★★★	★★★★	★	★★★	★	★	★
testability	★★	★★	★★★	★★★★★	★★★★	★	★★	★
workflow	★	★	★★	★	★	★★★★★	★★★★★	★

Created by Mark Richards, DeveloperToArchitect.com

Are the architecture characteristics demonstrated in the solution?

Agility

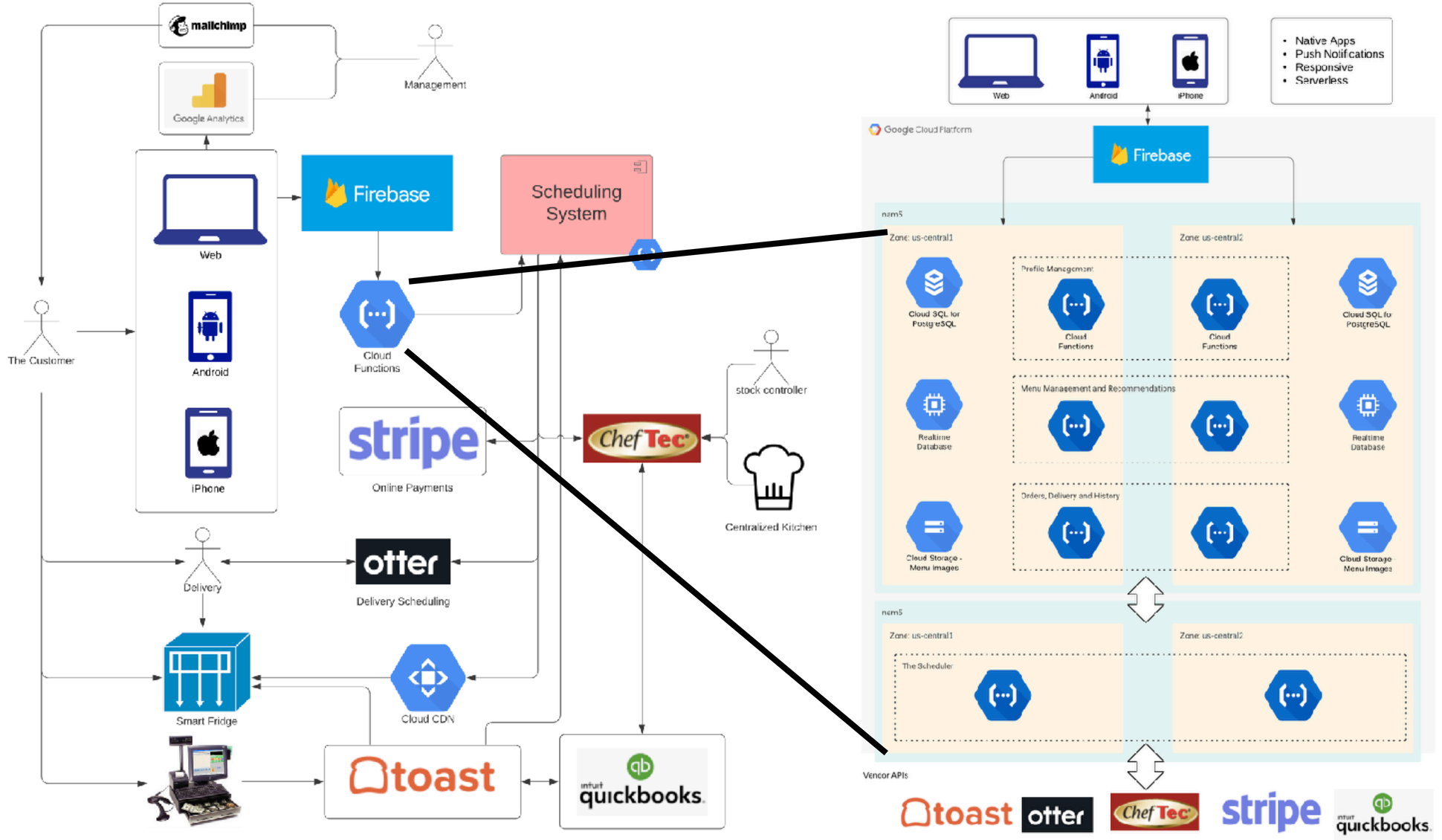
Viability

Flexibility

Availability

Security

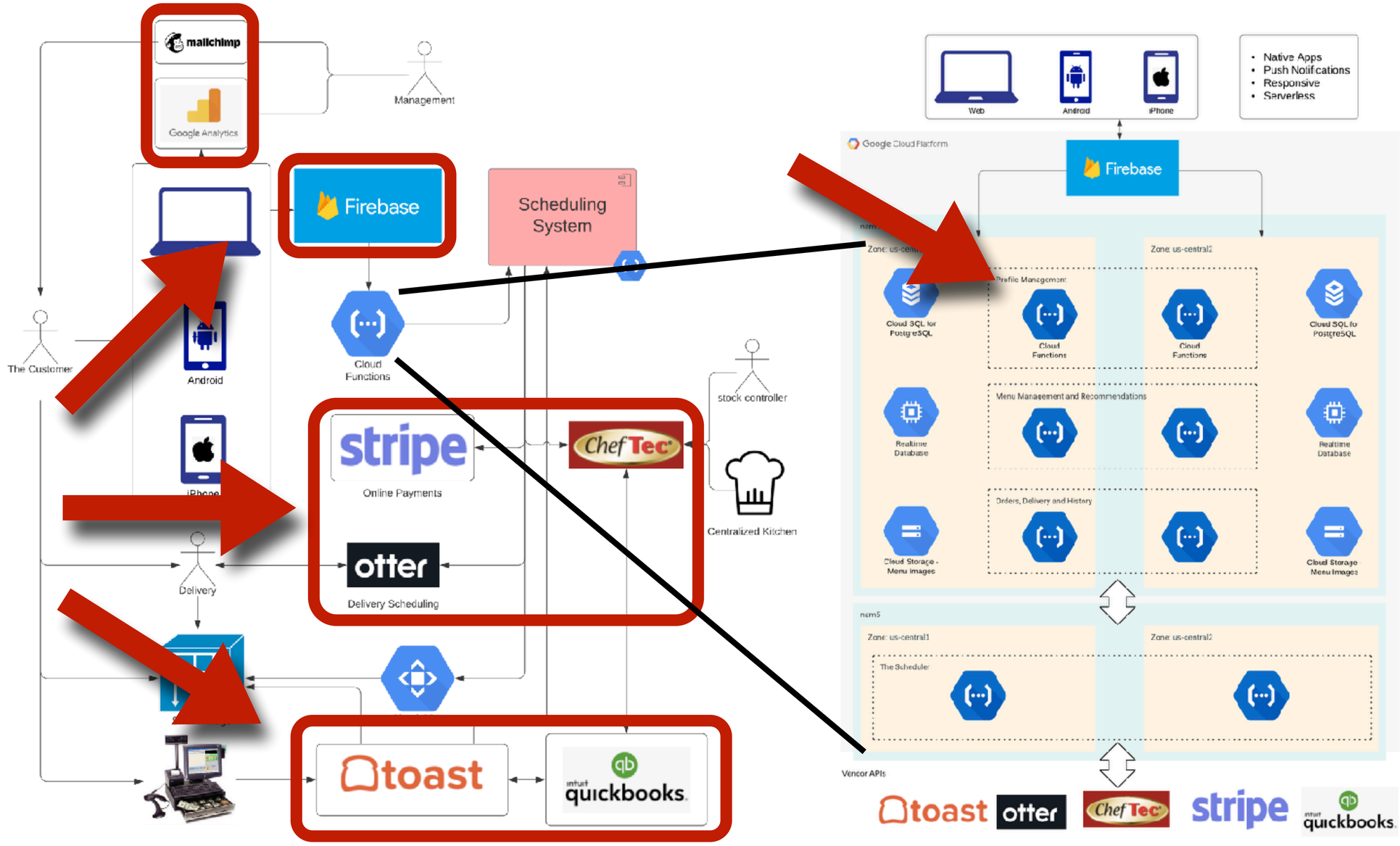
Scalability



Jiakaturi

Are the architecture characteristics demonstrated in the solution?

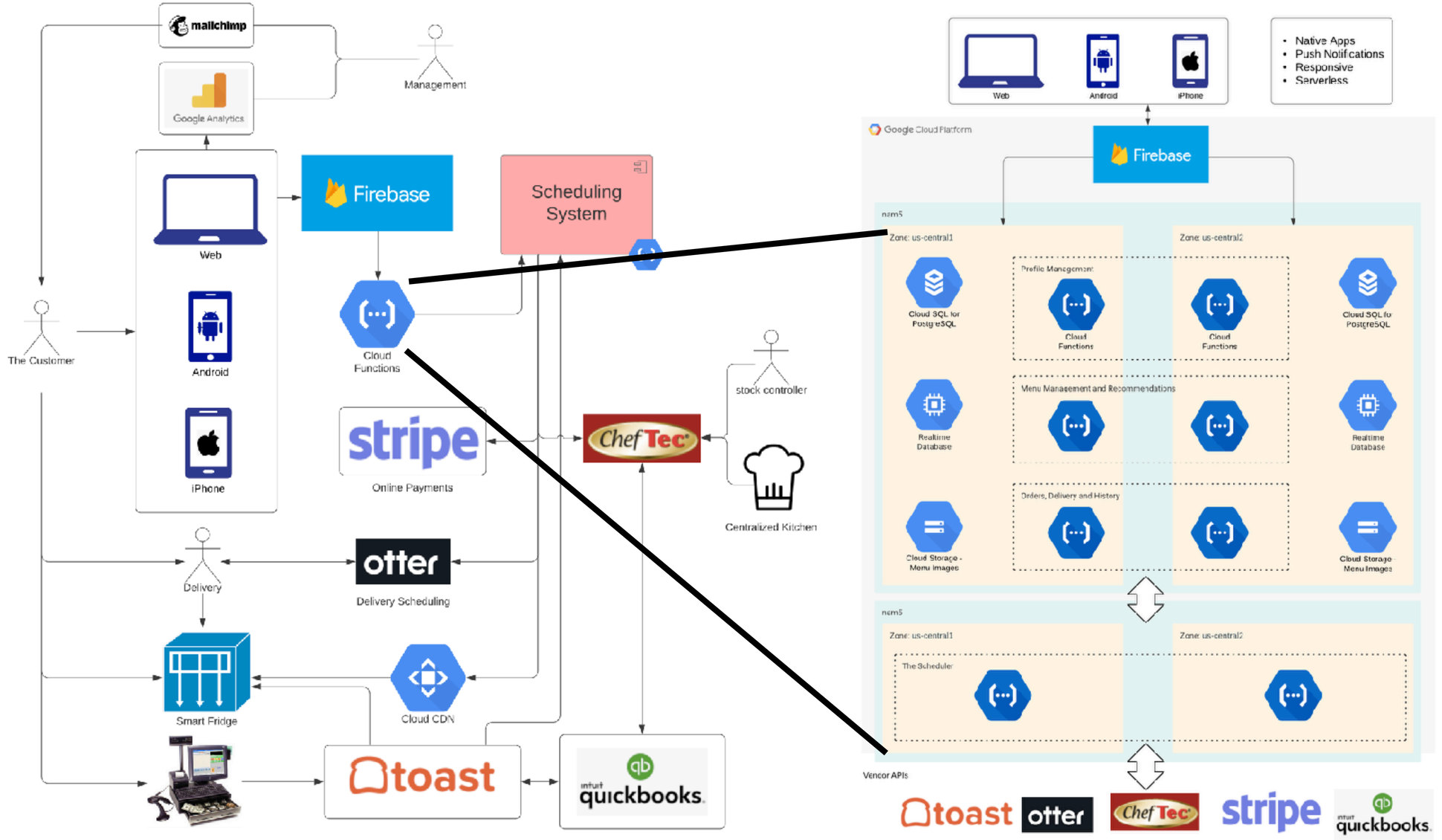
- ✓ Agility
- Viability
- Flexibility
- Availability
- Security
- Scalability



Jiakaturi

Are the architecture characteristics demonstrated in the solution?

- ✓ Agility
- Viability
- Flexibility
- ? Availability
- Security
- Scalability

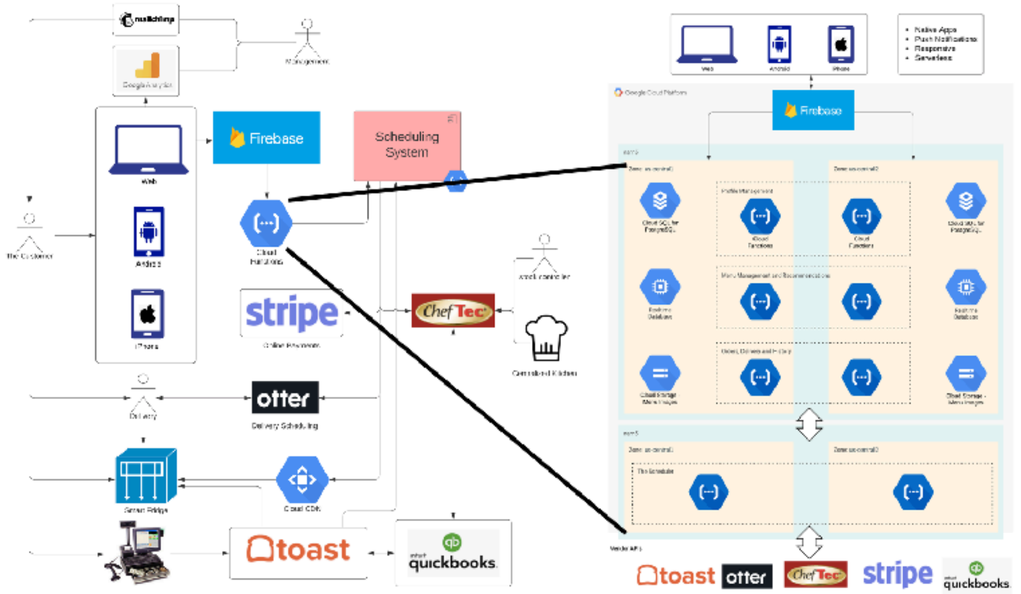


Jiakaturi

architecture solution

YOU BE THE JUDGE!

Do you think availability is demonstrated by the proposed solution by Jaikaturi?



Poll question:

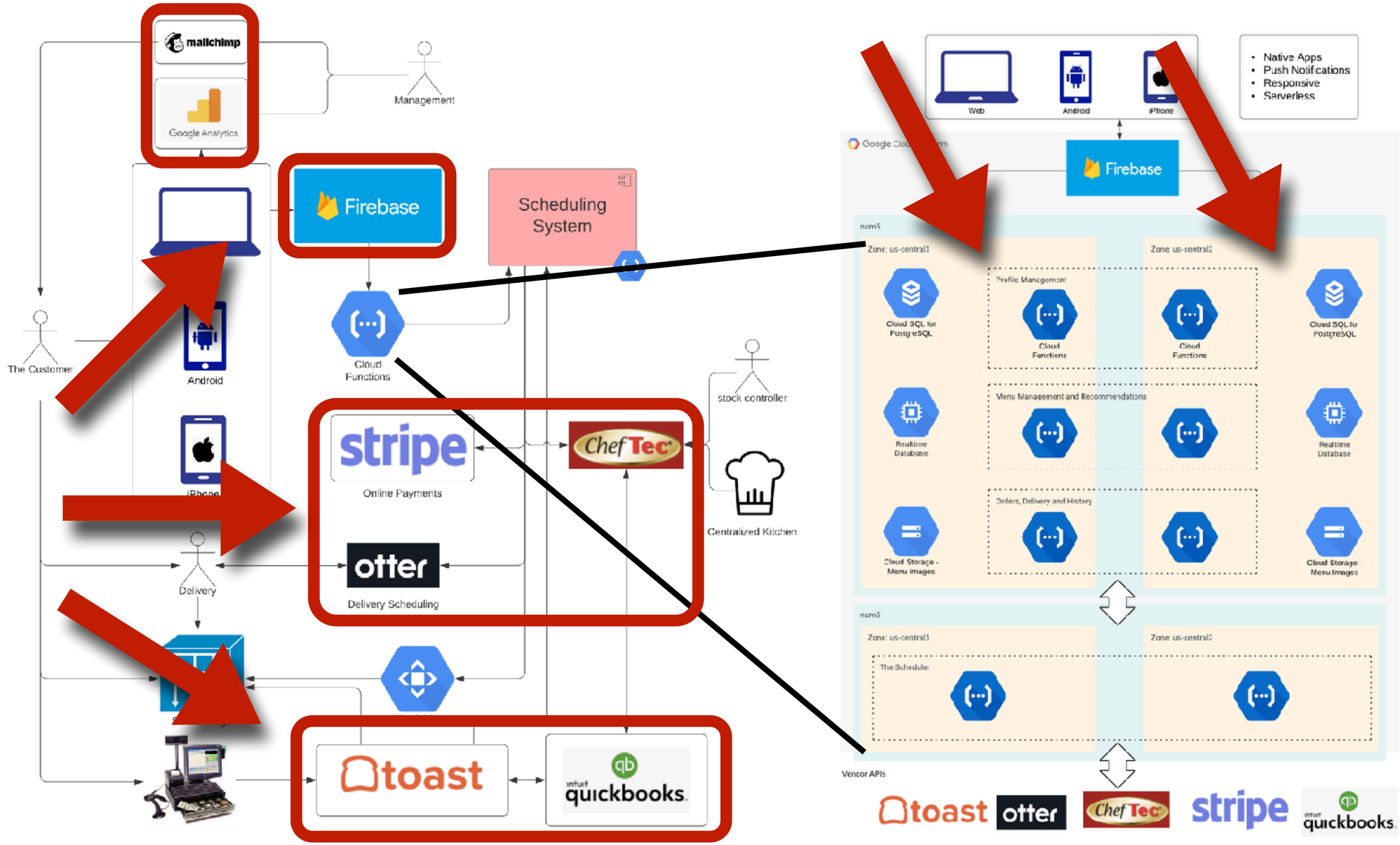
Do you think availability is demonstrated by the proposed architecture by Jaikaturi?

Yes, it satisfies availability requirements

No, it does not satisfy availability requirements

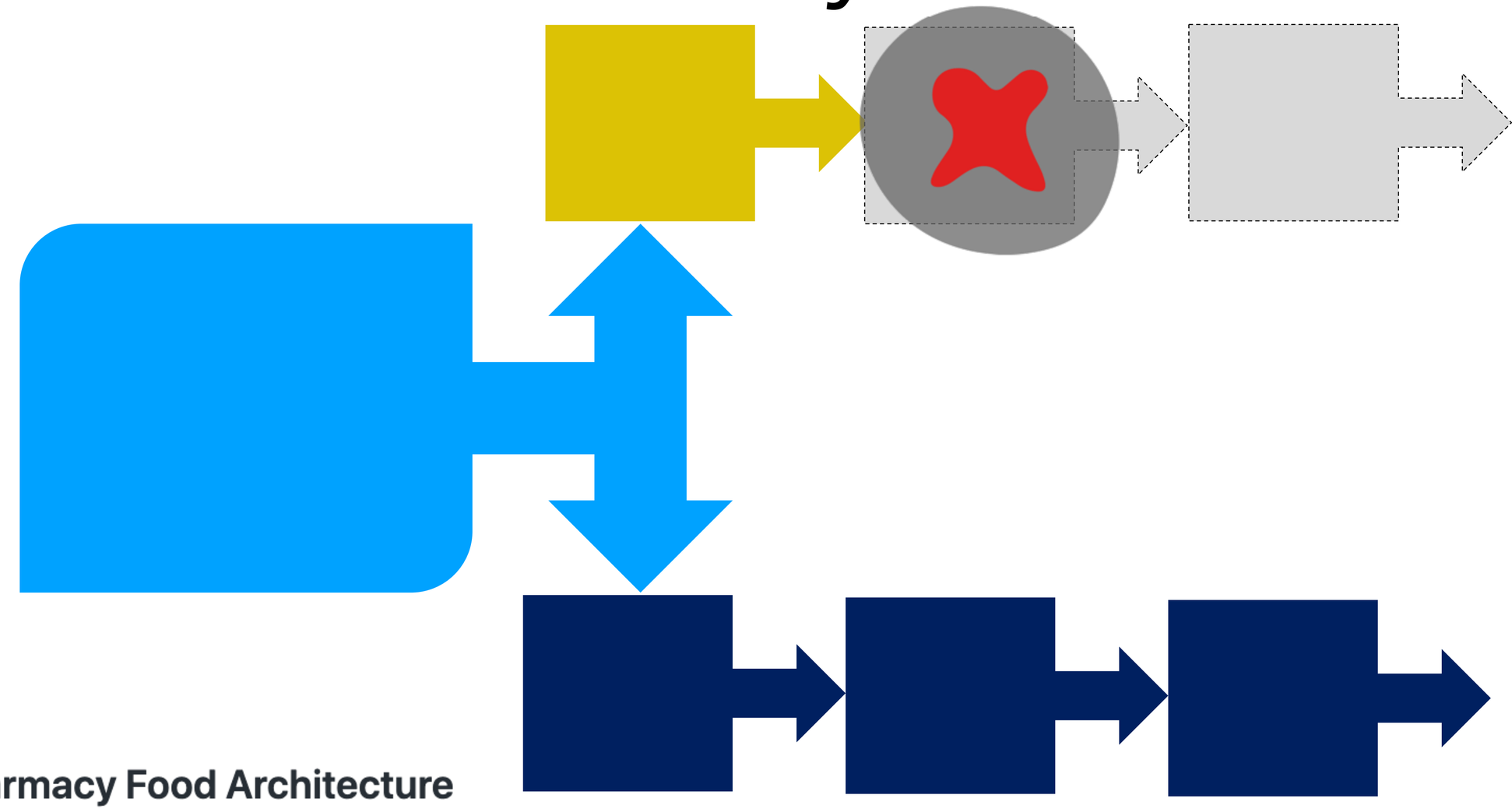
Are the architecture characteristics demonstrated in the solution?

- ✓ Agility
- Viability
- Flexibility
- ? Availability
- Security
- Scalability



Jiakaturi

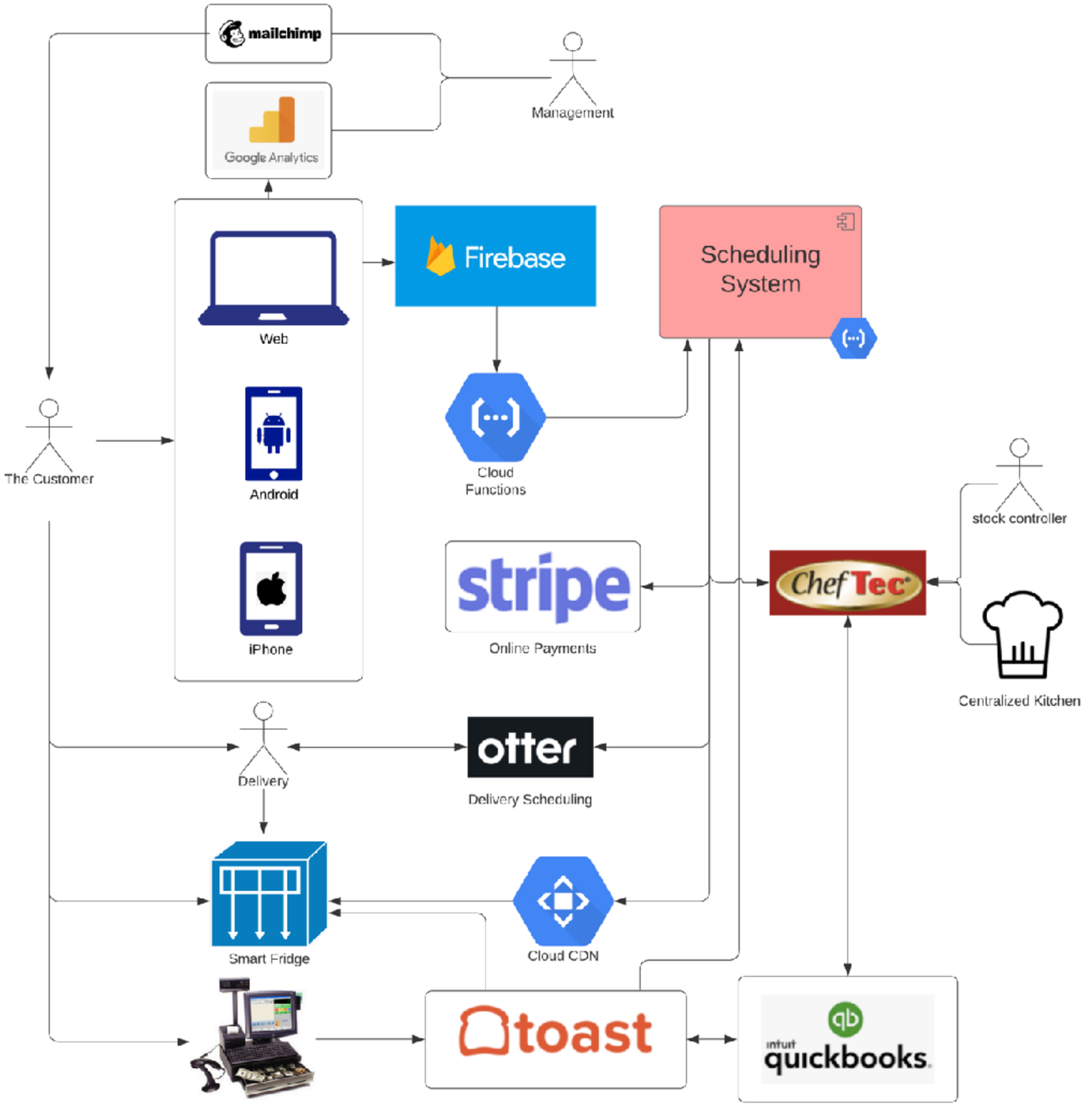
One-way Decisions

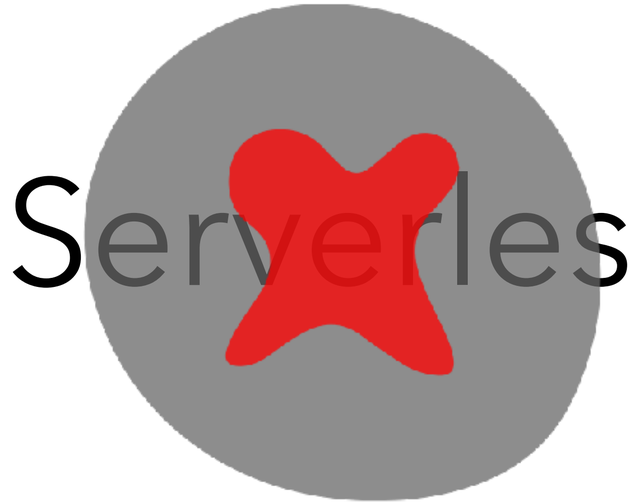


Farmacy Food Architecture

by Jiakaturi

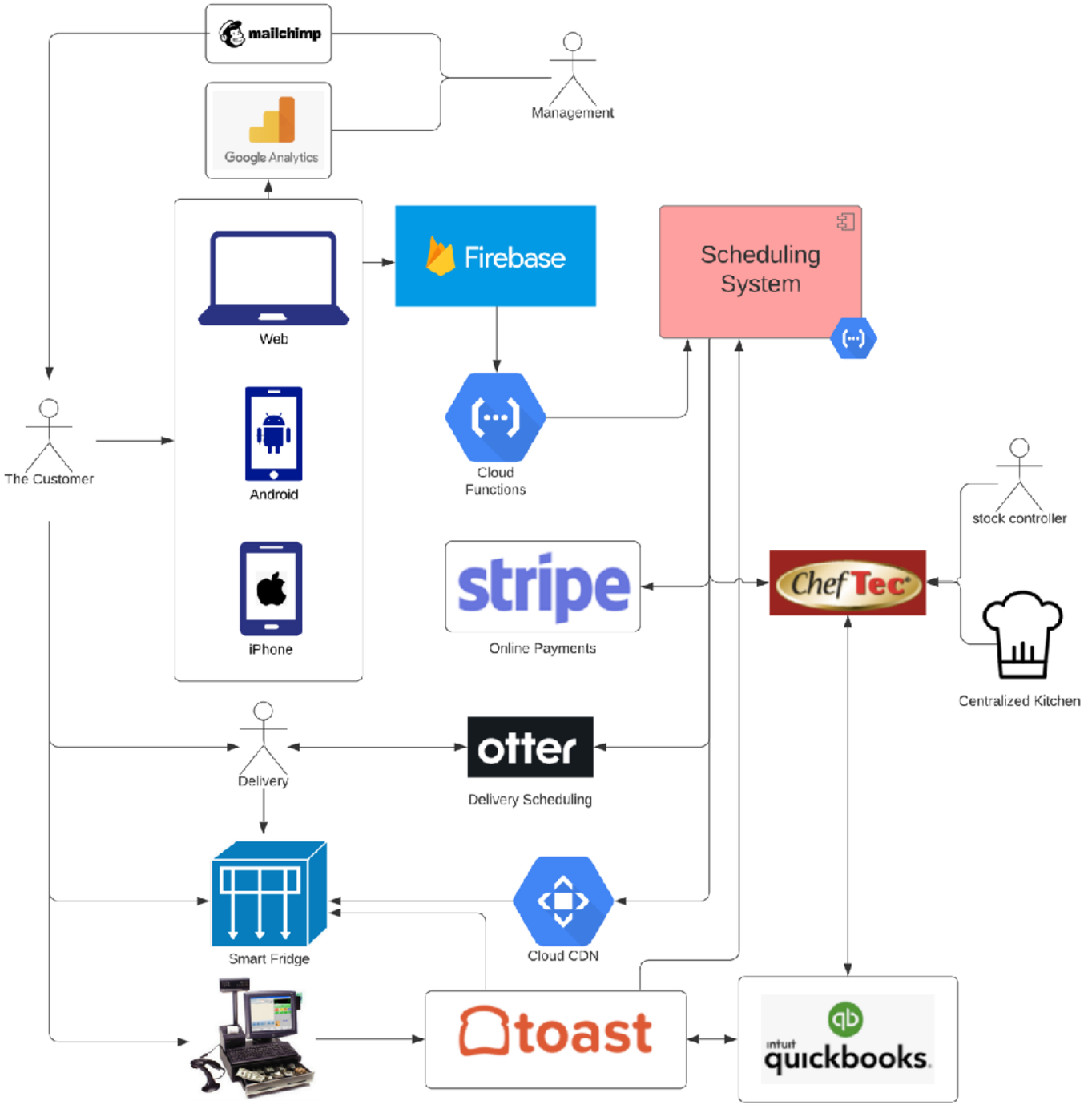
Serverless


mailchimp
Google Analytics
Management
Web
Firebase
Scheduling System
Android
Cloud Functions
The Customer
iPhone
stock controller
stripe
Online Payments
ChefTec
Centralized Kitchen
Delivery
otter
Delivery Scheduling
Smart Fridge
Cloud CDN
toast
intuit quickbooks

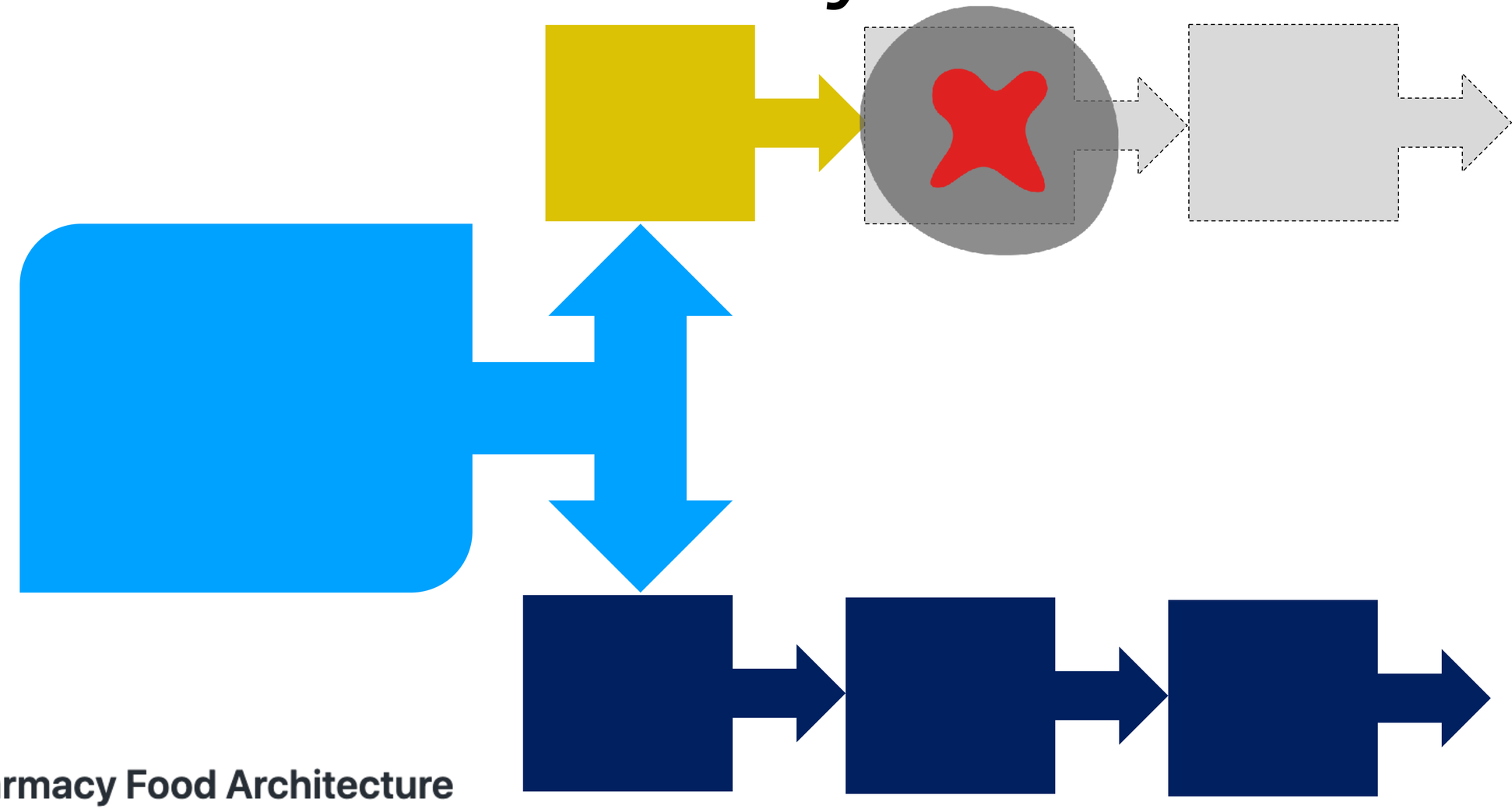


Serverless Integration architecture

Anti-corruption layers



One-way Decisions

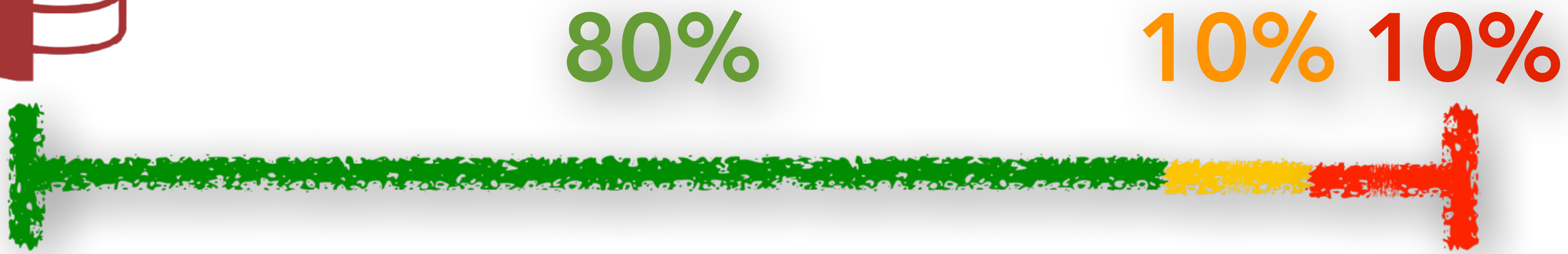


Farmacy Food Architecture

by Jiakaturi

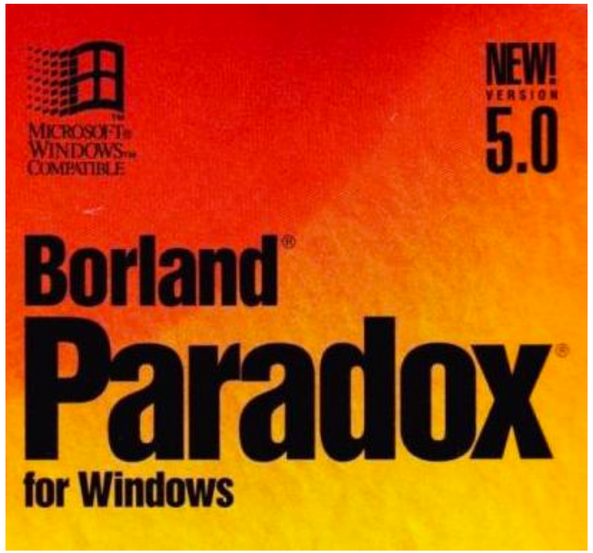
Last 10% Trap

"Users always want 100% of what they want (& are never satisfied with less)."

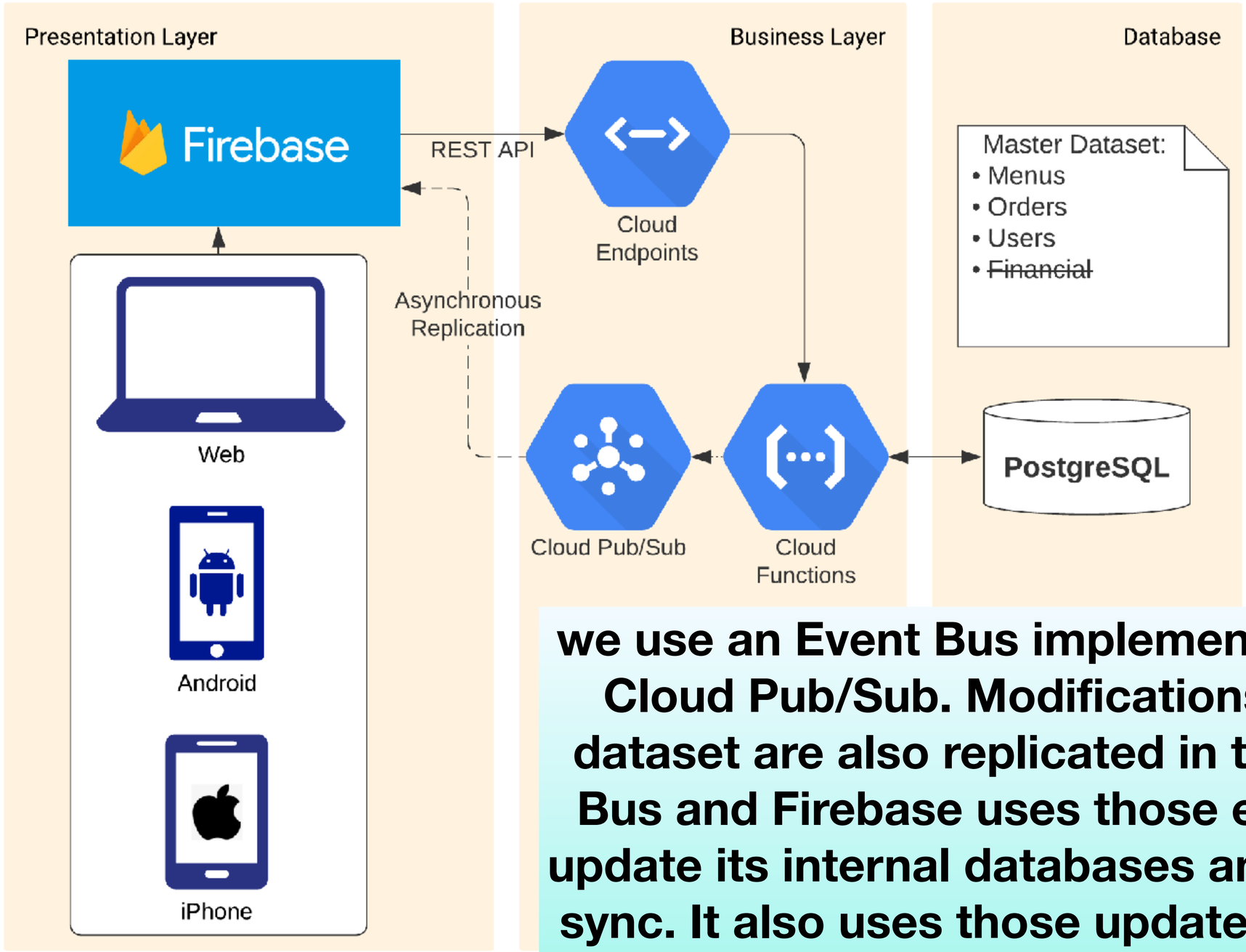


what the user wants

What happened to the 4GLs?



Avoiding Vendor Lockin

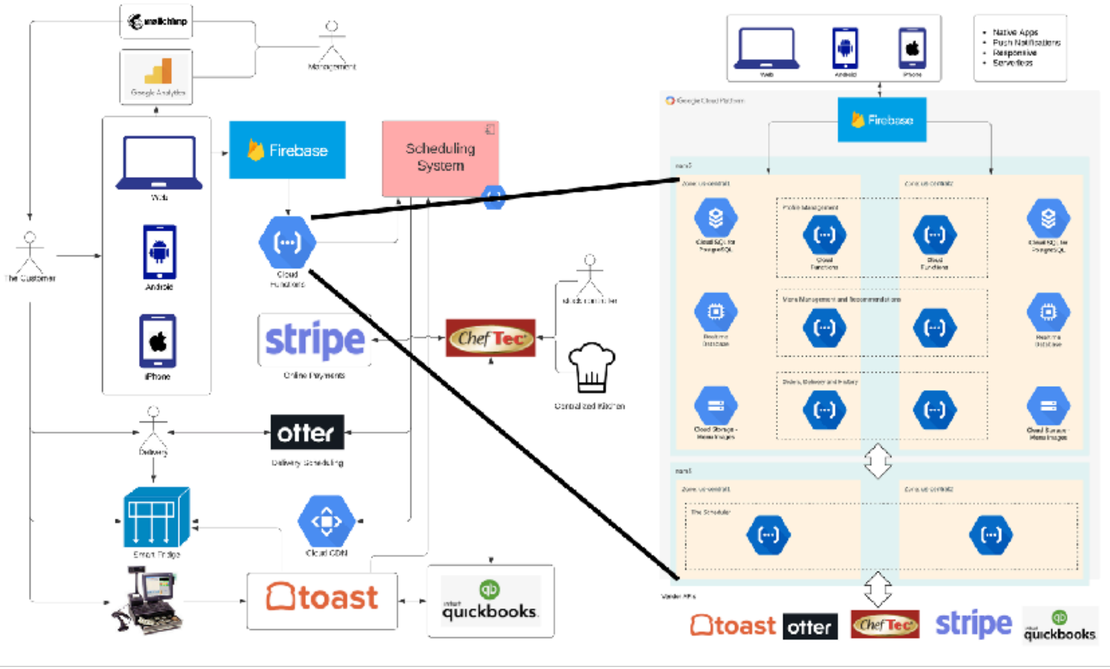


we use an Event Bus implemented using Cloud Pub/Sub. Modifications to the dataset are also replicated in the Event Bus and Firebase uses those events to update its internal databases and keep in sync. It also uses those updates to push notifications on the Web/App.

architecture solution

YOU BE THE ARCHITECT!

As the architect, would you choose this as your solution?!?



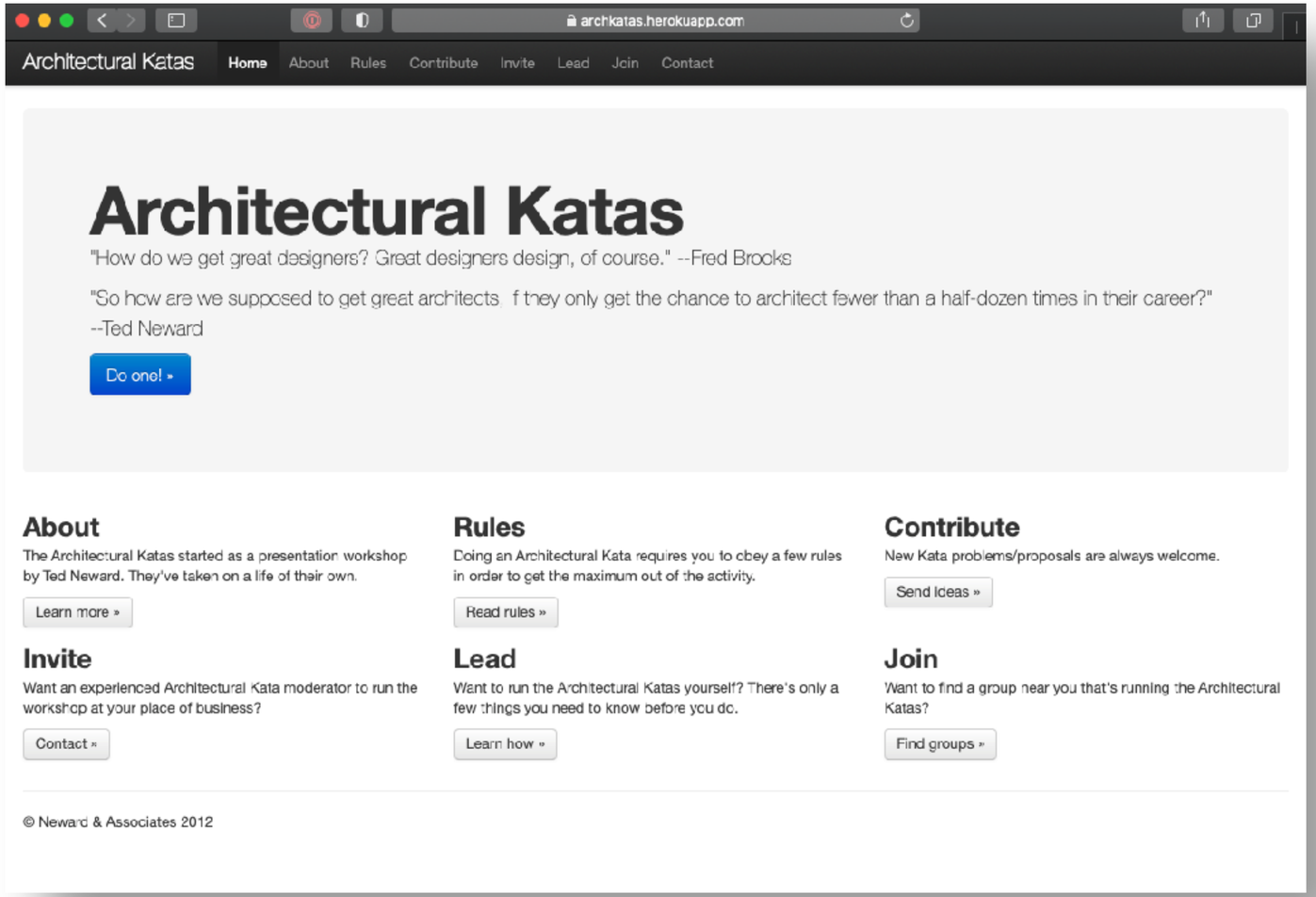
Poll question:

is this serverless integration architecture a viable and justified solution for Farmacy Foods?

YES

NO

Where did this idea come from?



archkatas.herokuapp.com

Architectural Katas | Home | About | Rules | Contribute | Invite | Lead | Join | Contact

Architectural Katas

"How do we get great designers? Great designers design, of course." --Fred Brooks

"So how are we supposed to get great architects, if they only get the chance to architect fewer than a half-dozen times in their career?" --Ted Neward

Do one! »

Where did this idea come from?

About

The Architectural Katas started as a presentation workshop by Ted Neward. They've taken on a life of their own.

Learn more »

Rules

Doing an Architectural Kata requires you to obey a few rules in order to get the maximum out of the activity.

Read rules »

Contribute

New Kata problems/proposals are always welcome.

Send ideas »

Invite

Want an experienced Architectural Kata moderator to run the workshop at your place of business?

Lead

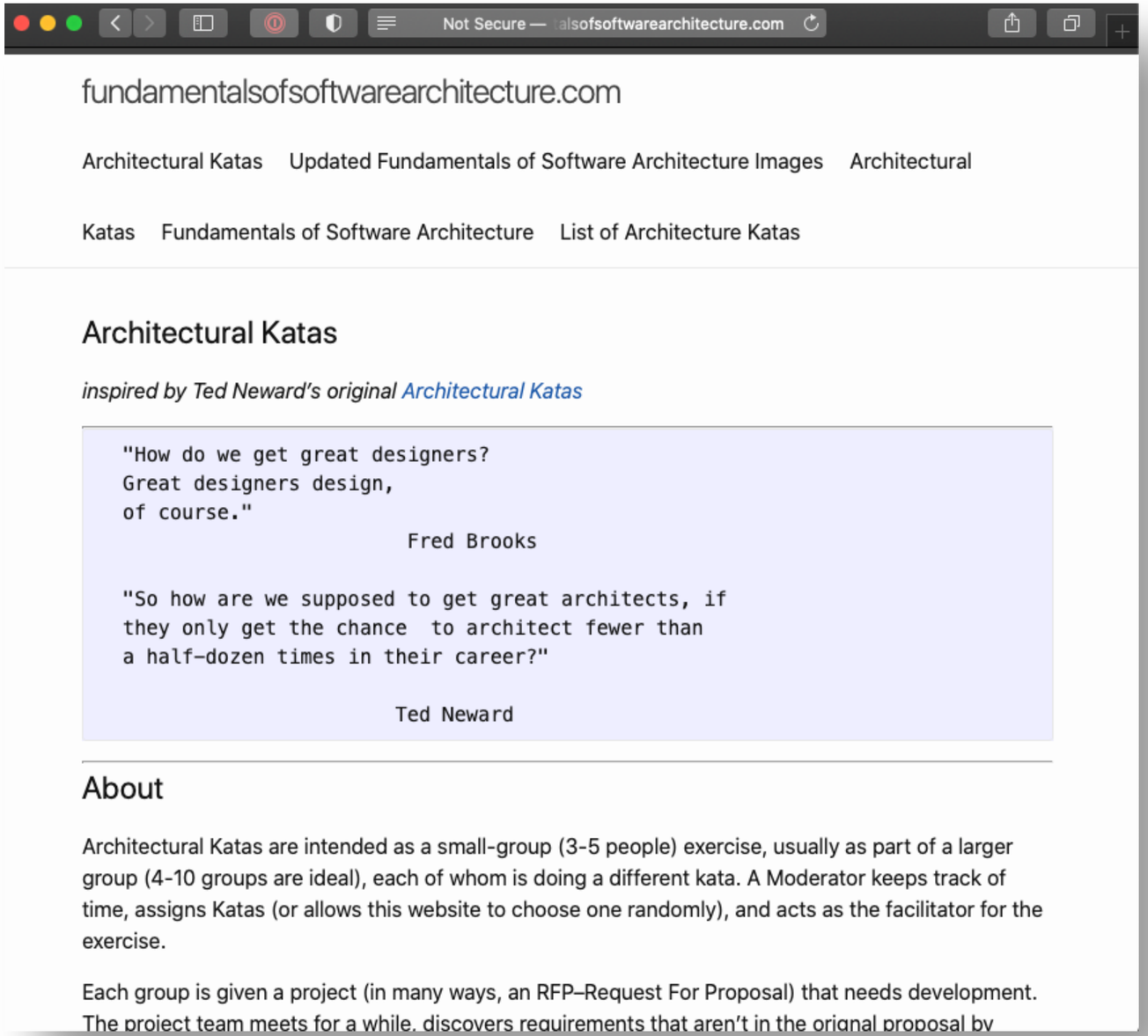
Want to run the Architectural Katas yourself? There's only a few things you need to know before you do.

Join

Want to find a group near you that's running the Architectural Katas?

...and then...

http://fundamentalsofsoftwarearchitecture.com/katas/




...and then...

oreilly.com

O'REILLY®

LIVE ONLINE TRAINING

Architectural Katas

Topic: Software Development

NEAL FORD

Late 2020…

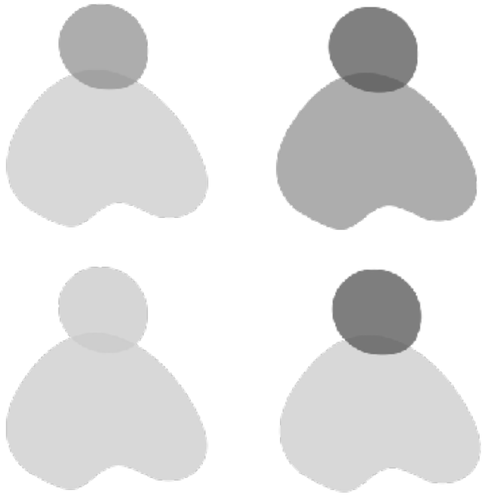
October 20, November 17 & December 3, 2020

10:00am – 12:00pm EST

This course has ended.

What you'll learn Instructor Schedule

New information after 10/20 kickoff:



3–5 members

completely virtual

team constraints

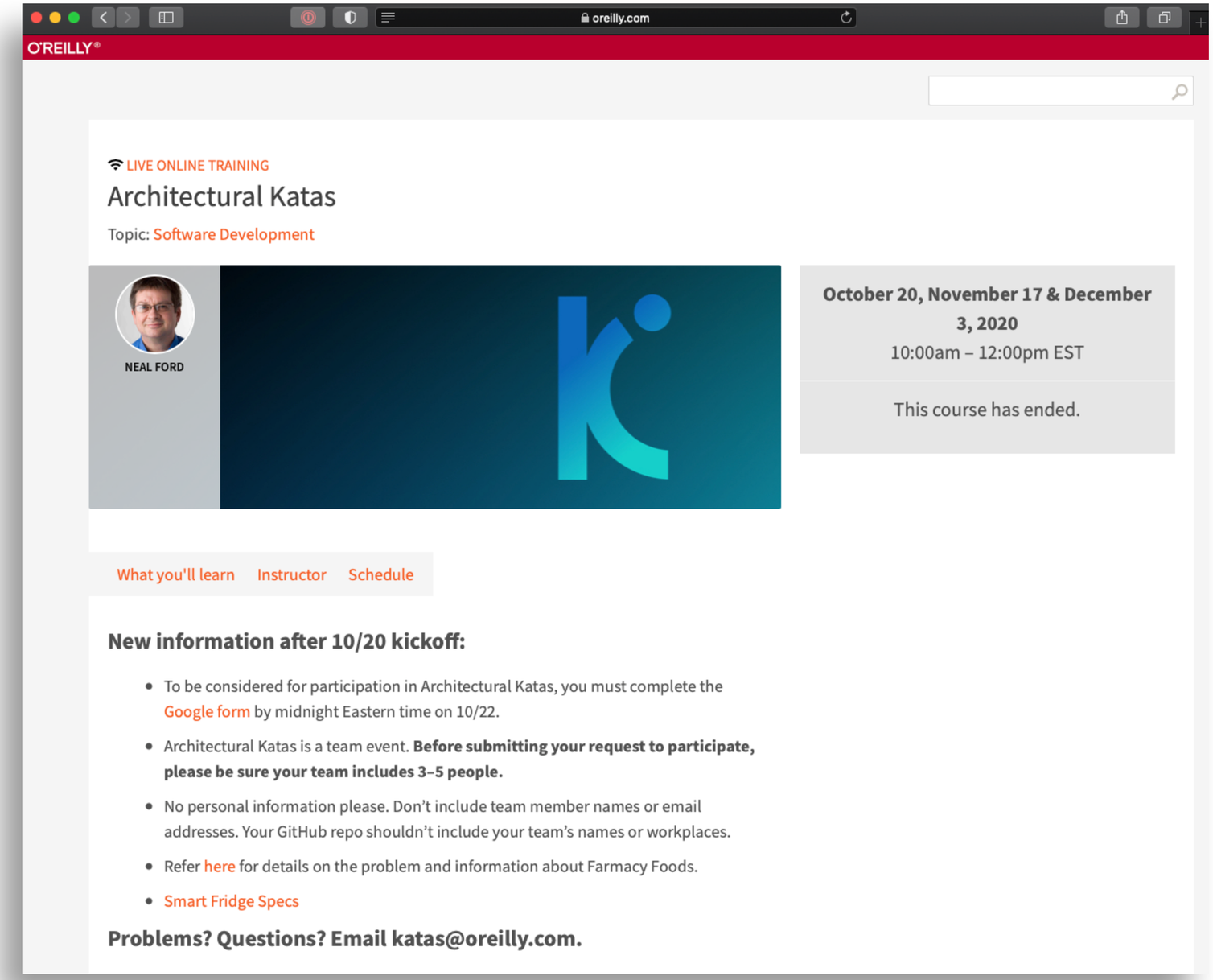
no public judging criteria

sometimes strangers

never been done before!

Epilogue:
THE WINNERS !!

Late 2020...


oreilly.com
O'REILLY
LIVE ONLINE TRAINING
Architectural Katas
Topic: Software Development
NEAL FORD
October 20, November 17 & December 3, 2020
10:00am – 12:00pm EST
This course has ended.
What you'll learn
Instructor
Schedule
New information after 10/20 kickoff:
To be considered for participation in Architectural Katas, you must complete the Google form by midnight Eastern time on 10/22.
Architectural Katas is a team event. Before submitting your request to participate, please be sure your team includes 3–5 people.
No personal information please. Don't include team member names or email addresses. Your GitHub repo shouldn't include your team's names or workplaces.
Refer here for details on the problem and information about Farmacy Foods.
Smart Fridge Specs
Problems? Questions? Email katas@oreilly.com.

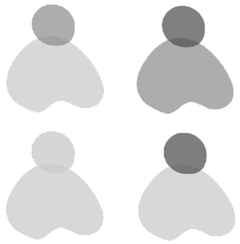
“Highly Commended”

Farmacy Food Architecture

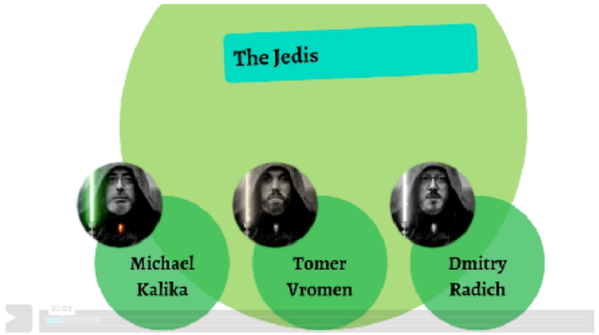
by

Jiakaturi

“**SELLING** an architecture and getting buy-in from stakeholders is a critical part of architecture - good job! Such an amazing video that truly demonstrated an understanding of the problem”



Third Place



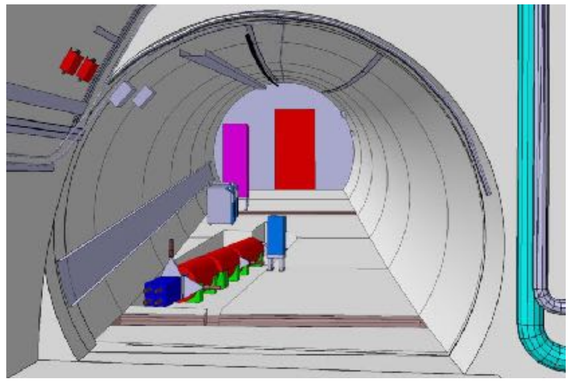
“LOVED the short/mid/long term view as well as a detailed plan on how the architecture would grow with the business. Excellent work tying business drivers to technical needs. No one size fits all architecture. Driven from the customer perspective.”

Second Place

Miyagi's Little Forests

“Good characteristics analysis, really good narrative. Diagrams very complete with lots of views. Good level of detail along with component descriptions, great actor / requirement analysis. Not fully convinced by the Microservices ADR.”

First Place



ArchCollider

“Implementing solution for a startup. Cost and Risk analysis (only ones to catch this).”

Resources

architecture narrative

Architecture Narrative Checklist
☐ What is the business problem?
☐ Why is the business doing this?
☐ Who are the major players (stakeholders)?
☐ What are the business and technical constraints?
☐ What architecture characteristics must be supported?
☐ What are the unique challenges of this business problem?
☐ What are the alternatives and corresponding tradeoffs?
☐ What is the proposed architectural solution?
☐ What artifacts are used to describe the architecture?
☐ What are the major architectural decisions?

architecture characteristics

Architecture Characteristics Checklist	
☐	What is the scope of the identified characteristics?
☐	Can you justify the identified characteristics?
☐	Can you tie them back to business needs or requirements?
☐	Are all the characteristics critical or important to success?

architecture decisions

Architecture Decisions Checklist	
☐	Is this an architecture or design decision?
☐	Is the decision properly justified?
☐	What is the scope of the decision?

effective diagrams

- [ ] Short, meaningful titles
- [ ] Lines: descriptions
- [ ] Lines: unidirectional
- [ ] Lines: synchronous or asynchronous
- [ ] Lines: blocking or non-blocking
- [ ] Shapes: consistency
- [ ] Avoid acronyms
- [ ] Color check for consistency, contrast, clarity
- [ ] Most important thing(s) centered
- [ ] Includes key

architecture solution

☐	What is the scope of the decision?
☐	Is your decision architecture or design related?
☐	Is your decision justified properly?

Kata Team Repositories

Kata Team	Github Repository
Super Kings	https://github.com/lastlegion/arch-katas
Pacman	https://github.com/icedhacker/architecture-katas
Hananoyama	https://github.com/hananoyama/architectural_kata
Jiakaturi (highly commended)	https://github.com/lookfwd/archkata
SelfDrivenTeam	https://github.com/selfdriventeam/kata
Hey Dragon	https://github.com/heydragon2020
daVinci	https://github.com/mtykhenko/davinci-kata
The Jedis (third place)	https://github.com/TheJedis2020/arch_katas_2020
ArchColider (first place)	https://github.com/ldynia/archcolider
Miyagi's Little Forests (second place)	https://github.com/miyagis-forests

O'REILLY® LIVE ONLINE TRAINING

Describing Software Architecture

Neal Ford
Thoughtworks
Director / Software Architect / Meme Wrangler
http://www.nealford.com
@neal4d

Mark Richards
Independent Consultant
Hands-on Software Architect, Published Author
Founder, DeveloperToArchitect.com
@markrichardssa