

# Road Warriors- Online Trip Management: Application Architecture

## Contents

Overview .....	2
Assumptions .....	2
Requirement Analysis .....	3
Actor and Components .....	5
High-Level Design .....	6
Application Architecture .....	6
Data Storage .....	6
Other Considerations .....	6
Design Diagrams .....	7
Use Case .....	7
Traveler .....	7
System .....	8
Sequence .....	9
User Flow .....	9
Trip Info .....	10
Email Scanner .....	11
Trip Data Processor .....	12
Trip Data Updater .....	13
Hosting .....	13
Other Services .....	14
Redis Cache .....	14
Key Vault .....	14
Application Insight .....	14
Monitoring and Alerting .....	14
Data Analytics .....	14

## Overview

A new startup wants to build the next generation online trip management dashboard to allow travellers to see all of their existing reservations organized by trip either online (web) or through their mobile device.

This document summarizes high level application architecture and various design decisions taken for the purpose of creating a modern and robust application architecture.

## Assumptions

- Only preidentified agencies would be supported:
  - o API integration and setup is a prerequisite.
- User can choose preferred agencies from preidentified agencies only

## Requirement Analysis

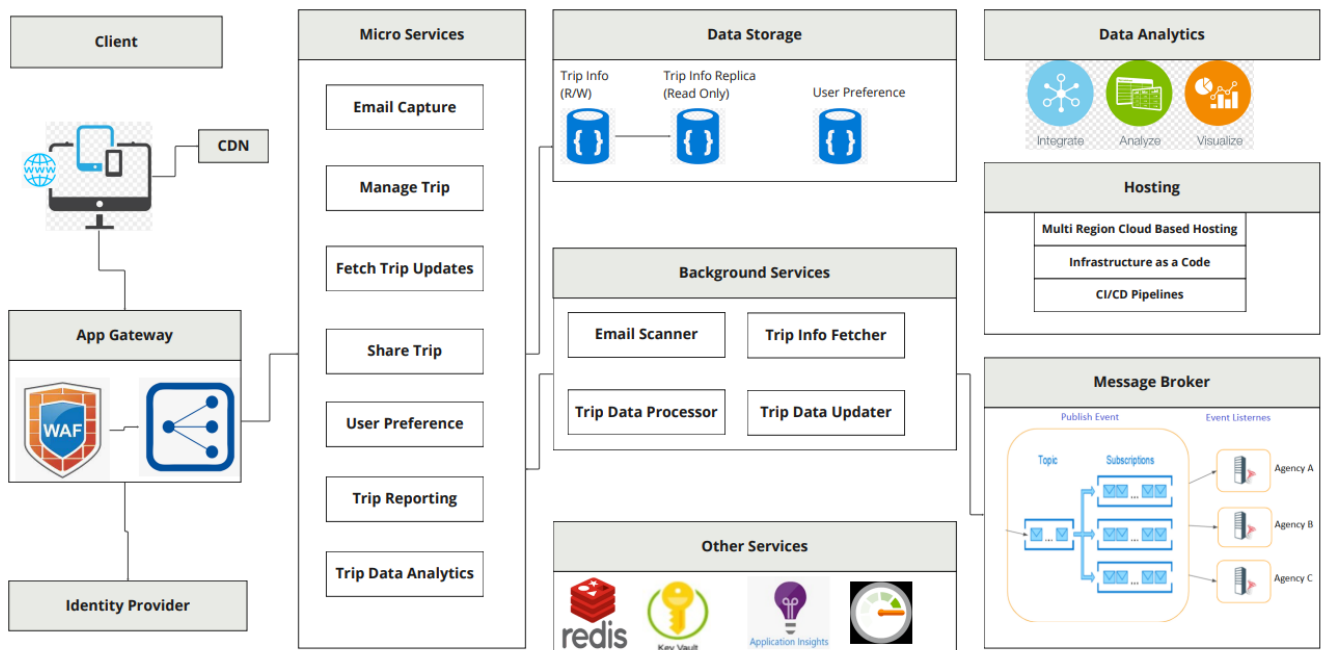
Type	Requirement	Task	Architecture chrematistics
Functional	1. Trip Management Dashboard (WEB/Mobile)		
		Manage Trip : Add/update/delete/View (group by) trip	
		Source of Trip: Scan Email for the trip (Booking ID/Type/Date/Agency, Pax Name, Destination, Flight/Hotel/Taxi info manual entry with booking (Identification Key)	
		Manage/Update Trip info: Third Party API call to sync the information against bookingId	
	2. Share the trip with Social platform integration	Integration with Social media platform to post-booking: Social Media Sign ->Share with -> Sign On->Consent->	
	3. RICH User Interface	for Web and Mobile both/ Responsive/Performant/UI/UX standards	
	4. Yearend report for users	Travel, Taxi, Hotel Stay- Summary for the year for the user (Graph, Matrices)- DASHBOARD	
	5. Gather analytical data for analysing behaviours analysis	Make sure all the required information needed for analysis has been captured	
	6. Must work internationally	Time Zone/ Localization, Geo hosting	
	7. Integration with preferred travel agency for help/support	List of preferred travel agencies to pick by user	
	8. Travel updates, must be presented within 5 mins		
Integrations	1. Email provider(filter only specific domain email)	Integration with email provider (AuthN, AuthZ)	
	2. Integration with third-party agencies and travel systems	To fetch the updates (any kind)	
	3. Integration with Social media platform to post booking		
NFR	1. Max 5 minutes downtime	Hosting environment downtime, Maintenance window	Availability, Reliability
	2. Travel updates, must be presented within 5 mins	Fetch details for all active bookings	Performance, Sufficient Infra support (Has dependency on API provider)

	3. Response time from web (800 ms) and mobile (1.4 sec)	Partial Loading, Async call, Data storage (Horizontal scaling)	Performance
	4. Total users: 15 million (user accounts), Active 2 Million per week	Support peak and seasonal load	
	5.Data security	Data regulations	Security
	6.Support Region Failure		DR
	Implicit		Responsiveness
			Elasticity
			Deployability
			Evolvability

## Actor and Components

S.No	Actor	Role	Actions	Components	Service
1	Traveller	End User	<ul style="list-style-type: none"> <li>-Email sync setup</li> <li>-Scan consent/Authorization</li> <li>-View Dashboard</li> <li>-Add booking entry (manual)</li> <li>-Share trip detail on Social Media</li> <li>-View yearly report</li> <li>-Opt for preferred agency</li> </ul>	Email Capture, Trip Dashboard, Manage Trip, Social Media /Posting, Trip Report, Agency Preference	<ul style="list-style-type: none"> <li>-EmailCaptureService</li> <li>-ManageTripService</li> <li>-ShareTripService</li> <li>-TripReportingService</li> <li>-UserPreferenceService</li> </ul>
2	Software System (Dashboard App)	Dashboard App	<ul style="list-style-type: none"> <li>-Integration with email provider, Agencies, Social Media apps</li> <li>--Email Scanning to capture data</li> <li>--Fetch booking updates</li> <li>--Post to social media</li> <li>-Globalization support</li> <li>-Support mobile and Web front end ( with BFF support),</li> <li>-Apply analytics tool for data analysis</li> </ul>	Email Scanner, Fetch Booking updates Social Media Integrator, Globalization, Data Processing	<ul style="list-style-type: none"> <li>-EmailScannerService,</li> <li>-FetchTripUpdatesService,</li> <li>-TripPostService</li> <li>-TripDataAnalyticsService</li> </ul>
3	Data Analyst	Data Analysis	<ul style="list-style-type: none"> <li>-Required data Analysis from the generated reports and metrics (use the data for the purpose)</li> </ul>	Data Analysis	

# High-Level Design



## Application Architecture

Microservice architecture was chosen for the following reasons:

- Autonomous
- Automation
- Business Domain Centric
- High Cohesion
- Observable
- Resilience

## Data Storage

No-SQL database was chosen for the following reasons:

- Horizontal Scaling
- Distributed storage
- Flexible/Schema less
- Rapid development
- High Performance

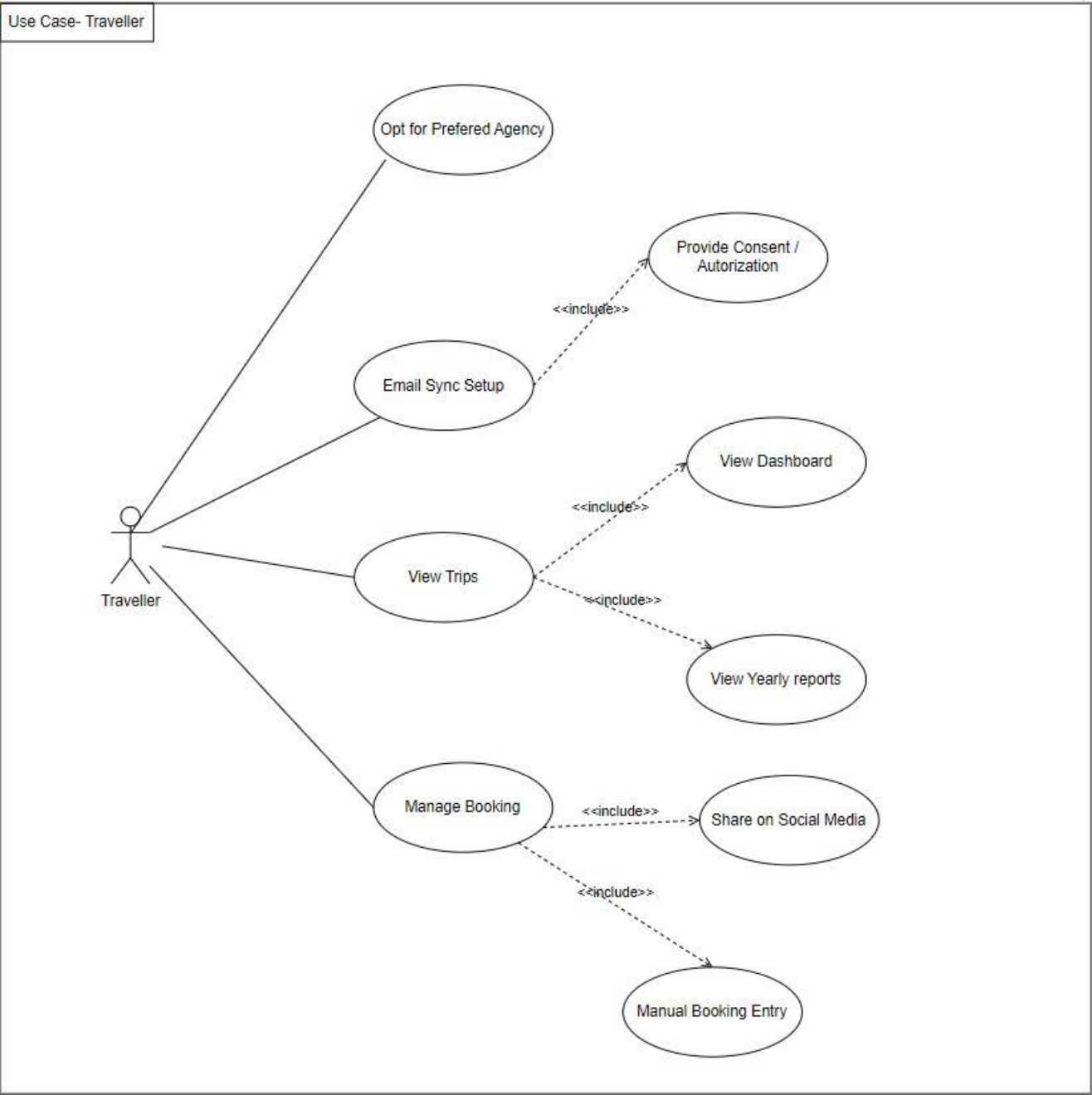
## Other Considerations

- Progressive UI for
  - o Rich user interface
  - o Support multiple devices.
- Globalization and multilingual support.
  - o Identify user locale and display information in respective language

# Design Diagrams

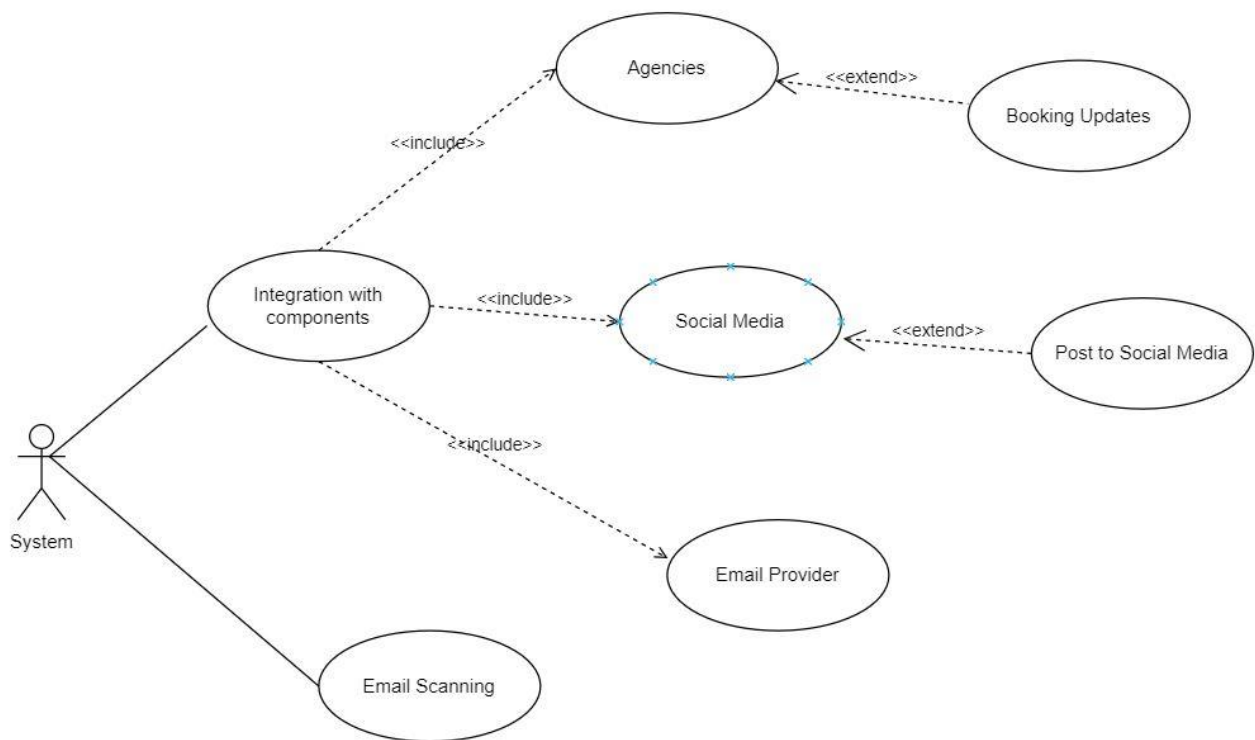
## Use Case

### Traveler



## System

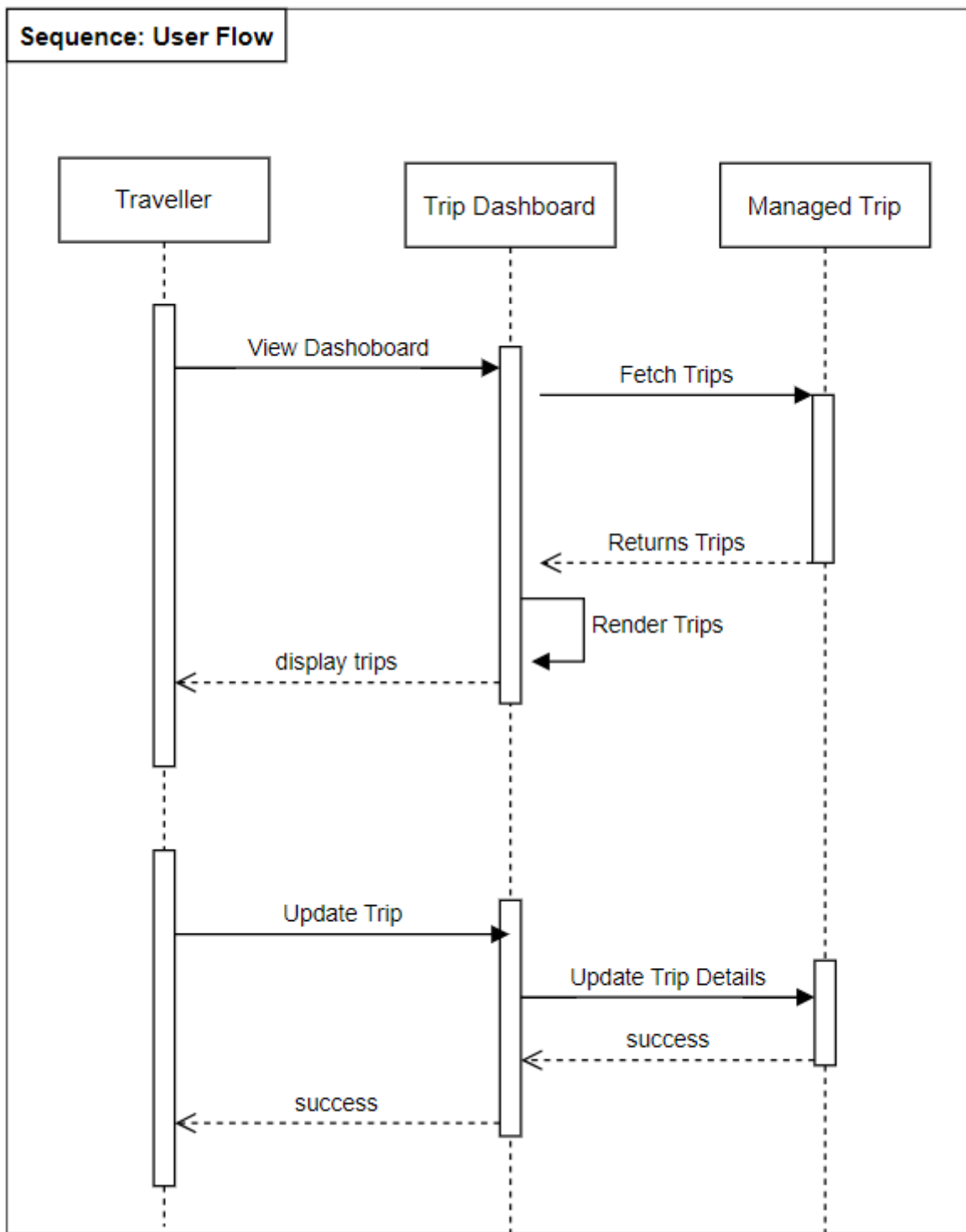
Use Case - System



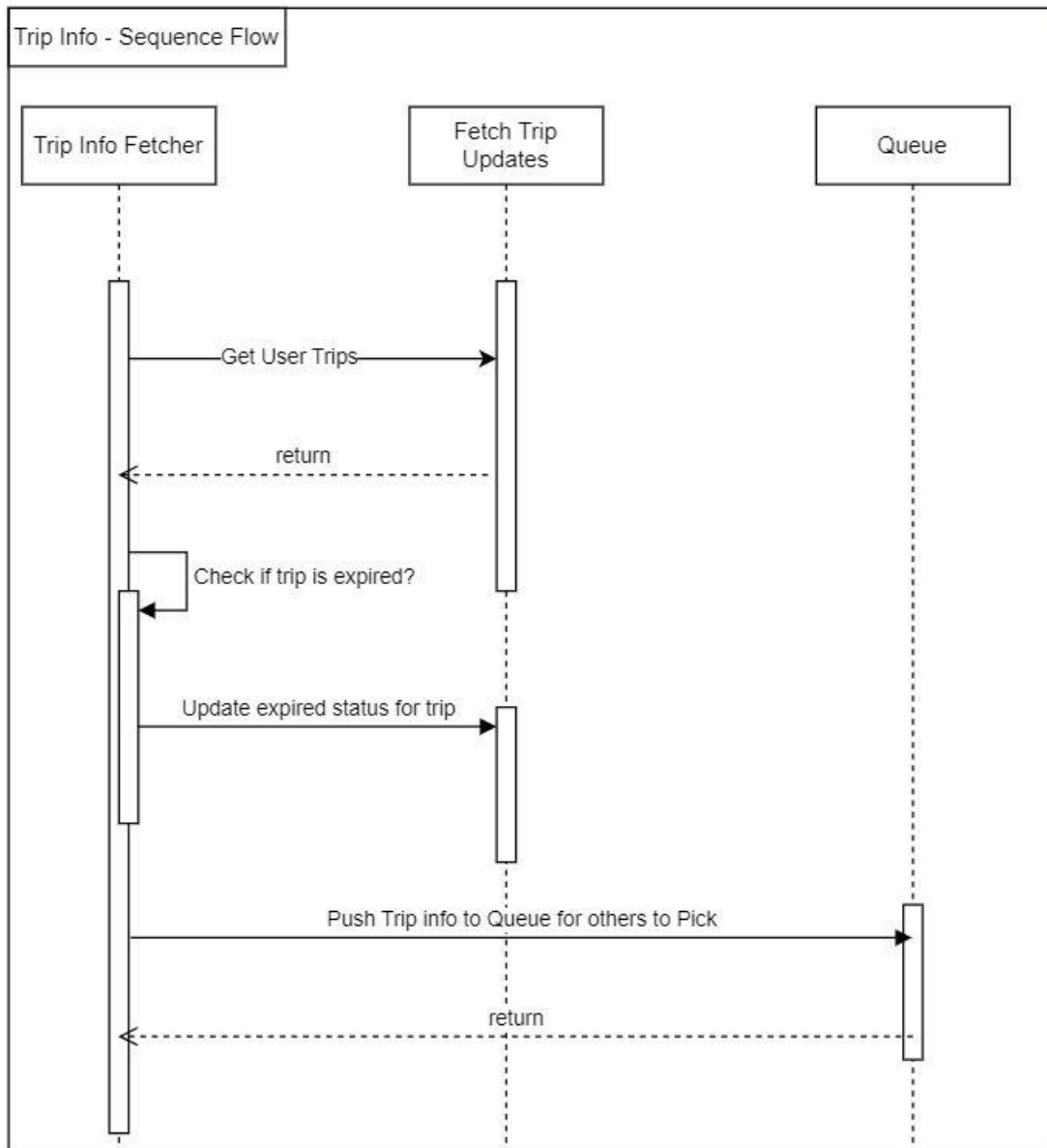


## Sequence

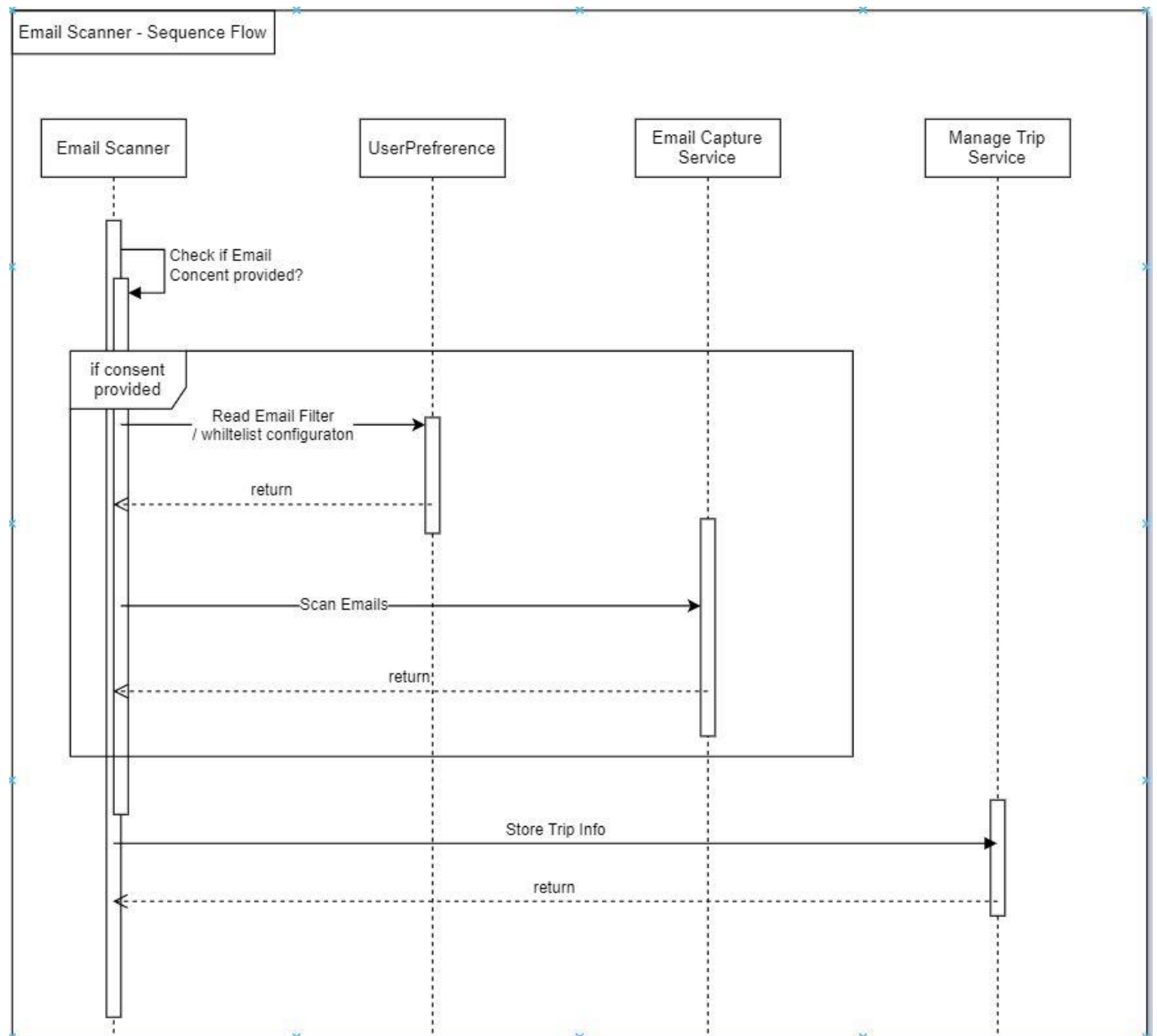
### User Flow



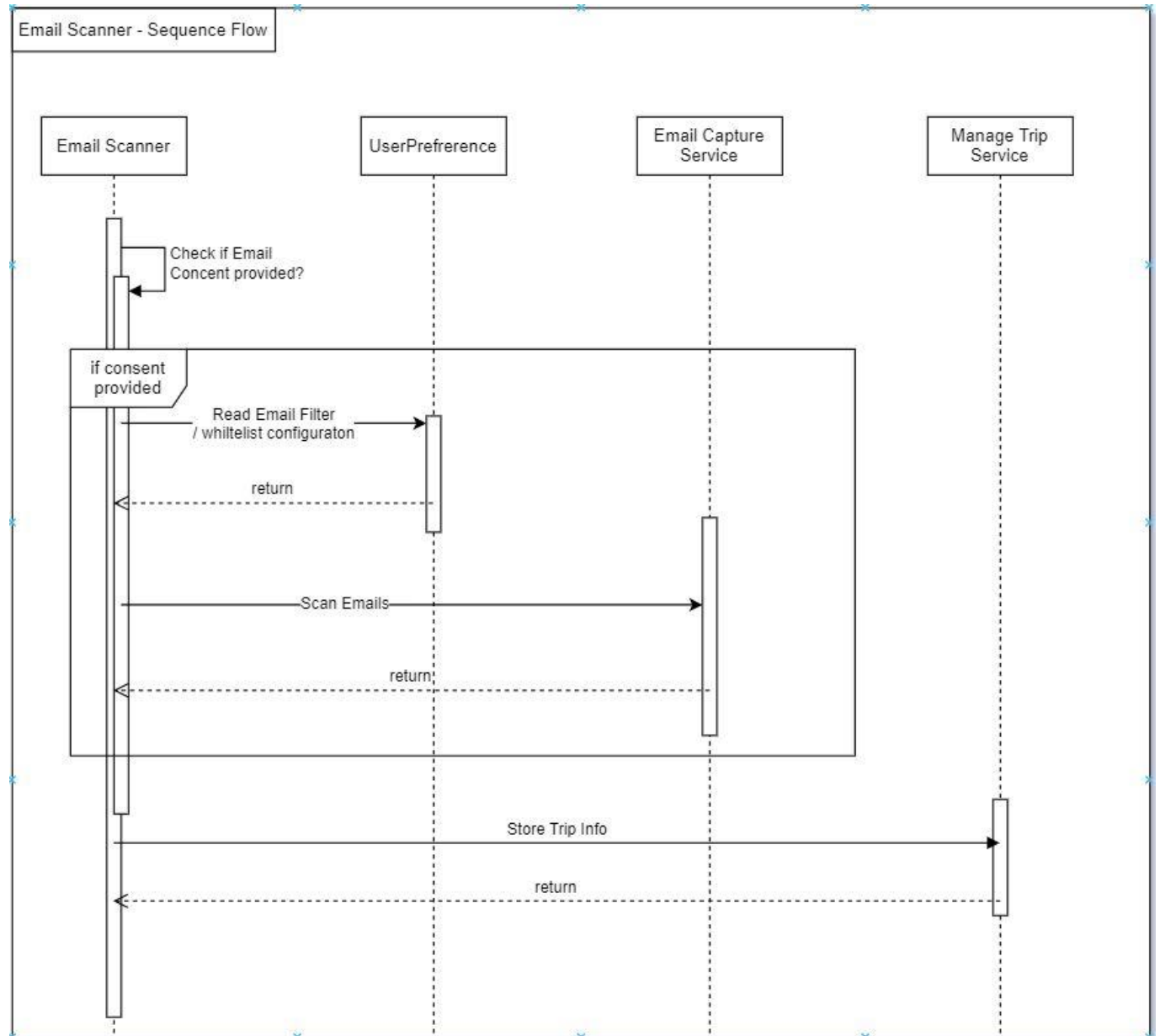
## Trip Info



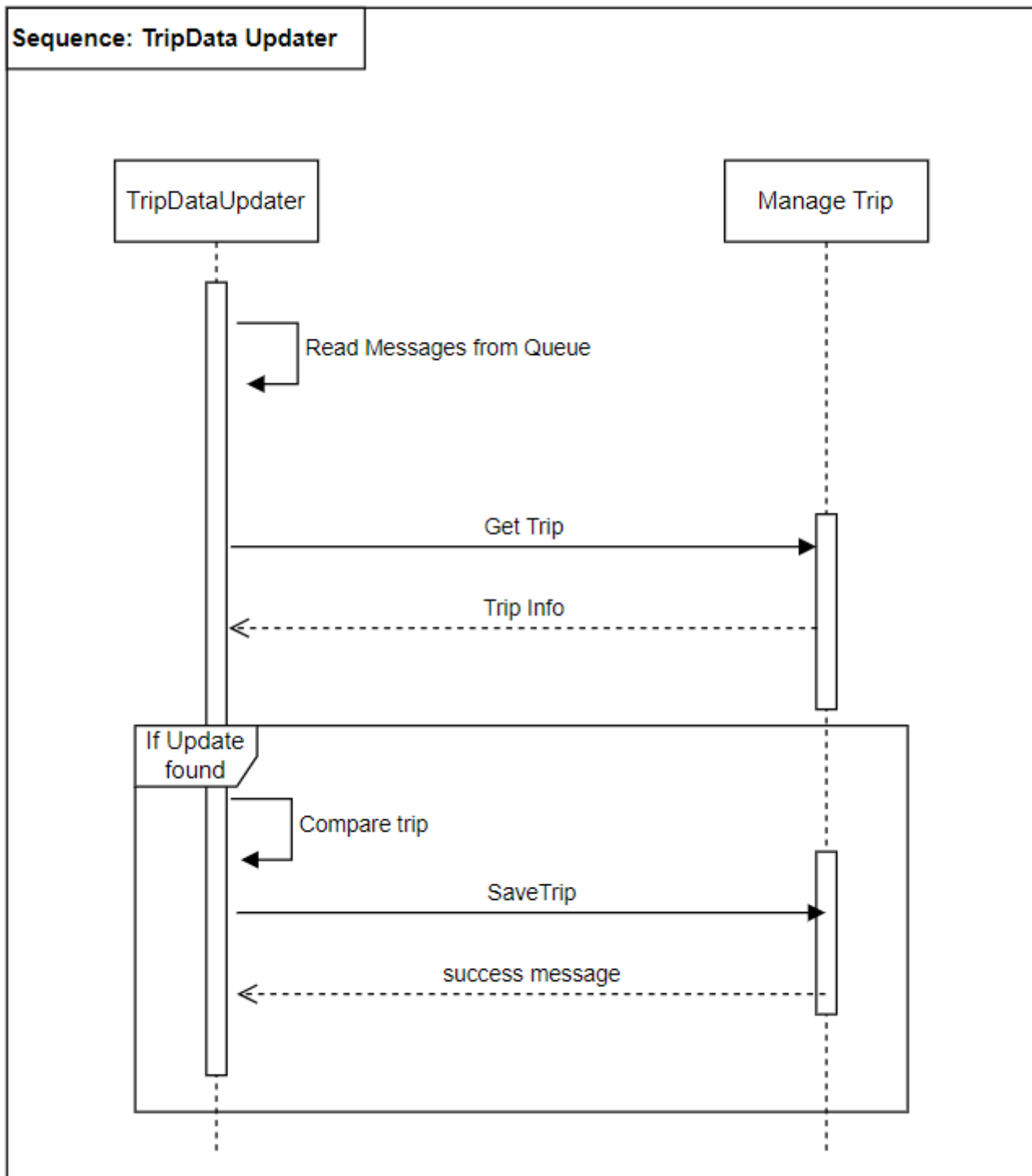
## Email Scanner



## Trip Data Processor



## Trip Data Updater



## Hosting

- Cloud based hosting to take advantage of the platform such as scalability and availability.
- Infrastructure as a Code for seamless infrastructure provisioning
- Automated build and release pipelines to support Continuous Deployment and Continuous Integration
- DTAP environment
  - o Slot swapping
- Rollback strategy

## Other Services

### Redis Cache

Enable caching capabilities to enhance performance.

### Key Vault

Store sensitive information Securely.

### Application Insight

Log usage, events, and traces that can be used to find inner details and to identify root cause of the issue

### Monitoring and Alerting

Application Usage and Infrastructure monitoring

### Data Analytics

Extract useful information from data and make the decision based on the data analysis.