

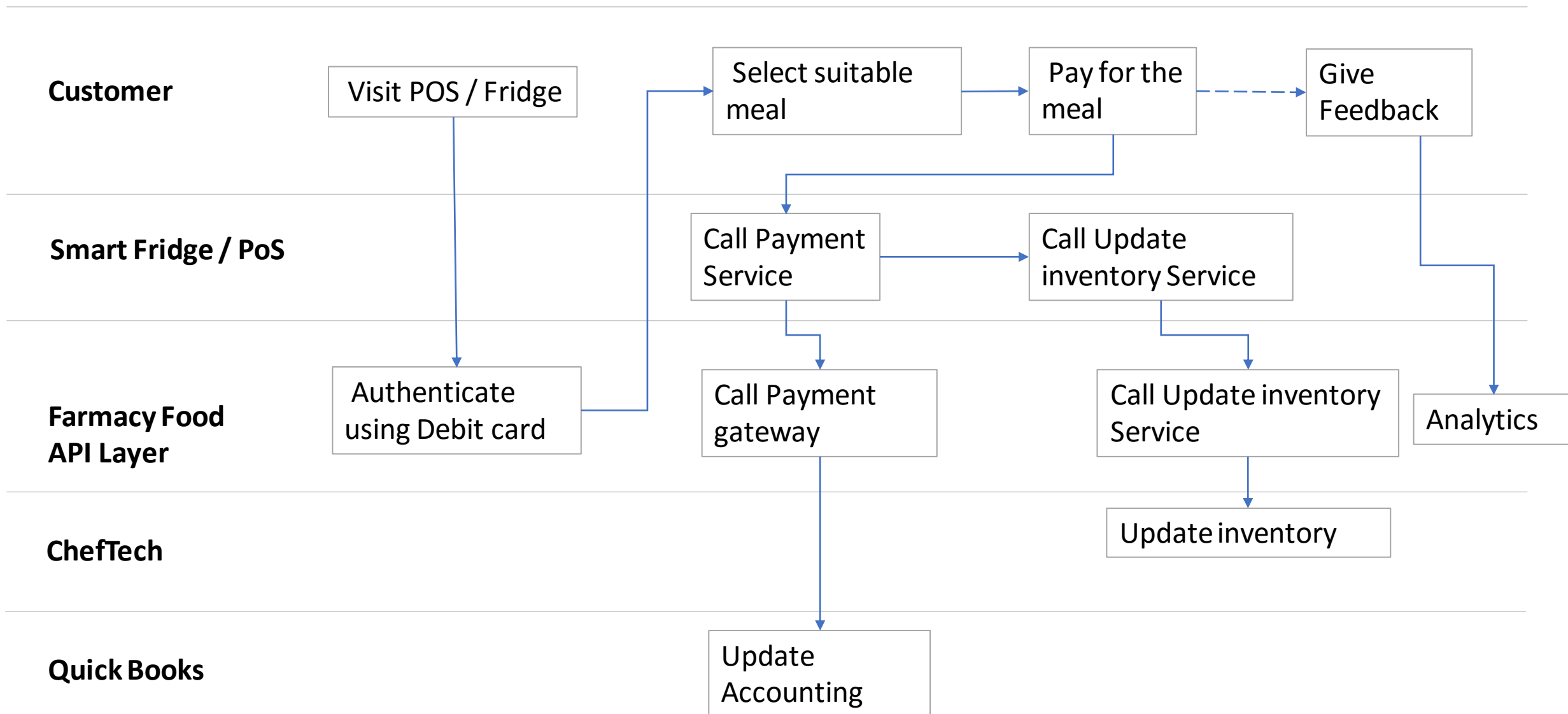
Farmacy Food Kata - Team MealAMan

# Assumptions

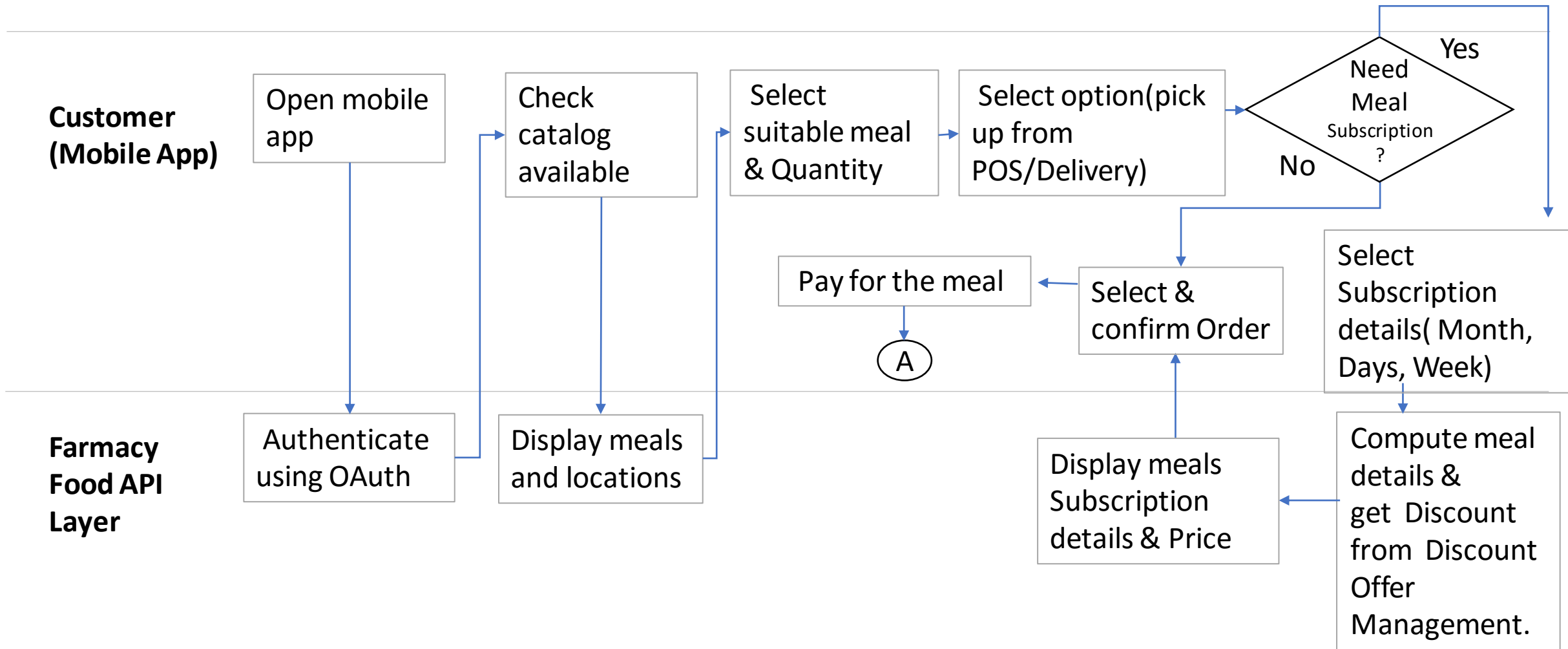
- SmartFridge has APIs available for fetching the purchase done including - item names, quantity, amount and location of the fridge
- SmartFridge has APIs available for fetching the details of inventory available - item names, quantity and location of the fridge, Item\_loadedDate
- PoS system will have web interface for the employee to enter item and quantity available
- PoS APIs are available for fetching the purchase done including - item names, quantity, amount and location of the PoS
- Location services will be used and integrated in the mobile app for showing availability option to user
- Existing delivery management system will be used.
- Payment collection and processing happens at FarmacyFood end. Commission and rent processing for SmartFridge and Kiosk Outlet space is out of the scope.
- Subscription model will not be available for Smart Fridge
- Supplier management will be taken care via Chef Tech

# Workflows

# Meal Purchase from PoS / Smart Fridges



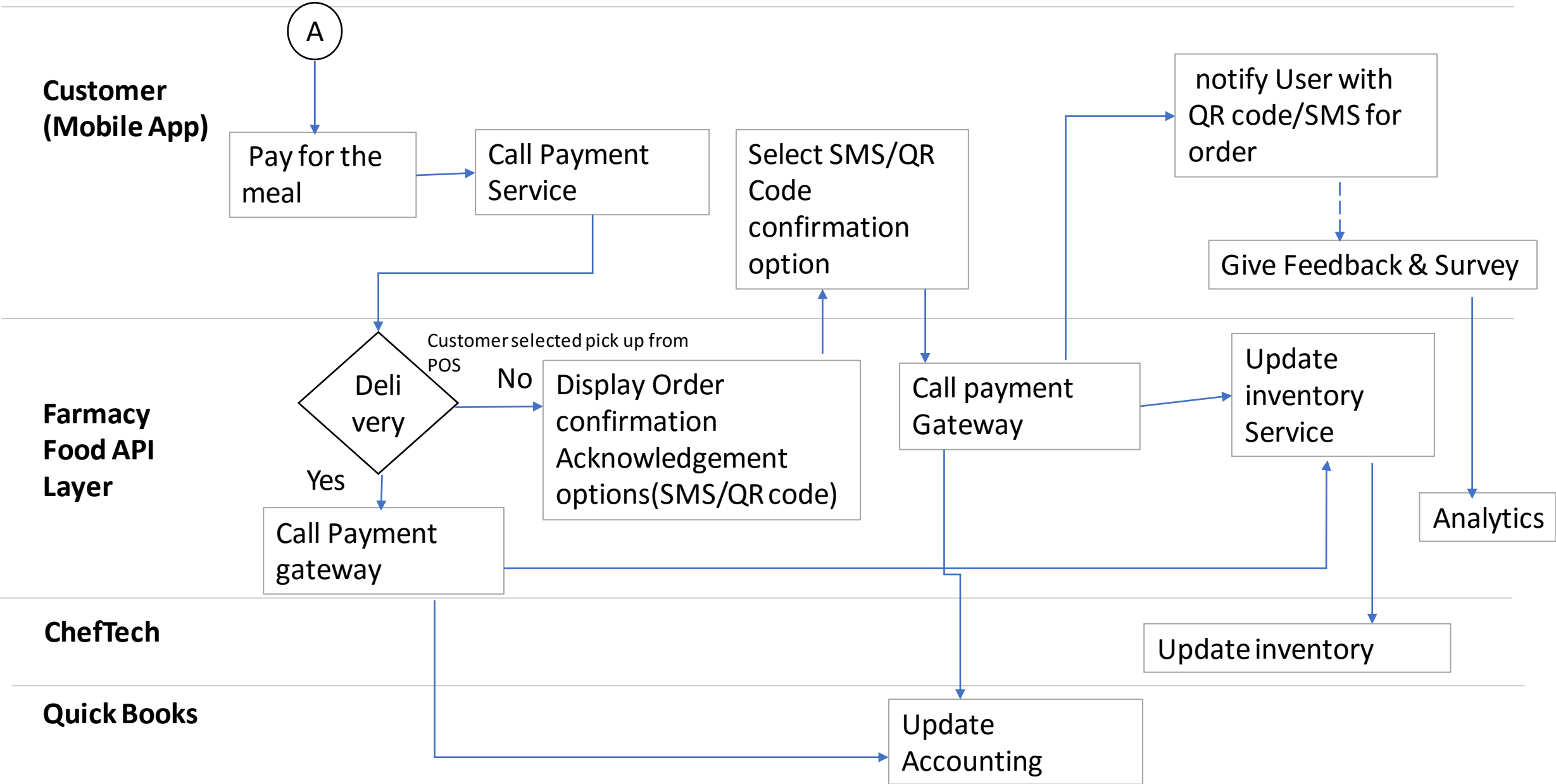
# Meal Selection Order from Mobile / Web



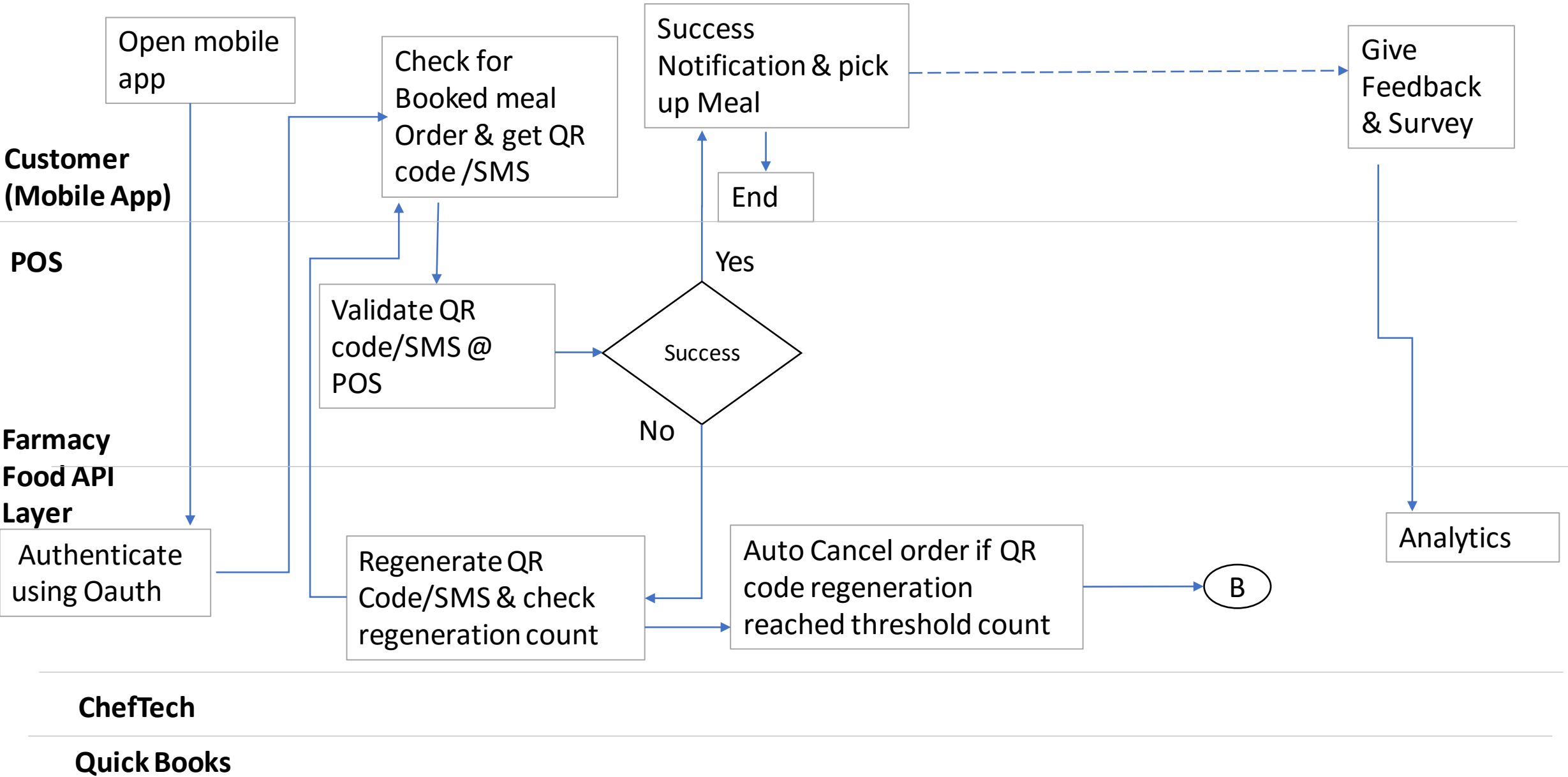
ChefTech

Quick Books

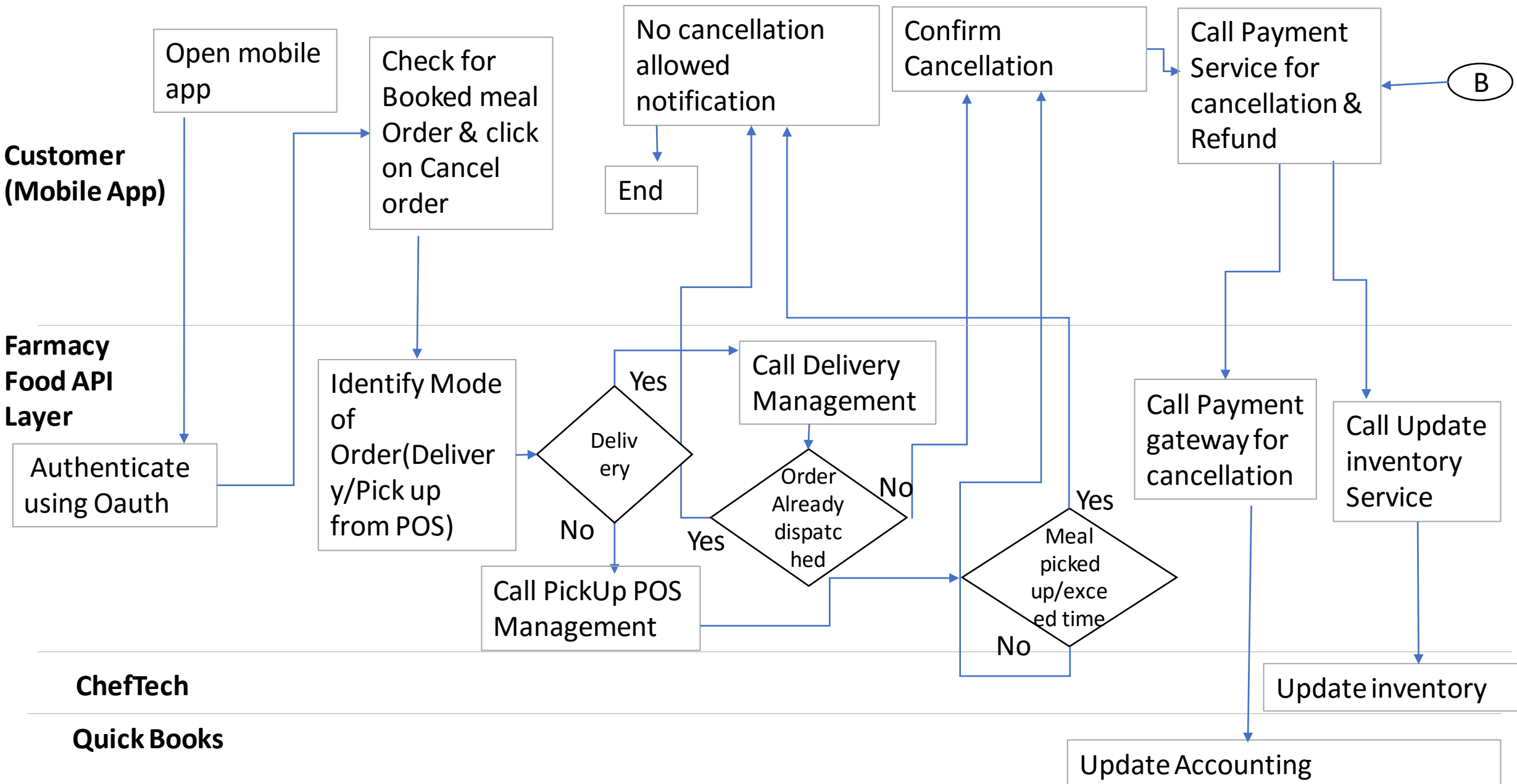
# Selected meal purchase from Mobile / Web



# Pickup from POS

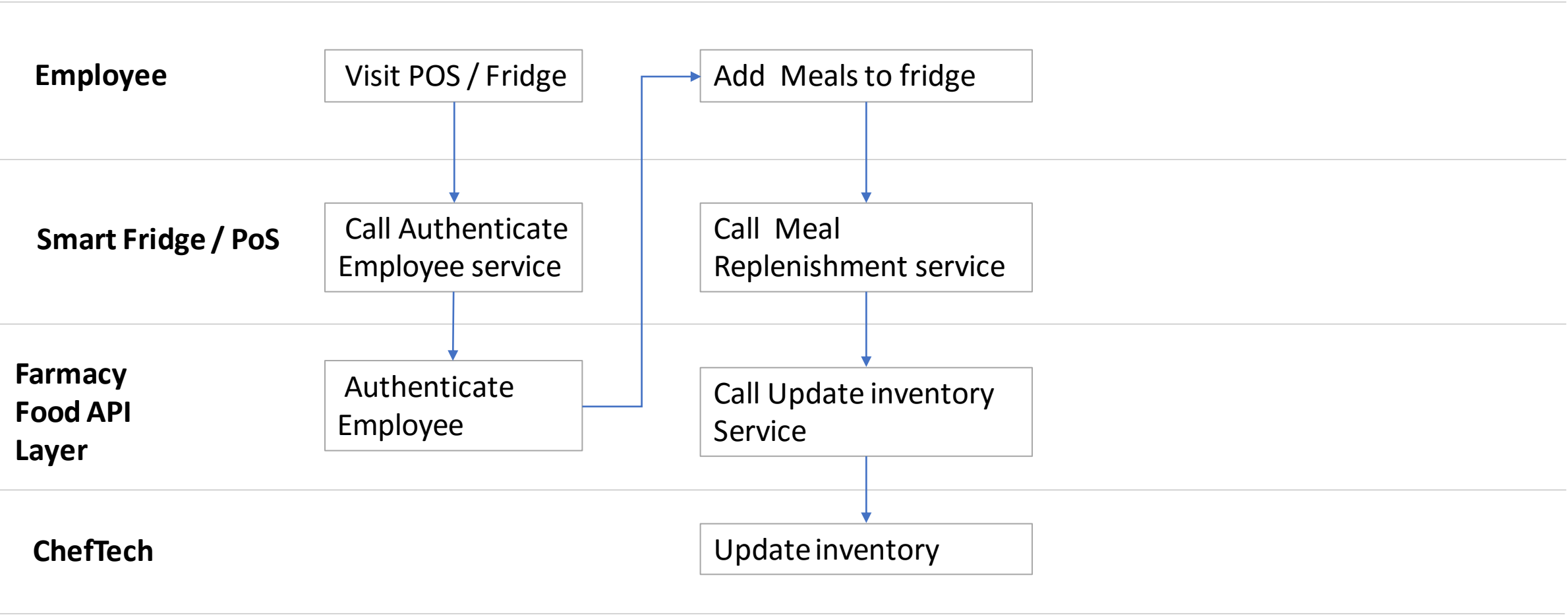


# Cancel order from Mobile / Web

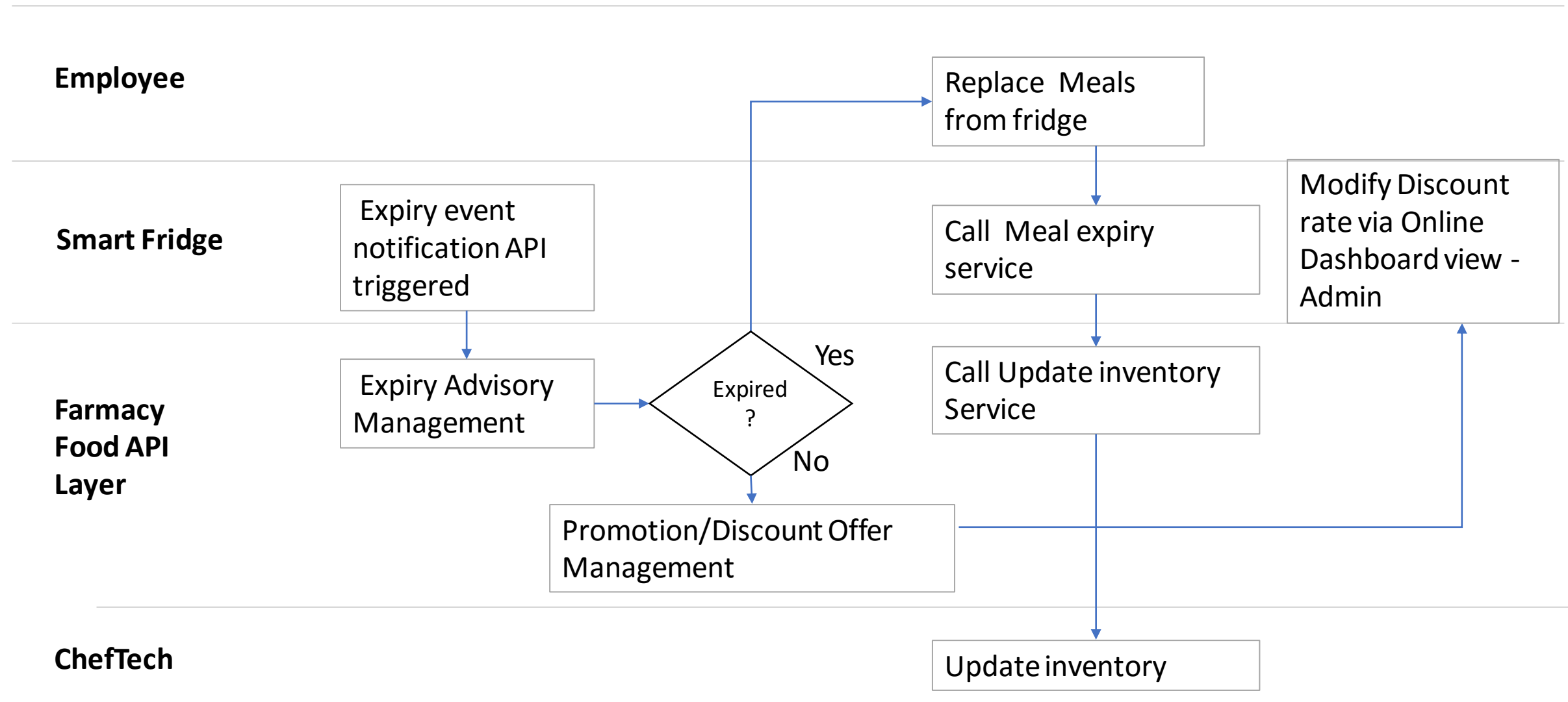




# Meal Replenishment

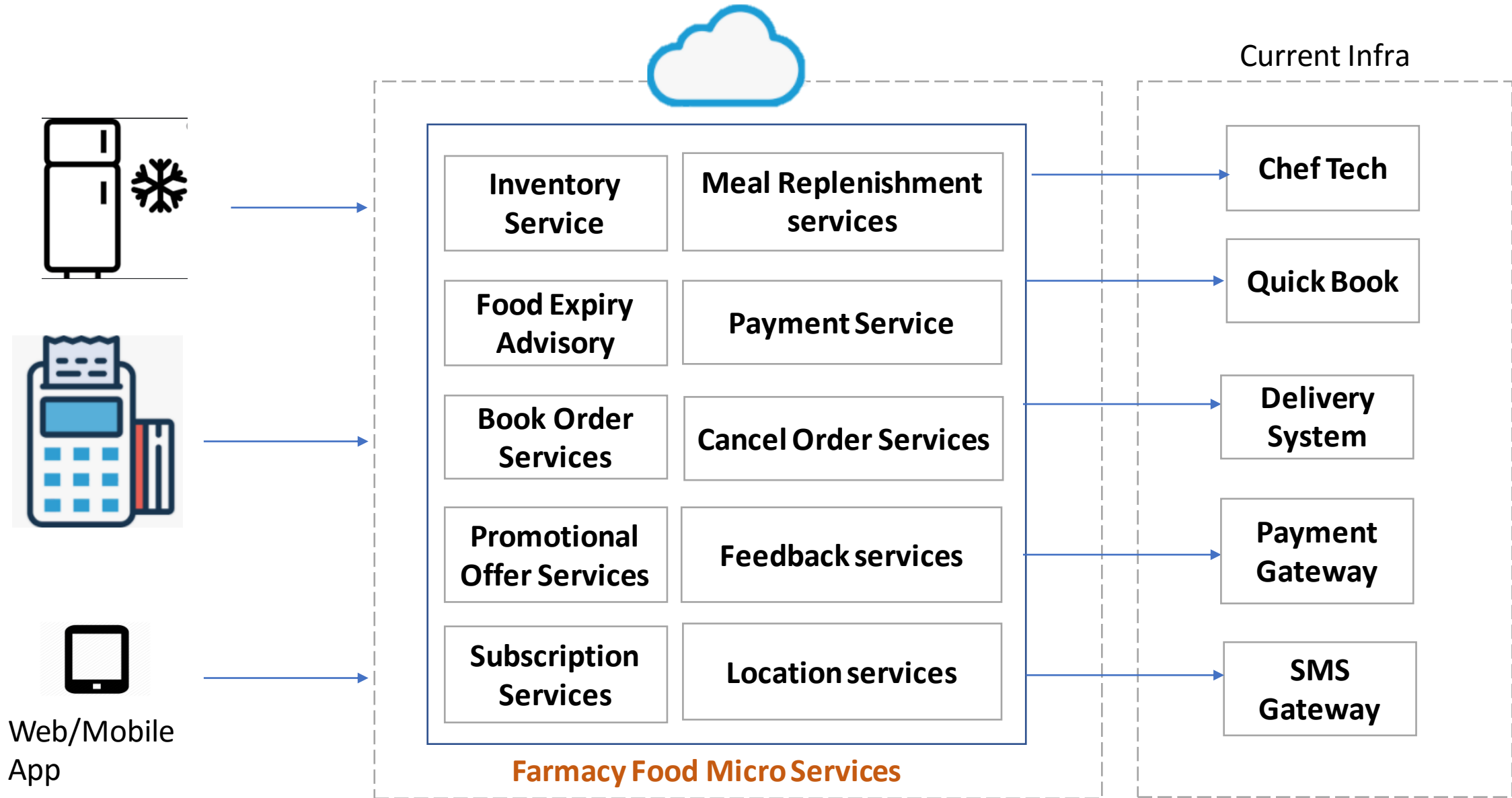


# Meal Expiry Advisory

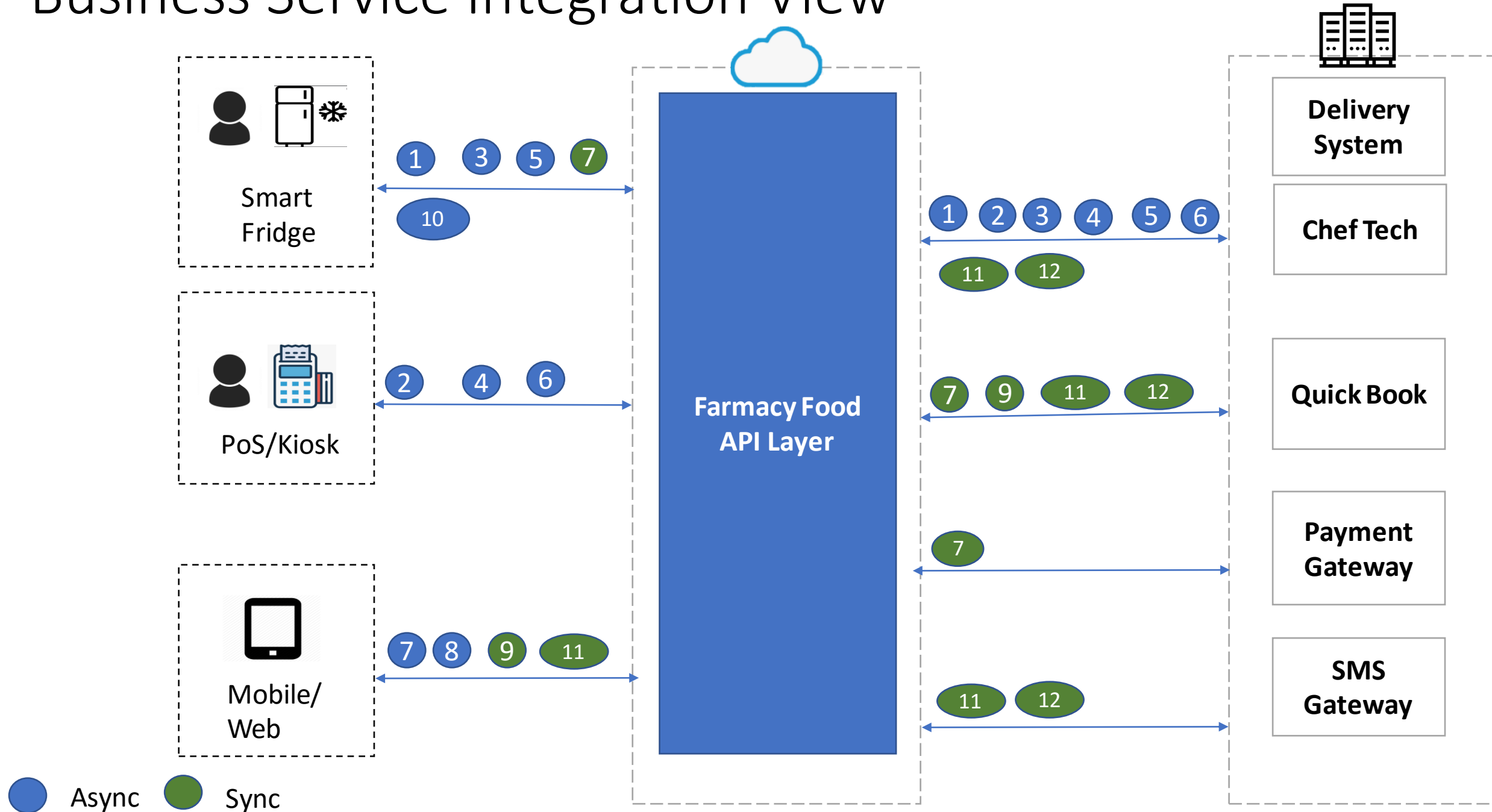


# Topology Diagram

# Farmacy Food Business Service View



# Business Service Integration View



# List of Business Services

- 1 **PushSmartFridgePurchase** - Push information on any purchase from smart fridge.
- 2 **PushPoSPurchase** - Push information on any purchase from PoS/Kiosk.
- 3 **PushSFItemAddedDetails** - Push meal replenishment information from smart fridge.
- 4 **PushPoSItemAddedDetails** - Push meal replenishment information from PoS/Kiosk.
- 5 **PullSFFoodExpiryDetails** - Meal expiry notification generated from SmartFridge.
- 6 **PushSmartFridgePurchase** - Meal expiry notification generated from PoS/Kiosk.
- 7 **PaymentService** – Handle the user payment and related accounting updates.
- 8 **PostFeedback** - User sends feedback on the purchased food items.
- 9 **ApplyPromotionalOffer** - Push meal replenishment information from PoS/Kiosk.
- 10 **SetSFPromotionalOffers**– Admin can set Promotional offers to be applied on a particular smart fridge & item
- 11 **BookOrder** – Manage the Order Booking from PoS/Kiosk
- 12 **CancelOrder** – Manage the Order cancelling logic.

# **Architecture Decision Register**

## Key Architecture considerations

- Minimum initial investment in infrastructure
- Minimum changes to the vendor ecosystem. Which ensures no vendor lock in, loosely coupled architecture



# 1. # Micro / Mini Service Architecture Style with containers for external ecosystems

## ## Status: proposed

## ## Context :

Business requirements includes multiple business services, which need interaction with external ecosystems via API for 3rd parties. Many of these services are independent and have different scalability and availability needs. Thus application handles requests (HTTP requests and messages) by executing business logic; accessing a database; exchanging messages with other systems; and returning a HTML/JSON/XML response

## ## Decision :

Decision is taken to define architecture that structures the application as a set of loosely coupled, collaborating micro / mini services which will interact real time with smart fridge , Kiosk PoS, Payment gateways etc.

## ## Consequences:

Each service is:

- Highly maintainable and testable - enables rapid and frequent development and deployment
- Loosely coupled with other services - enables a team to work independently the majority of time on their service(s) without being impacted by changes to other services and without affecting other services
- Independently deployable - enables a team to deploy their service without having to coordinate with other teams
- Capable of being developed by a small team - essential for high productivity by avoiding the high communication head of large teams

## 2. # Asynchronous integration with Style with back-office system

**## Status: proposed**

**## Context :**

In addition to the external integrations, system also needs to integrate with Chef Tech and Quick Books. However the real time availability of data is required. Hence architecture needs to focus to optimize the operational cost and for maintainability and reliability.

**## Decision :**

Decision is taken to implement async integration using queues for Quick Books and Chef Tech

**## Consequences:**

Selection of suitable queue-based integration platform based on standard technology stack aggregable in the organization

### 3. # Cloud deployment with containers

#### ## Status: proposed

#### ## Context :

Business requirements includes multiple business services, which need interaction with external ecosystems via API for 3rd parties. Organization is new and has evolving needs , thus setting up data center will be costly. Internal systems used e.g. Quick books are also cloud based. Services involved have varied , availability , reliability and performance + scalability needs

#### ## Decision :

Decision is taken to opt for cloud-based deployment with containers

#### ## Consequences:

This approach will need selection suitable Cloud partner considering current organizational set-up and vendor connect  
Selection of suitable container and orchestration ecosystem based on cloud platform selection , standard technology stack aggregable in the organization

## 4. # Reserve meal option on mobile / web not available for Smart Fridge

**## Status: proposed**

**## Context :**

If a customer wants book a meal on mobile / web, reserve it for Fridge pick up. It will have following architectural impacts as Fridges are not attended by any employee / other human being.

- All bookings from mobile / web need to be integrated back real time to smart fridge to reserve meal and deplete inventory
- Hardware / software-based controls need to be enabled in fridge to avoid accidental pick up of this meal other customers

As per inputs given by customer, current smart fridge ecosystem needs some improvement to achieve this sophistication. This may result in multiple complex real time interactions between systems . Also it will also lead to dissatisfaction for the customers visiting Fridge in case someone else has picked up the meal by mistake.

To avoid this it was decided to provide "Track my meal" functionality as against "Reserve my meal" for smart fridges

**## Decision :**

We will not block the order but just track the meal for Fridge. Only availability will be shown to user for Fridge. User can go to the location and pickup item. No purchase/blocking for the meals via web/mobile app for smart fridges. Instead a map view of smart fridges and availability of meals at each fridge can be shown user to check higher probability of available meals.

Meals can be blocked for pick up from Kiosk or be delivered using existing delivery network.

## 4. # Reserve meal option on mobile / web not available for Smart Fridge

### ## Consequences:

This approach will need following component

1. Map view to show fridge and meal count
2. Integration with map service to provide fridge location
3. service to check availability of meals for the fridge

## 5. # QR code scanners at kiosk for reserved meal pickup

**## Status: proposed**

**## Context :**

When the meal is booked online using mobile app or web, user needs to affirm order at the time of pick up. Since many users will be visiting kiosk during same time duration (e.g. lunch time) and will have limited time availability, validation of meal needs to be quick with less human involvement

**## Decision :**

PoS/Kiosk to have QR code scanners. Users who have purchased meal via mobile app can scan the QR code received on their mobile app and pickup their order. Employee will handle this transaction at PoS

**## Consequences:**

Hardware installation for QR code scanner will be required

## 6. # SMS based order confirmation to support Low tech user base

**## Status: proposed**

**## Context :**

One of the business requirement is to be able to support the user base which does not have smart phones. For such users, it is possible to book meals on web sites, but they may not have smart phone to carry QR code for meal verification

**## Decision :**

User can opt to receive sms for the order confirmation, which can be shared at the Kiosk to collect the meal

**## Consequences:**

Integration with sms based service will be required

This may partially slow down delivery process as the confirmation number will need to be manually keyed in at the kiosk.

## 7. # Payment collection processing done at Farmacy food side.

**## Status: proposed**

**## Context :**

Since Farmacy Food is relatively new in business and has expansion plans, system is expected to undergo considerable transitions in near future. Leveraging / modifying payment service at smart fridge / kiosk will develop hard coupling / dependency with the vendor ecosystem

**## Decision :**

Payment processing done at Farmacy food side. Commission/rent processing for Fridge/Pos is out of the scope

**## Consequences:**

Integration with payment gateway need to be developed to collect payment



Thank You