



LIVE ONLINE TRAINING

Describing Software Architecture Workbook



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

The image shows a screenshot of the FarmacyFood website. At the top, there's a navigation bar with icons for search, refresh, and user account, followed by the URL "farmacyfood.com". On the right side of the bar are links for "ABOUT US", "INTERESTED?", "Sign Up", and a shopping cart icon with the number "0".

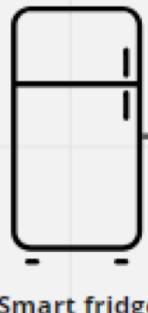
The main visual is a large photograph of a bowl of shrimp soup garnished with cilantro and green onions. Overlaid on this image is the text "Let Food be Thy Medicine" in a large, white, sans-serif font. Below this, a green button contains the text "LEARN MORE >".

Below the main image is a green banner with a repeating leaf pattern. It features two smaller images: on the left, two black meal prep containers filled with pasta and vegetables; on the right, a smartphone displaying the FarmacyFood app interface, which includes sections for "OUR FOOD", "OUR MISSION", and "YOUR IMPACT".

The FarmacyFood logo, consisting of the word "Farmacy" above "Food" with a stylized leaf icon, is positioned in the center of the green banner. Below the logo, the text "Healthy, locally sourced meals for delivery or pick-up." is displayed in a white, sans-serif font. At the bottom of the banner, a white button contains the text "GET STARTED".



Customer
in person)



Smart fridge



Customer
in person)



Cashier POS



Customer
using the app

Farmacy Food system



Smart fridge mgmt
system (in the cloud)



Vendor mgmt system
(in the cloud)



PayPal (or other
payment gateway
provider)



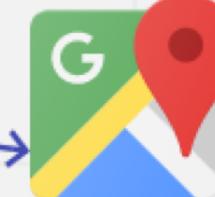
Auth0 (or other
authentication
provider)



Farmacy Food
central kitchen



eDietitian (expert
system for diet
recommendation)



Google maps
(or other map
provider)

architecture narrative

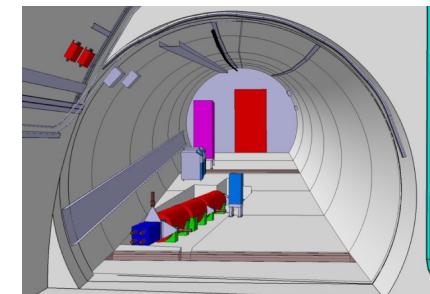
YOU BE THE JUDGE!



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



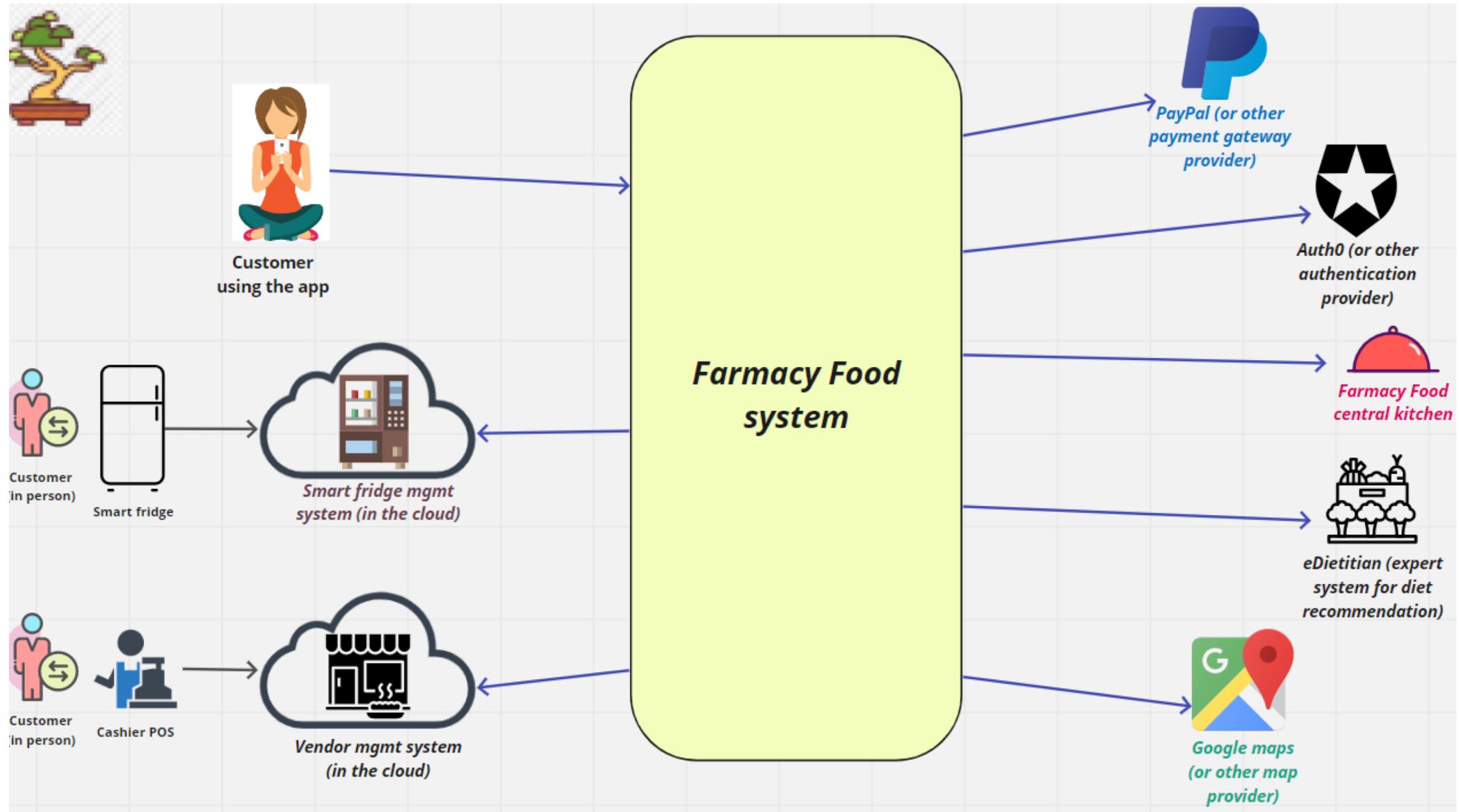
Myagi's Little Forests
<https://github.com/miyagis-forests/farmacy-food-kata>



Arcolider
<https://github.com/ldynia/archcolider>

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

Viability

Flexibility

Availability

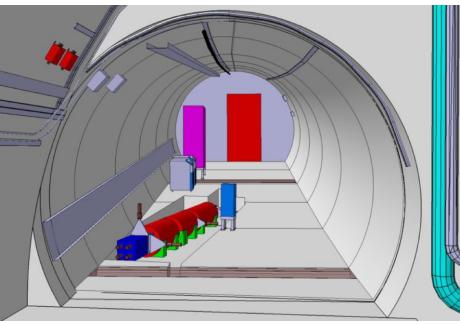
Security

Scalability

Performance

 **JiaKaturi**

<https://github.com/lookfwd/archkata>



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

Viability

Availability

Security

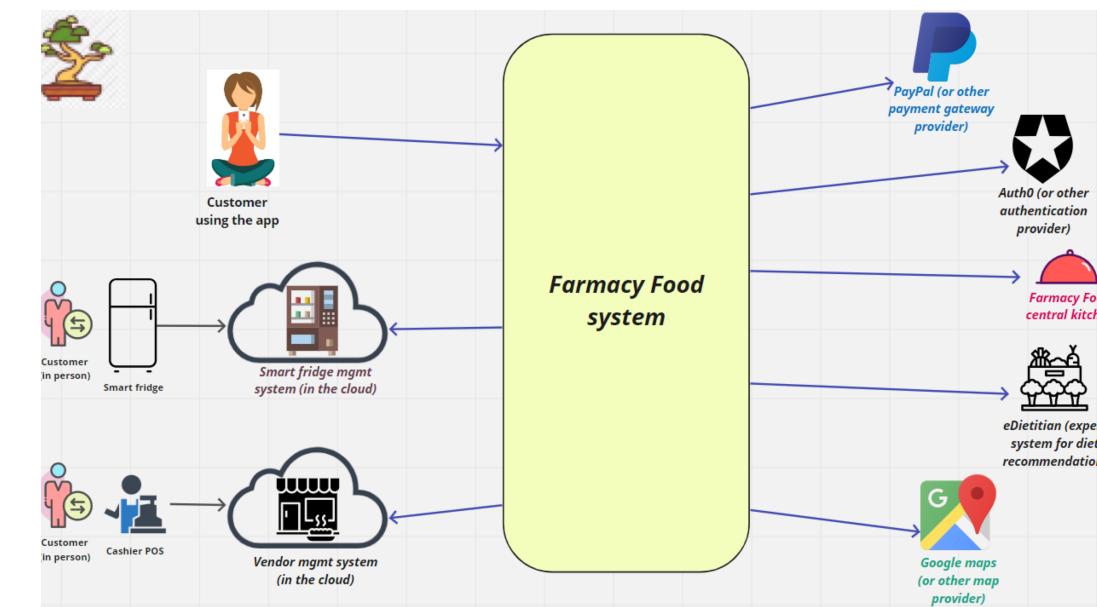
Scalability

Performance

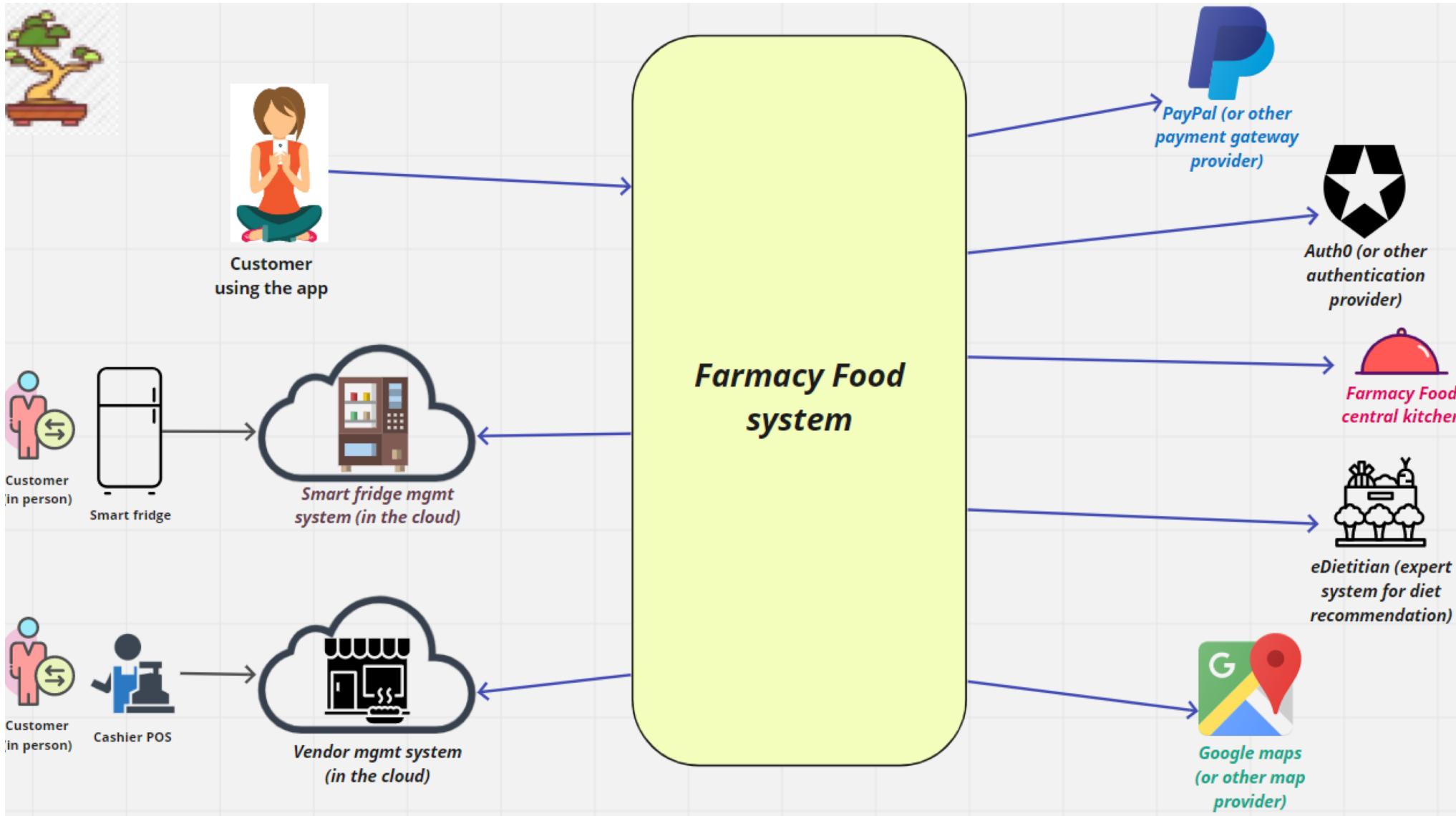
Interoperability

Data Integrity

Simplicity



architecture characteristics



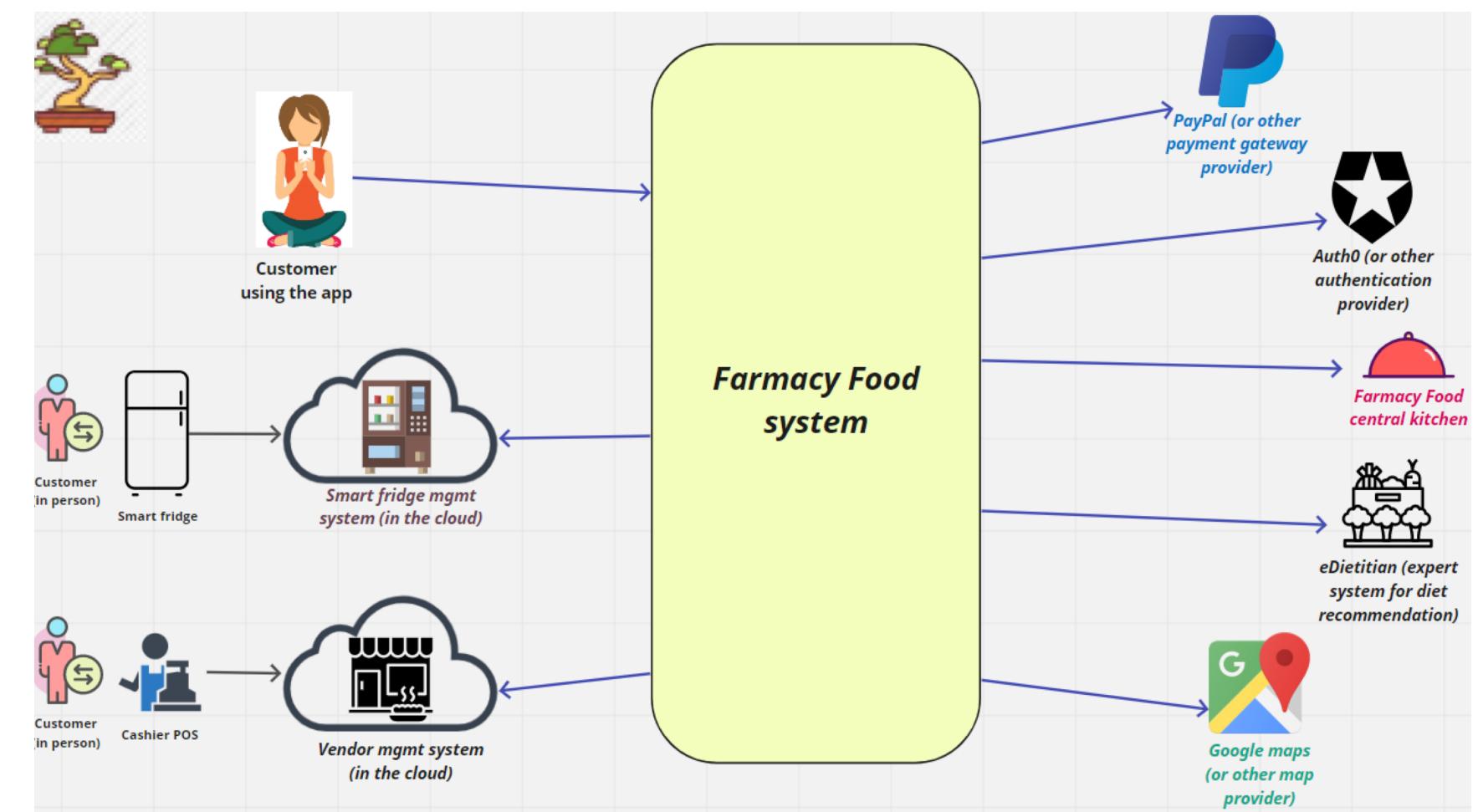
Agility
Viability
Availability
Security
Scalability
Performance
Interoperability
Data Integrity
Simplicity

architecture characteristics

YOU BE THE JUDGE!



Should Farmacy Foods be a monolithic or distributed architecture?



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 implementation.



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

architecture decisions

YOU BE THE JUDGE!



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

(who has the responsibility to make the decision?)

architecture

design

A

B

C

D

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.

architecture

design

A

B

C

D



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.

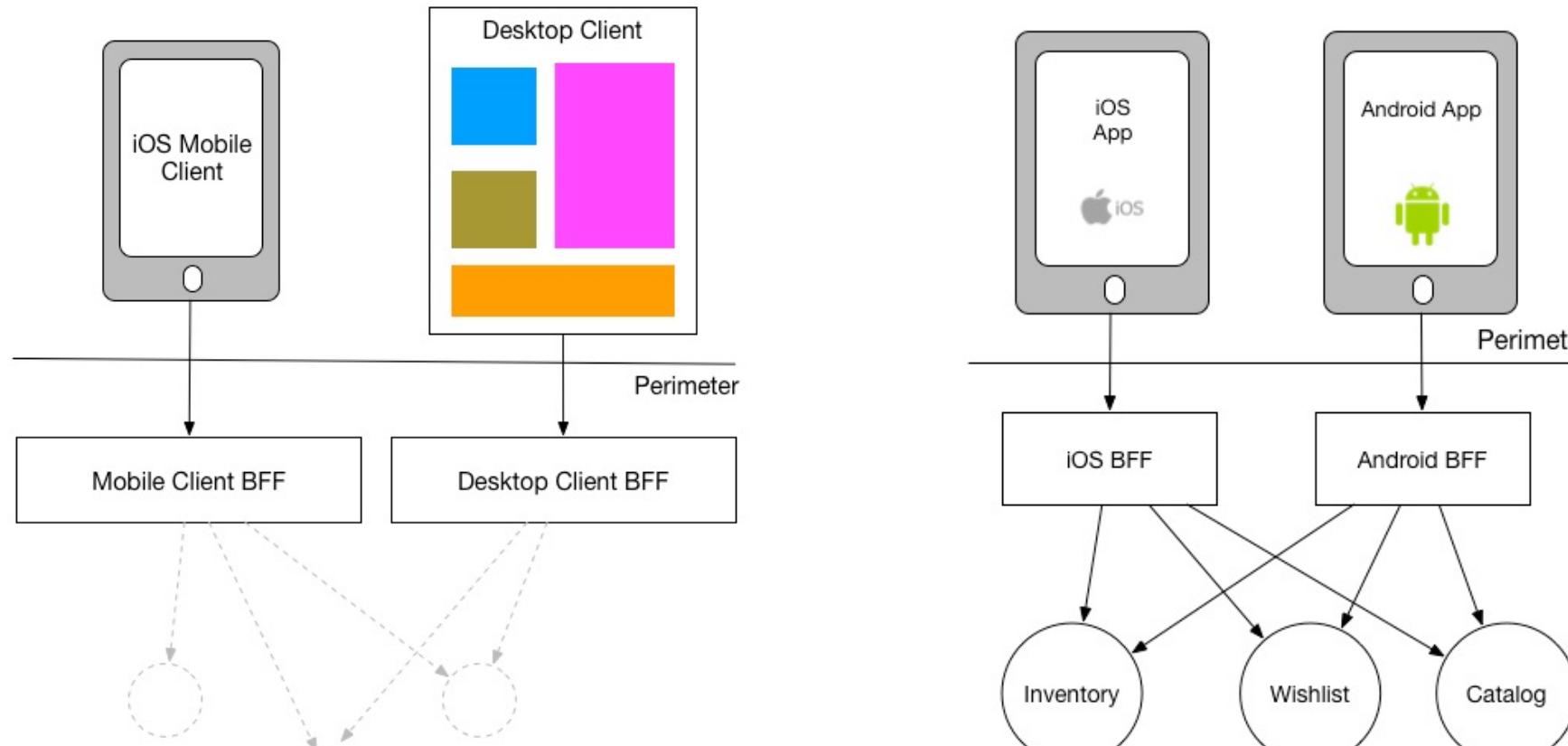
Decision

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

A

B

C

design

D



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?

architecture

A

B

C

design

D



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.

architecture

A

B

C

design

D

Are the architecture characteristics demonstrated in the solution?

Agility

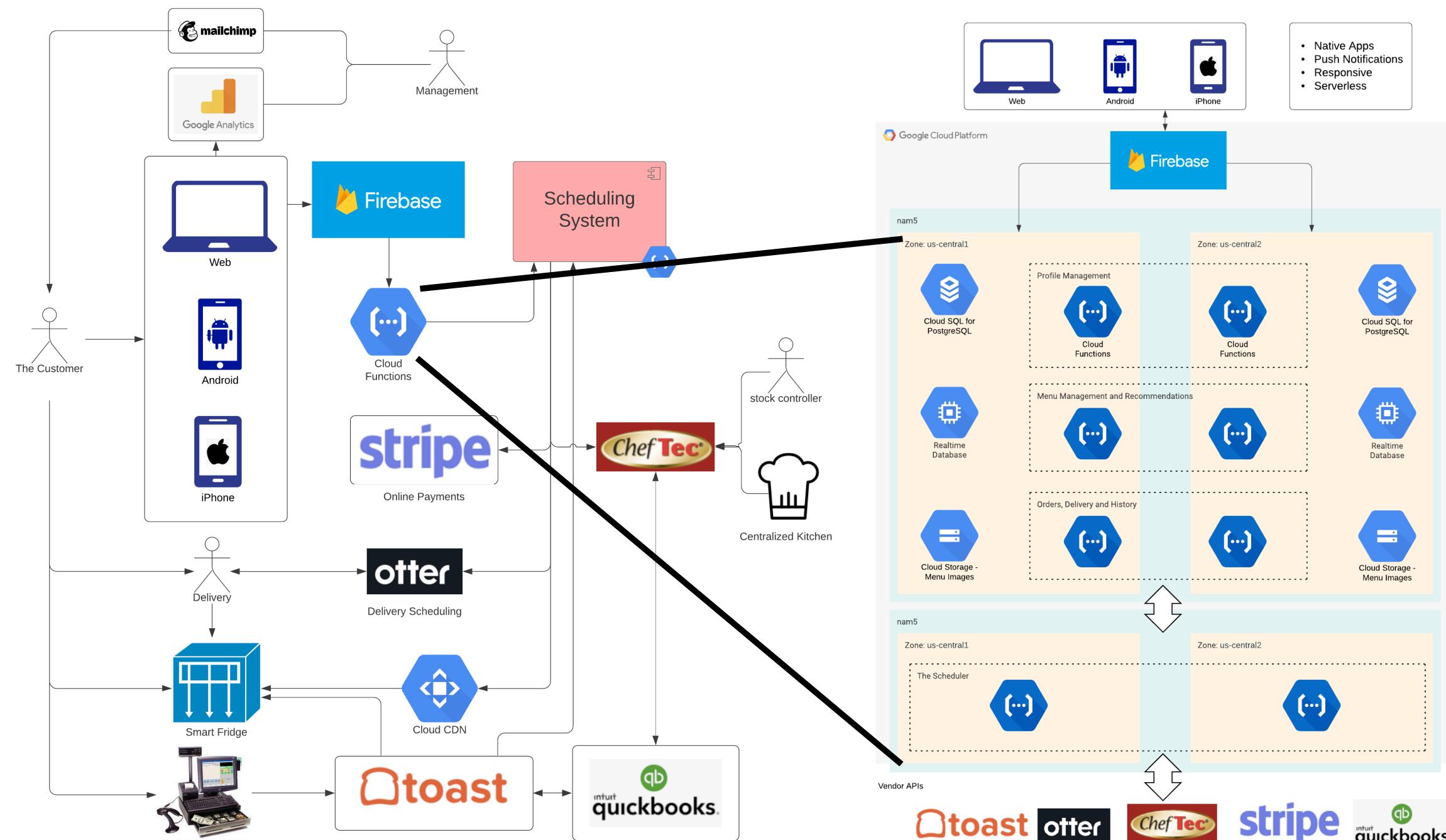
Viability

Flexibility

Availability

Security

Scalability



Are the architecture characteristics demonstrated in the solution?

✓ Agility

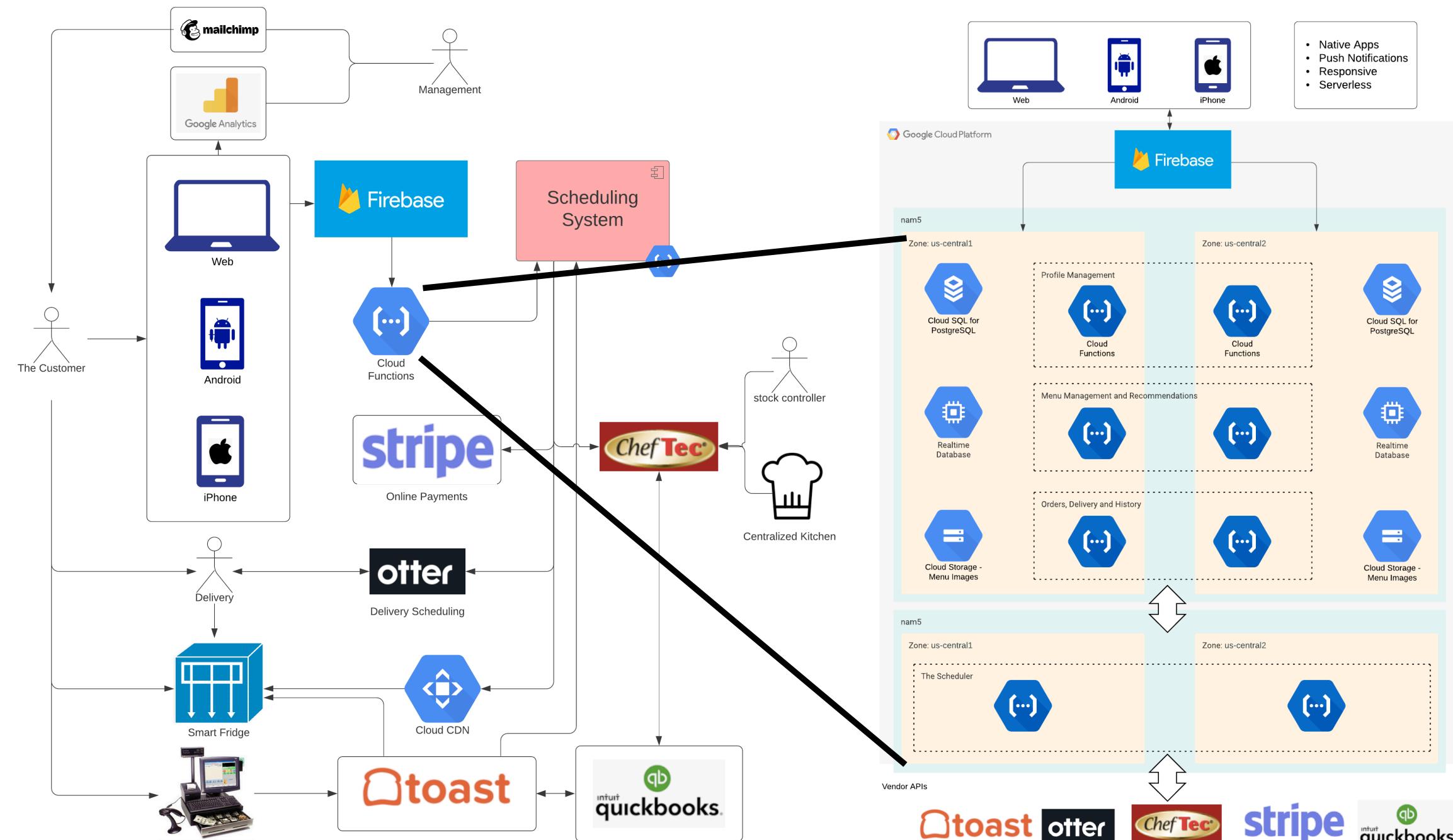
Viability

Flexibility

✗ Availability

Security

Scalability

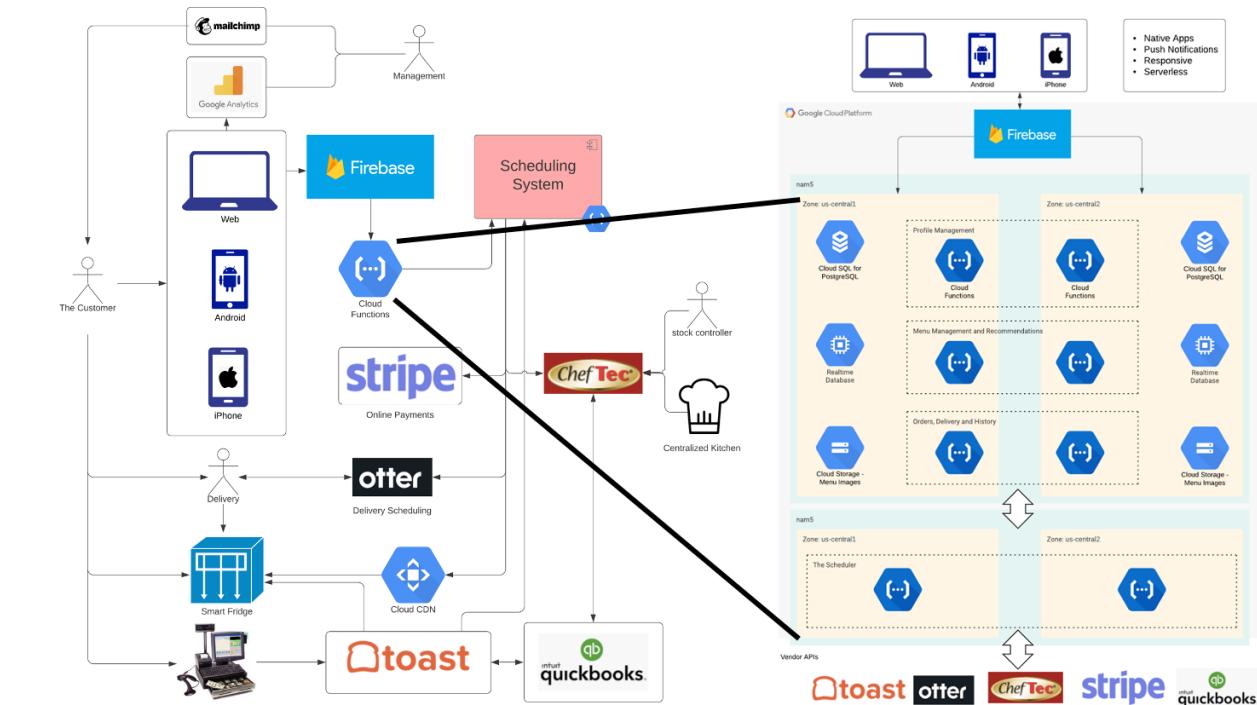


architecture solution

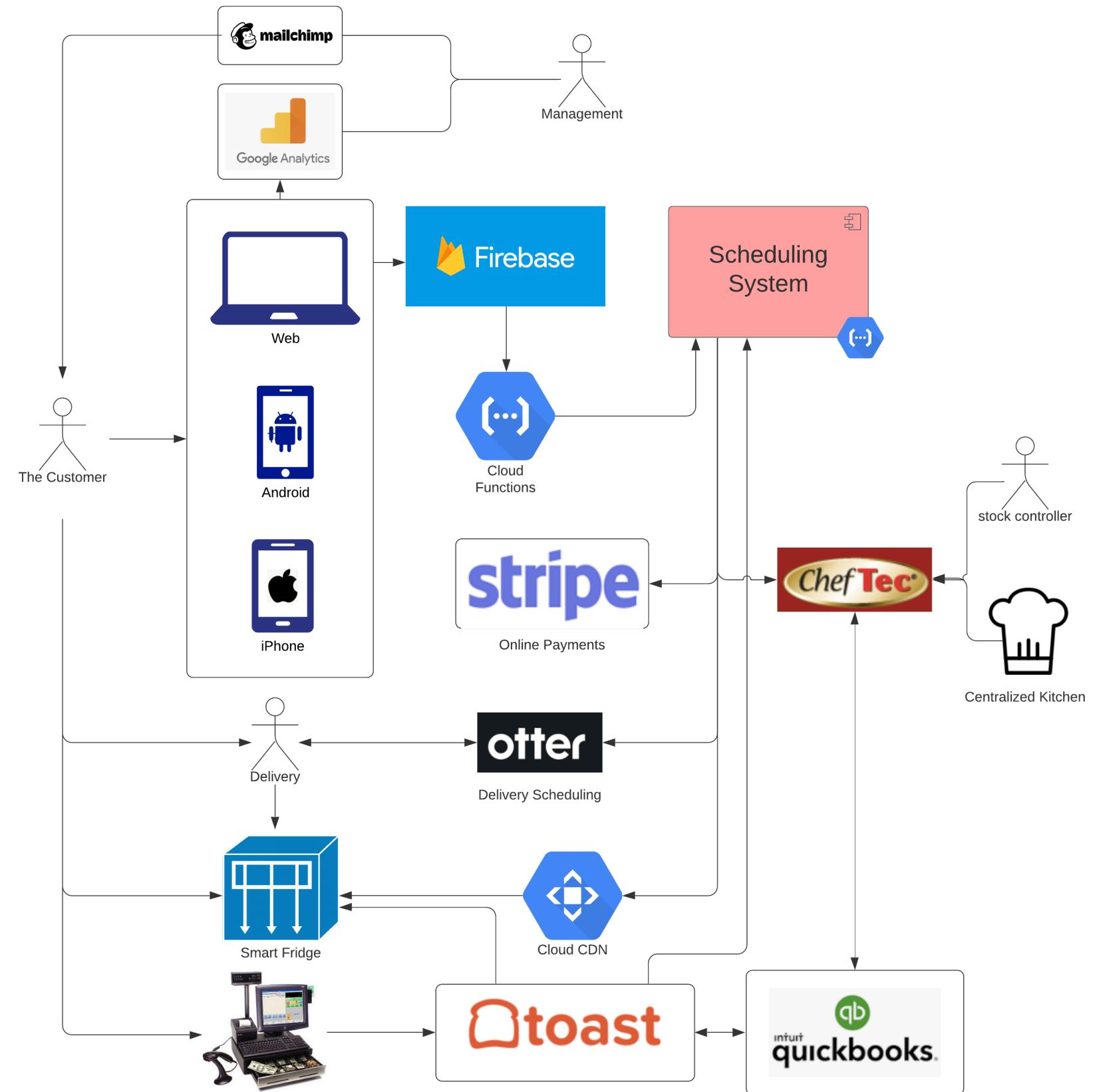
YOU BE THE JUDGE!



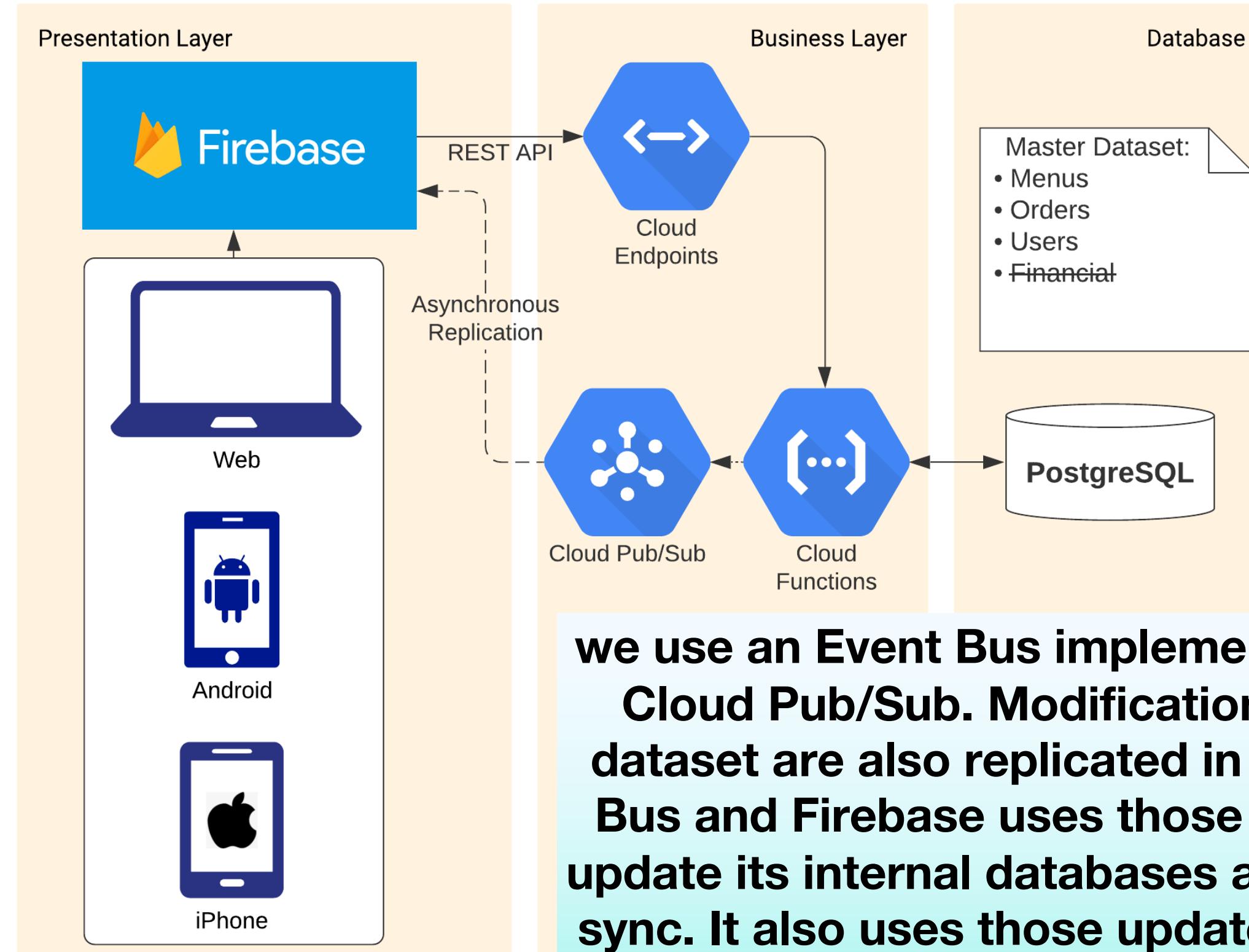
Do you think availability is demonstrated by the proposed solution by Jaikaturi?



Serverless



Avoiding Vendor Lockin



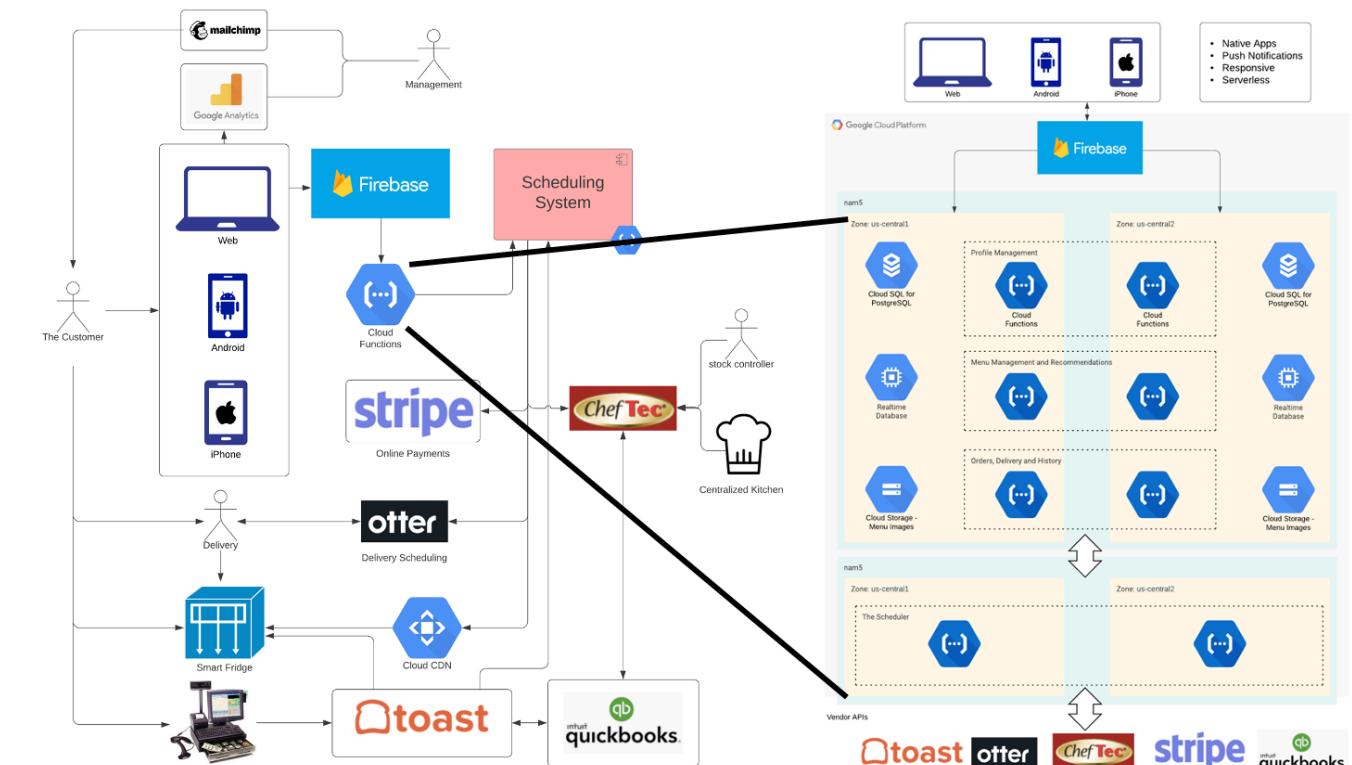
we use an Event Bus implemented using Cloud Pub/Sub. Modifications to the dataset are also replicated in the Event Bus and Firebase uses those events to update its internal databases and keep in sync. It also uses those updates to push notifications on the Web/App.

architecture solution

YOU BE THE ARCHITECT!



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



Resources

architecture narrative

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

architecture characteristics

Architecture Characteristics Checklist	
<input type="checkbox"/>	What is the scope of the identified characteristics?
<input type="checkbox"/>	Can you justify the identified characteristics?
<input type="checkbox"/>	Can you tie them back to business needs or requirements?
<input type="checkbox"/>	Are all the characteristics critical or important to success?

architecture decisions

Architecture Decisions Checklist	
<input type="checkbox"/>	Is this an architecture or design decision?
<input type="checkbox"/>	Is the decision properly justified?
<input type="checkbox"/>	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/selfdriveteam/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