

# Mobile Apps Sample Solution

## 1 PREFERRED TARGET AUDIENCE

---

Operations Management Team led by Christopher Giovanni, Chief Operating Officer at Crazy Taxi Cab Co.

## 2 PREFERRED SOLUTION

---

Crazy Taxi Cab Co. liked the idea of a highly scalable, agile solution that they could both execute on and manage with a small IT team with limited back-end development experience in .NET. The company's Microsoft representative introduced them to Azure Mobile Services, a cloud based mobile platform that provides backend CRUD data services, offline data sync for mobile devices, custom REST API services, push notifications and services for login via a social identity provider. This was exactly the "back-end in a box" that they were looking for.

Crazy Taxi Cab Co. leveraged the many features of Mobile Services available to them with the Standard tier in order to minimize their backend development burden and accelerate solution delivery.

The implementation of the proposed Azure Mobile Services solution would create a strategic partnership that will help Crazy Taxi Cab Co. to overcome its challenges with:

- Minimizing system downtime
- Sending multiple user specific notifications to mobile devices
- Managing user accounts leveraging social identity providers, such as Microsoft Account, Facebook, and Twitter
- 24/7 (secure) data accessibility throughout the Crazy Taxi Cab Co. network
- Scalability in software solutions in an agile marketplace

As the FastRide system continues to be improved upon, Mobile Services will help to inject velocity into the development cycle by providing a mobile back end to the application. Mobile Services offers cross platform compatible components, which gives Crazy Taxi Cab Co. the flexibility to change their mobile platform as the needs of their business dictate. This "back-end as a service" approach will allow Crazy Taxi to focus on building an app that merges the right functionality with a great user experience for each market the operate in.

By utilizing push notifications, Crazy Taxi Cab Co. can optimize their customer pickup messages through the FastRide app. This allows for faster, and more streamlined in-app communication. When a new fare's pickup address is entered into FastRide by a dispatcher, or the customer facing mobile app, Mobile Services will enable FastRide to automatically send a notification to the closest available driver—eliminating the need for manual notifications. Since push notifications can be managed, each base will have control over the messages sent to its drivers.

To propose a more complete solution and ensure deployment success, it would be helpful to know:

- Type and operating system of tablets being used
- Expected product life of current tablet choice
- Current number of dashboard tablets
- Projected number of tablets after the planned expansion this year and next year
- Current average number of fares completed per day, week, month, year
- Projected average number of fares completed per day, week, month, year after the expansion into new markets
- Rate of growth across bases (customer and driver)
- Other software products used to operate the company

Understanding these details and decisions will help identify the current and future software, hardware, and infrastructure needs of Crazy Taxi Cab Co., and to provide solutions that are consistent with their short and long term business goals.

## 3 HIGH-LEVEL ARCHITECTURE

---

Crazy Taxi Cab Co. leveraged the many features of Mobile Services available to them with the Standard tier in order to minimize their backend development burden and accelerate solution delivery.

### 3.1 AUTHENTICATION

Drivers login to the FastRide app on their device using their Microsoft, Google, Twitter or Facebook credentials, the federated login process being handled by Mobile Services in conjunction with the aforementioned identity providers.

### 3.2 NOTIFICATIONS

Once logged in, the app registers with Mobile Services, associating the driver's identity with the Notification Hub (associated with the Mobile Service). In this way, Crazy Taxi dispatch can send broadcast notifications to all drivers, but still be able to send targeted Fare Alert messages to a particular driver

By having Mobile Services in place with Push Notifications, Crazy Taxi Cab Co. is well positioned in the future to light up the ability to deliver a customer-oriented app that deliver push notifications to customers informing them of events relevant to their pickup.

### 3.3 OFFLINE DATA

For the driver's iOS and Android devices, in the construction of the FastRide app, they leveraged the native client Offline Data Sync functionality for synchronizing Fare Data when temporarily disconnected from the cellular network. This Fare Data is stored using Tables in Mobile Services, which ultimately rests as relational tables in a SQL Database. This SQL Database also stores the driver profiles (that associate social credentials with driver profile information).

### **3.4 BACK END CUSTOM SERVICES**

When a driver accepts or declines a fare received via a push notification, that message is sent using a Custom REST API hosted by Mobile Services and built using ASP.NET Web API.

### **3.5 FRONT-END WEBSITE**

Dispatch uses a website hosted in Azure Websites to manage the taxi cab dispatch process. The Notification Hub is configured so that only the dispatch website is allowed to send push notifications to the drivers (the FastRide app for drivers is Listen only).

### **3.6 MONITORING**

Crazy Taxi corporate IT monitors the health of the solution using the Dashboard for the Mobile Service or Website in the Azure Portal. To assist the IT team with visibility into the health of the system, they should configure monitoring endpoints, again using the Azure Portal, on their website and mobile services and enable e-mail alerts should the Response Time and Uptime metrics for those fall below values they deem acceptable.

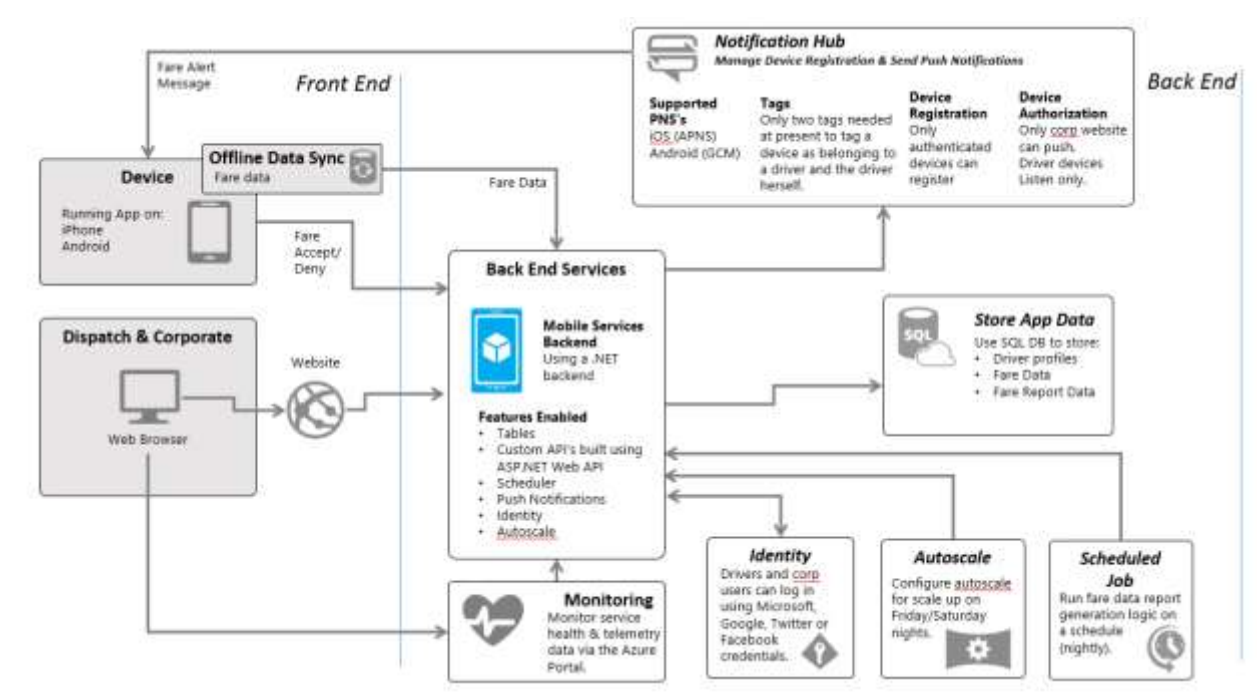
### **3.7 SCALING CONFIGURATION**

They have configured Autoscale on their Mobile Service, via the Scale tab in the portal, to double the number of instances it uses on Friday and Saturday night in order to handle the increased load, then return back to their normal instance count for the rest of the week.

### **3.8 BACKEND JOBS**

They have also created a Mobile Services Scheduled Job that processes the Fare Data on nightly basis to generate the data sets that power the Fare Reports. This data is stored in the same SQL Database that stores all the other data used by the solution.

## 4 PREFERRED SOLUTION DIAGRAM



## 5 CHECKLIST OF PREFERRED OBJECTION HANDLED

### 5.1 GENERAL

#### Doesn't Azure Mobile Services only work on Windows devices?

- Azure Mobile Services provides native clients for iOS, Android, Xamarin, PhoneGap, Sencha and Appcelerator in addition to Windows universal C#, Windows universal JavaScript and Windows Phone. In addition, Azure platform services offer REST APIs that extend the reach to platforms for which there is not a native API, but are capable of making REST style requests.

#### Our development team doesn't know node.js. We had heard mention of Mobile Services, but thought it only supported JavaScript backends.

- Mobile Services supports using .NET for the backend logic, and node.js (or JavaScript logic) does not have to be used anywhere in the backend code.

#### Our development team seems to think implementing push notifications using Apple and Android apps directly is easy, but we (as the executives) aren't so sure. How difficult can it be?

- While using the Push Notification System of a particular device platform directly is typically made fairly simple by the provider of that platform (e.g., iOS apps have a straightforward API for leveraging Apple's Push Notification System), this simplicity is only true for the individual device and does not address the complete solution that requires at minimum a

backend managing device registrations at scale and sending push notifications cross platforms in a timely fashion. Azure Mobile Services provides that backend functionality, which can be easily scaled to meet demand.

### **Can't we just build all of our backend using Azure Websites?**

- Azure Websites is effectively a superset of Mobile Services and so can be used to implement the backend for Mobile Applications. However, Websites do not deliver the services tailored for the mobile application scenario, requiring the developers to write their own logic to integrate with Notification Hubs, SQL Database, Identity services and WebJobs. Additionally, Mobile Services is prescriptive in the patterns used for developing custom API's, and so speeds the development of such API's by allowing the development efforts to focus on the business logic instead of dealing with structural and hosting decisions. These become important factors to consider when taking into account the development team size, capabilities and timeframe.

## **6 PROOF OF CONCEPT CHECKLIST**

---

The primary items a Proof of Concept for this solution could demonstrate include:

- Scalability / Scheduled Autoscale
- Mobile Services ease of integration (e.g., the backend in a box)
- Streamlined communication with Push notifications
- Integration of social identity platforms to aid in customer authentication and profile management
- Device offline data storage & synchronization
- Monitoring solution health

The Proof of Concept will be considered successful if the Crazy Taxi Operations Management Team believes they can realize value in:

- Speeding up the delivery of the overall solution
- Push notifications to streamline communication and send fare updates to tablet devices
- Authenticating users via social media platforms and future benefits of successfully leveraging social media integration.
- Minimizing system downtime by keeping app data in the cloud
- Scalability in a mobile cloud solution

The personnel resources you would leverage to build the PoC, may include:

- Partner Resources in the Region or MCS to help assist with migration design and implementation
- Microsoft Azure CAT for level 300 expertise requests with Azure