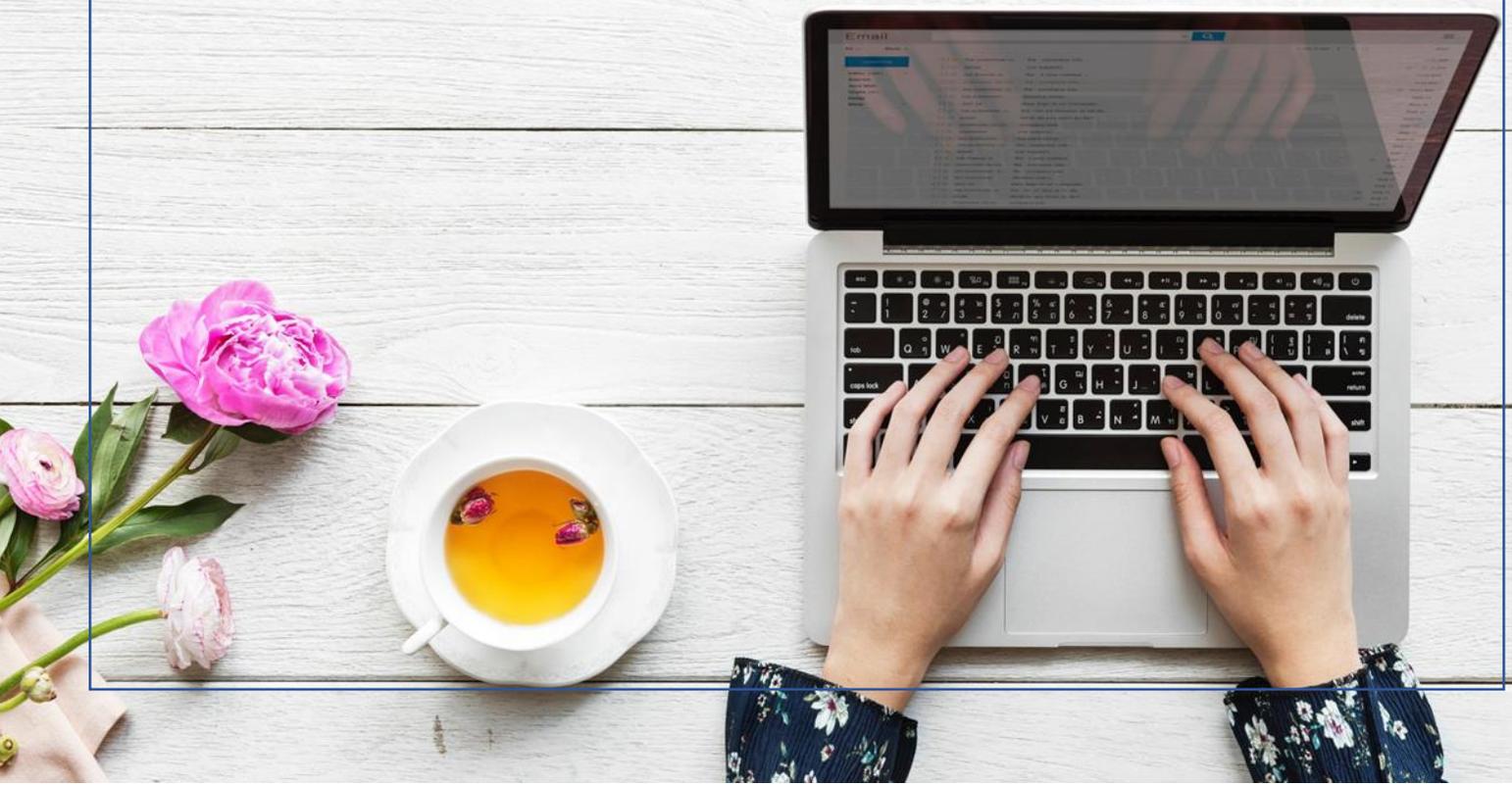


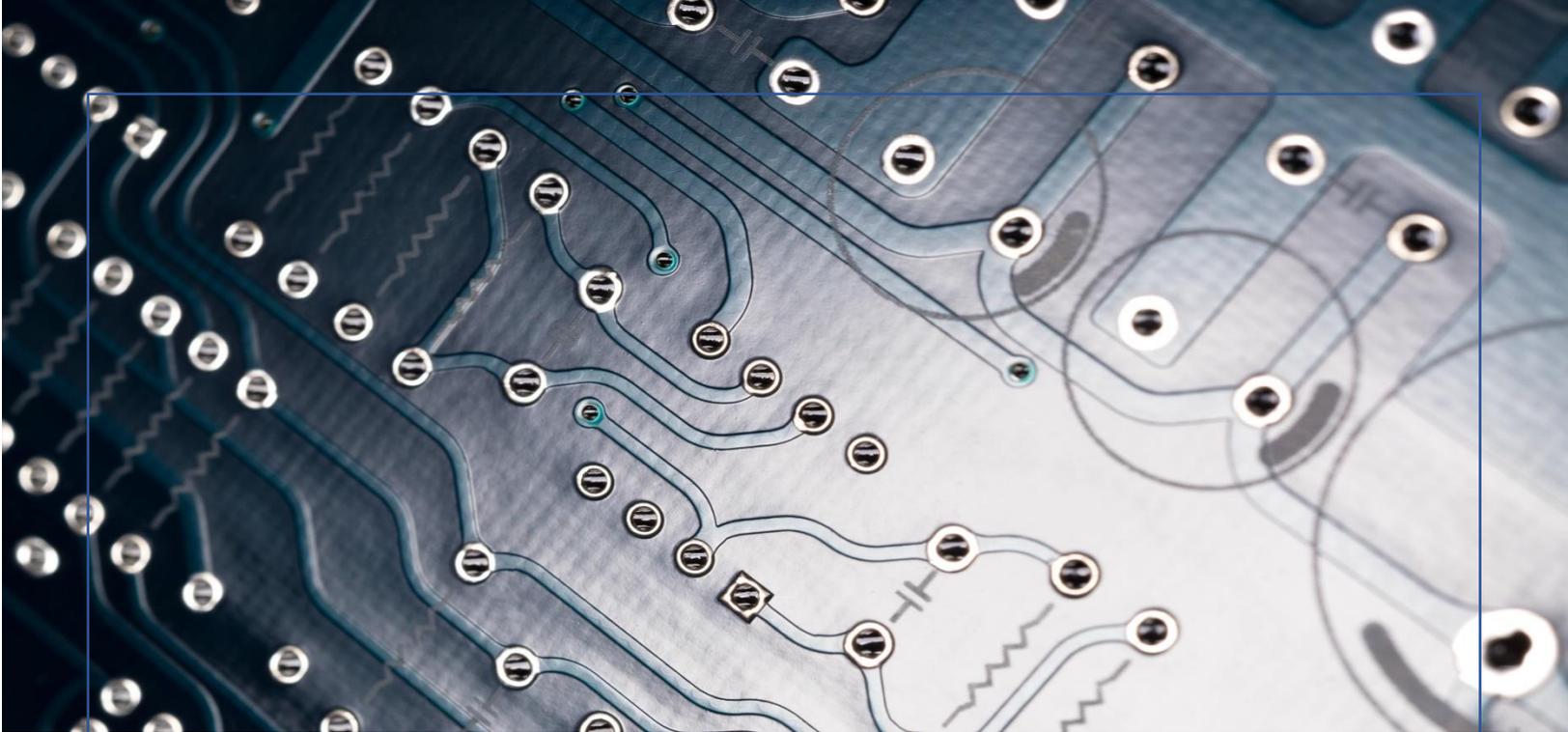
# Reality Premedia Services

Whitepaper - Tranquility Lifestyle Solutions IOT



# TABLE OF CONTENTS

- Introduction ..... 3
  - Abstract..... 3
  - Background ..... 4
  - Objectives ..... 4
  - Key Considerations ..... 4
- Solution ..... 5
  - Inputs ..... 5
  - Proposed Architecture ..... 5
    - Technology Stack ..... 6
    - Rationale ..... 6
  - Process & Team..... 7
  - Challenges and Workarounds ..... 9
  - Timeframe..... 9
  - ..... 10
- Conclusion..... 10
  - Key Takeaways ..... 10



## INTRODUCTION

### ABSTRACT

Reality Premedia Services has architected and deployed a state-of-the-art IOT cloud and mobility solution for a client that focusses on Geriatric care.

The solution is extremely fault tolerant, scalable and GDPR and HIPAA compliant. It is important to note that this solution, while being very performant, is also very cost-effective, which is a key factor in driving sales for the client. The solution caters to the B2C, B2B as well a Retail markets where devices can be bought off the shelf. The IOT devices used in the ecosystem do not need any special configurations or certifications, they are essentially plug and play. It also supports multiple modes of communications, from HTTP to MQTT to Web Sockets.

The mobile apps are quick, responsive and are available on multiple form factors (iOS Phone + Tablet, Android Phone + Tablet). The mobile apps also contain cutting edge functionality like Inbuilt Video Calling, Zendesk Integration (including live chat), Direct Messaging, Calendar Synchronization etc. to name a few. Reality also helps the client manage all the infrastructure.

The cloud solution is “self-maintaining” with jobs in place to automatically sort, index, and backup any relevant information. It also moves data into different tiers based on frequency of access. Security, Administrative and Usage logs are also saved for posterity. For B2B clients, a self-service portal is available for administrators of different B2B business to provision and manage their own devices as well as users.

The solution also includes a Hub App that is installed on the Hubitat hub that sets up a secure tunnel between the hub and the cloud service. A similar solution has also been setup for the Vayyar devices, which do not directly communicate with any Hub and can act as a standalone device.

## BACKGROUND

The client is a lifestyle solution company, founded in 2018 in New Jersey, that focuses on lifestyle solutions for the elderly. They set out with the objective to make senior living more comfortable and affordable. The long-term goal is that this can be reused for Pet Care, Infant care, and other similar domains.

The way the client wanted to achieve their prime objective is by selling a service where they setup the senior's home with sensors that can detect when something irregular happens and then the "caregiver" or "family member" would get the alert on their app. This was part one of the plan. The second part was to approach senior living homes and sell this service to those businesses. The second phase of the plan involves integrating with a Neural Net to process the available to predict events before they happen.

Aside from the plug and play retail devices, the client partnered with two other hardware providers who have revolutionary solutions in the form of 4D radars. One is Vayyar (for Fall Detection) and the other is Neteera (for Vitals Monitoring), both of which are non-intrusive devices that are "setup and forget".

The client reached out to Reality Premedia Services through another of Reality's clients and sought technical guidance as well as development services.

## OBJECTIVES

- The solution must be plug and play and be compatible with almost any retail device
- The solution must support multiple licensing models
- The solution must be self-serviceable
- The solution must be intuitive to the user
- The solution must be configurable by the user to set his/her day/night sleep schedules
- The notification of a "fall" or other event that required attention should be immediate (<1000ms).

## KEY CONSIDERATIONS

- Cost per device / Cost per user
- Infrastructure Performance and Scalability
- Notification Speed
- Compliance (GDPR & HIPAA)
- Data storage for Neural Network
- Information Security
- Non-Intrusive

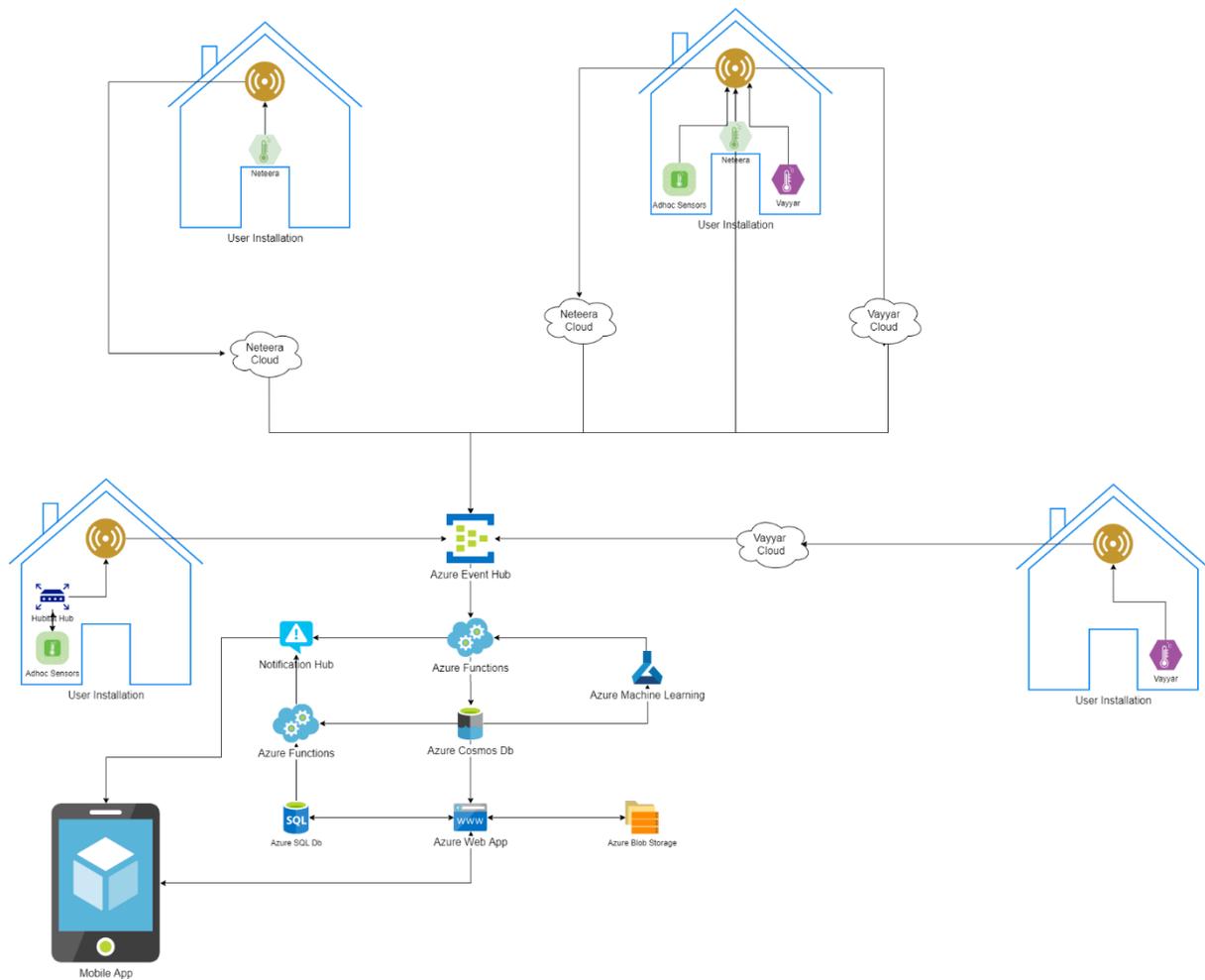
# SOLUTION

## INPUTS

The main input received by Reality was a huge number of mind maps on the end user experience and proposed functionality. The UI was put in place by a designer that was already hired by the client before they approached us.

## PROPOSED ARCHITECTURE

Below is the architecture that was proposed and implemented.



## Technology Stack

The technology stack used was as follows.

- Mobile Apps – Xamarin Forms
- Backend API – ASP.NET Web API 2.0 – Web App
- Backend Admin Console – ASP.NET MVC 5 – Web App
- App Video calling – Twilio
- App Messaging – Twilio
- Helpdesk Provider – Zendesk
- IOT Cloud Hub Endpoint – Azure Event Hub
- IOT Rules Processing – Azure Functions
- Content Data Storage – Azure Blob
- Admin Console – Azure SQL
- Data Backup – Azure Blob Storage – Using Azure Data Factory
- IOT Event Sink – Azure CosmosDb
- IOT Hub for Adhoc Devices – Hubitat
- MQTT Sink – Azure IOT Hub with Azure IOT Edge

## Solution and Rationale

The time to go to market was very short and we knew that the maintenance turn-around time were also going to be very low as we needed to capture the market quite quickly. We decided on using Xamarin Forms with a good level of customization natively on Xamarin iOS and Xamarin Android. This also allowed us to push out multiple products in the app store using the same underlying code structure and framework. Liberate and Embrace, both are essentially the same app where we have used compile time constants and configuration to “build” two apps from the same code base.

The backend system for administration was a no brainer. We went with MVC 5 with Web API 2 leveraging most of the out of the box functionality and building around it. This was the quickest way to get up and running considering we are a .NET house.

The core functionality of getting sensor data and performing actions on the cloud was the trickiest part in terms of constraints. The main consideration was the volume of data and the latency involved in completing a cycle. The industry standard for this is MQTT. Azure has an IOT hub that allows devices to communicate via MQTT. Unfortunately, Hubitat (at the time of development) does not support MQTT. So, instead we used the azure event hub that allows you to submit packets via http. We then wrote an app to install on Hubitat that used a security token and pushed data to the event hub. Processing of this data also could not be done with a standard scheduled job. We needed it to be immediate. So, we used Azure functions to write snippets for processing.

The Event Grid is not permanent storage, and for some rules, we need data of the last few days. SQL performs extremely badly when there are multiple writes. SQL does very well for multiple reads and single writes. We used CosmosDb due to its scalability and multi region write performance.

## PROCESS & TEAM

The team consisted of two backend developers with an additional cloud architect to setup both the management console as well as the cloud rules engine. On the front-end side, we had two Xamarin developers working on the mobile applications. From the PM and QA standpoint, the teams also contained, three QA resources along with a Dev Manager / PM and one BA.

We follow Agile Scrum internally for all projects and this project is no exception.

We used Azure DevOps as our code repository (using the underlying git support). All user stories, features and bugs were tracked on azure boards along with all sprint management features.

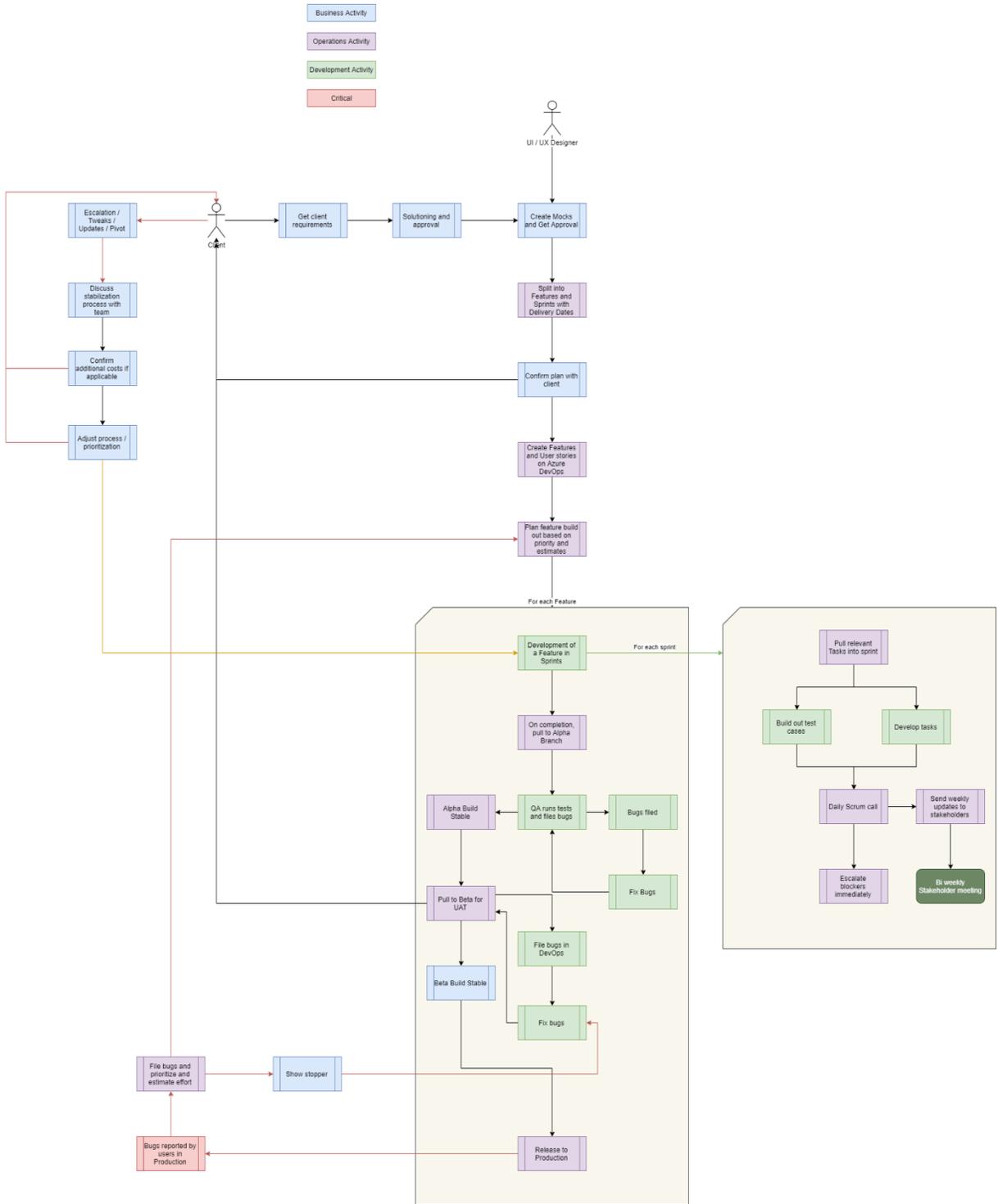
All deployments and builds are completely automated using the Azure CI/CD pipelines which deploys both mobile apps as well as to the azure cloud.

Automatic crash reporting as well as distribution of the mobile app is controlled and done via Microsoft App Center.

Bugs and user stories are “auto resolved” based on links created with user code check-ins and alpha pulls, instead of developers manually resolving items.

All pull requests made from a feature branch need to be code reviewed by the architect before it is approved and integrated.

We also have a stakeholder or “Principals meet” every alternate week to realign on progress as well as discussing future changes and strategic goals.



## CHALLENGES AND WORKAROUNDS

### TWILIO VIDEO CALLING

We have used several Twilio services in this project. Most of them have binding libraries for Xamarin, but the Video call add-on specifically does not have any Xamarin support. We did try another provider that supports Xamarin (Vidyo) but the support and stability wasn't very good and we had to go back to Twilio. Based on that, we set out to build our own add-on for Xamarin.

We created a Xamarin iOS and Xamarin Android Twilio library from scratch. This library in turn was bound to a Xamarin Forms application. This was a self-contained NuGet package that could be distributed with ease.

### BULK DATA STORAGE

One big concern while running the entire application was cost. The cost per device was very large. The main contributor to this was CosmosDb. While necessary and very efficient, the read/write for this module was very high and was driving up cost by quite a bit. We went for one round of enhancements where we cleaned up the data, but that was not sufficient. We then switched to a TTL expiration instead of a hard delete on legacy entries. This saved us a decent chunk of costs, but we pushed further. Instead of using Azure functions to selectively transfer data from CosmosDb to our data dump, we switched to using Azure Data Factory and setting up rules within it for data migration. This was pivotal. As a result of all these tweaks, we had about 80% savings in cost per device.

### TIMEFRAME

The initial implementation for us to go live with Liberate was seven months. Following that, we worked on enhancing that app with B2B features (Embrace). This was done over the next two months.

Over the subsequent months, we went on to create a third application as well as take over the entire computation and cloud capabilities from one of the sensors (Vayyar) systems, including provisioning a device via Bluetooth etc.



# CONCLUSION

Reality Premedia successfully designed and executed an end-to-end IOT solution for Tranquillity Lifestyle Solutions with minimal per device cost and a highly scalable ecosystem that is GDPR and HIPAA compliant. It is also highly flexible which minimizes the amount of effort to configure any new sensor that is released in the market.

## KEY TAKEAWAYS

- Highly Scalable Ecosystem
- Cross region compliance
- Extremely low running costs
- Highly performant with up to 500ms error to notification speed.