Challenges to Developing a Secure, Cloud-Based Offline Mobile Application

Sambit Kumar Patra[1(✉)] and Navin Kumar[2]

[1] Accenture Technology, Bangalore, India
sambit_sai@yahoo.com
[2] Amrita School of Engineering Bengaluru, Amrita Vishwa Vidyapeetham Amrita University, Bangalore, India
navinkumar@ieee.org

Abstract. The alliance with mobile device cloud computing technology promises new ways of developing business application. Using web and cloud technology, it is possible to transfer a small part of secured business data from cloud to the small storage mobile devices. However, it is challenging to keep secure data in offline mode when the web application is unable to connect to the cloud and sync those offline data at online mode. The paper discusses architecture of cloud-based mobile application. The proposed architecture helps us to develop cloud-based mobile application. Special consideration of low memory and network connectivity on the mobile device is taken into account. We present various challenges to design a cloud-based mobile application, store the data in a secured manner at offline mode and sync those at online mode. This would greatly improve enterprise productivity even when users are working offline.

Keywords: Mobile application development · Secure data transmission
Offline data · Cloud

1 Introduction

The fast evolution of commercial mobile devices has made the technology an essential requirement for the government, enterprise and commercial end users. Mobile technologies are transforming enterprises, industries, and the entire world [1]. Also, the growing need to produce new and innovative mobile applications which provide enhanced business capable government workforce, together with common capabilities like secure email, has led to the huge challenge of providing standardized solutions. The technology today offers the opportunities to drive business transformation through mobility.

Mobility is more invasive than ever before with increasingly rapid evolution of new devices and technology, placing demands and creating opportunities for enterprises and consumers around the globe [2]. Mobility provides better and faster decision making through improved access to key data and analytics capabilities anytime, anywhere. It also enables access to workflow tools on the job, reducing manual processes, supporting on-the-go secure operations, services and management. However, the future of mobility

N. Kumar and A. Thakre (Eds.): UBICNET 2017, LNICST 218, pp. 46–53, 2018.
https://doi.org/10.1007/978-3-319-73423-1_5

is in the cloud [3], but when a connection to the cloud is not available, the mobile user is out of the loop.

Now-a-days, many large and small businesses use cloud computing either directly (for example, Google or Amazon) or indirectly (Twitter) instead of traditional on-site alternatives because of various reasons such as cost benefits, universal access, flexibility, and so on [4]. But the predicament with mobile devices is that they are constantly mobile; consequently they tend to lose connectivity with the cloud. For example, if a customer is travelling, most of the time s/he would be out of the coverage for one or other reasons e.g., remote location or cell is down. So, s/he won't be able to update information such as editing an opportunity or updating a case because of the poor connectivity.

Modern businesses, their information systems and mobile devices cannot be expected to hold the data that users need at any moment of time. Any disconnection from network because of any problems such as high latency, low bandwidth or even presence or absence of the network may result in reduced productivity. To successfully moving and promoting the usage of cloud centric enterprise architecture, there would be important and necessary requirements to have support for offline data transfer [5]. Offline support for mobile applications is the ability for the application to react elegantly to the lack of stability in the network connection, high latency and even low bandwidth.

However, providing offline access is not easy: There are significant technical challenges and one of the reasons cloud-only has been pushed as the future is that offline was seen as being a bit complicated. There are many challenges to think of when we consider providing offline access to applications. The challenge is to determine what degree of offline operations is possible at any given time, maximizing productivity in working offline. The challenge is in security: what data is stored on the client, and how this can be compromised. Any offline access requires storage on the device. This means that the data could be read by malware or if the device is stolen, how mobile device and application management that is mobile device management (MDM) and mobile application management (MAM) will address this issue.

In this paper some of the challenges in offering offline support are discussed. We present them in brief and we also discuss our approach in addressing these challenges. A simple architecture is presented to describe the approach which would support offline data transfer in the situations like network unavailability, low bandwidth and so on. It is a new concept and idea. To the best of our knowledge, we have not found this in any literature.

The contents of the rest of the paper are as follows. Section 2 presents the challenges in brief. The architecture and proposed solution is discussed in Sect. 3. Section 4 presents the best practices which can be followed. The conclusion is given in Sect. 5.

2 Challenges

There are number of technical challenges in providing offline support. There are even challenges if developing application to support offline data transfer. It is possible for a browser to store resources on the local device, such as caching. But caching is normally used for speeding up page loads and reducing the bandwidth usage. Furthermore, on the

mobile device, we have limitation of data/cache. Therefore, there seems a no. of issues in offline support. We discuss some of them in this section and try to understand the development constraint.

2.1 Responsive Design

Irrespective of different mobile size and operating system, the user interface of the application should remain the same across the various platforms like Android, IOS, Windows, and Blackberry [6]. Also, the application should be responsive enough to manage both portrait and landscape mode for small devices like phone, tablet to bigger screen like TV.

2.2 Developing a Dynamic Scalable User Interface (UI)

Looking at the business needs, the application flow may change constantly and consistently, however, it should not impact to the application architecture. As redeveloping and redeploying process of the application on respective application store like Android, IOS and Windows takes some considerable time and are more expensive, the application architecture should be capable of handling the future business changes. A business based mobile application should be robust enough to create the pages dynamically, handle events, throw errors and manage huge number of data within the limited memory and battery constraints.

2.3 Accessing Native Functionality of Mobile

As the business applications are developing for the mobile devices, the applications are trying to enhance the functionality of the mobile device such as Call, SMS, Camera, Calendar, Notes. Irrespective of device constraints and OS, the application should be integrated with native mobile OS in such a way that it should be merged with those functionality when required. It will be more challengeable.

2.4 Storing Secure Data

The business data is sensitive. Transaction is normally conducted in online mode which is a big challenge to run such application in a mobile device due to the poor network connectivity especially while roaming and moving from one page to another. The cloud based mobile application should be designed such a way that it has the ability to store the data when the device is offline and upload the same at online mode.

2.5 Syncing Offline Data

Synching is another important challenge. During syncing, the device uploads the updated data into server and retrieves data from the server. In a large business application, sometimes same data of some module of the application are handled by multiple

users. The main challenge is updating the offline data into server. Again, due to poor network connectivity the updated data are stored in the device for a longer time. During this time, some other devices update the data into server. When the device becomes online, it uploads the offline data into server which is older than the data from the server.

2.6 Connect the Devices with Same Application Flavor

User can able to access the same application and data using his/her multiple devices from remote locations. The application architecture should maintain the same flavor across all the devices. It should consider the devices are connected to each other.

2.7 Personalize the Device

Application should allow user to customize the application with personalized data with his/her own device. It should not be shared to other devices. To personalize the device, user may like to change the theme, add his/her own photo, contact and reminders to the application.

2.8 Keep Personalize Data During Upgrade

Sometimes user may reinstall the application to the same device either he might have deleted the application accidentally or the newer version of the application has been upgraded. During this process, user should not lose his/her personalize data. After upgradation, the new application should maintain those personalize data.

2.9 Embedded with Data and Location Visualization

To take better decision with analytics, the data of the business application should be represented in a chart rather than tabular form. For presenting the data in a better way including touch functionality, the chart library [7] should support zooming and tool tips with bar, line and pie charts. Similarly, data for location visualization application should be embedded with a Maps library which shall provide the distances, travel road guides, time to cover to a location and must visit location near to the business location.

3 Our Approach for Offline Support

There are different approaches to address the above challenges. For example, cloud computing company Salesforce.com, the world's first platform as a service (PaaS) [8], which is best known for its customer relationship management (CRM) product. It provides Salesforce mobile software development kit (SDK), which is an open-source suite of familiar technologies allowing developers to rapidly build HTML5 application that connect to the Salesforce platform. Using this SDK, we can develop a single page mobile application using JavaScript libraries like JQM, Sencha, AngularJS for different

OS platforms like Android, IOS and Windows. Based on this concept, the generalized architecture and framework is given as shown in Fig. 1.

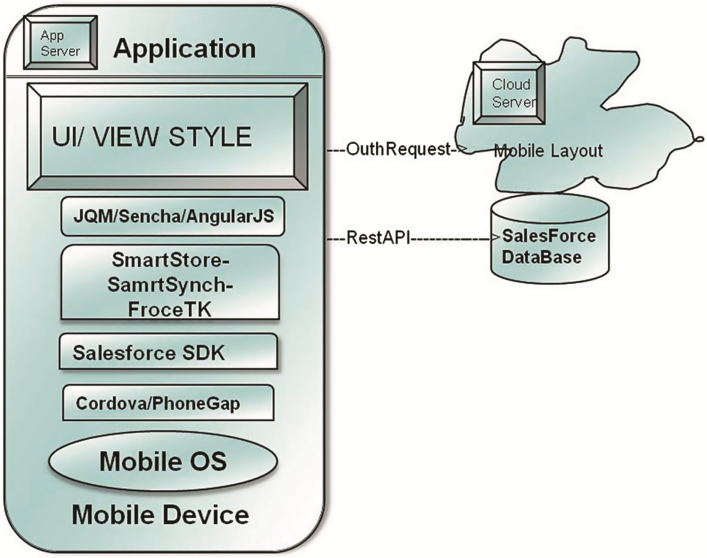

Fig. 1. Development framework

3.1 JQM (jQuery Mobile)

jQuery Mobile framework [9] follows the "write less, do more" approach and take it to the next level. Instead of writing unique applications for each mobile device or OS, the jQuery mobile framework allows us to design a single highly-branded responsive web site or application that will is expected to work on all popular smart phones, tablets, and desktop platforms.

3.2 Sencha ExtJS

Sencha ExtJS [10] is the most comprehensive MVC/MVVM JavaScript framework for building large features cross-platform web applications targeting desktops, tablets, and smart phones.

3.3 AngularJS

AngularJS [11] is a development platform for creating applications using modern web standards. Angular includes many essential features such as mobile gestures, animations, filtering, routing, data binding, security, internationalization, and beautiful UI components. It's extremely modular, lightweight, and easy to understand.

3.4 SalesforceSDK

Salesforce [8] Inc. is a global cloud computing company best known for its customer relationship management (CRM) product. Salesforce Mobile SDK3.0 is an open-source suite of familiar technologies—like a REST API and OAuth 2.0—that we can use to build great mobile apps. The Salesforce Mobile SDK supports three development approaches for building mobile apps: native, HTML5 and hybrid.

Hybrid development combines the best (or worst) of both the native and HTML5 worlds [8]. We define hybrid as a web app, primarily built using HTML5 and JavaScript that is then wrapped inside a thin native container that provides access to native platform features. Phone Gap is an example of the most popular container for creating hybrid mobile apps.

Smartstore, is an encrypted NoSQL-style JSON document data store. It is the cross-platform encrypted NoSQL mobile database in the market that works with both hybrid and native development models. "NoSQL" databases are a class of database that unlike their "relational" brethren, are designed to be inherently schema less. They take care of storing data, and let the application worry about how to structure it. Some of the benefits of this style of database are their raw speed, a dynamic schema where fields and tables can be added by the application at will, and ease of use for the developer. JavaScript Object Notation (JSON) is a lightweight industry-standard way to encode data for transfer between systems. Some NoSQL databases are designed to be simple key/value stores, but more advanced systems like Smartstore allow for storing and indexing full JSON documents. The Salesforce mobile services provides the Smartstore tools needed to build enterprise mobile apps that allow us to securely transfer and store data on our mobile device for highly performant offline access [6].

SmartSync. The SmartSync library is a collection of APIs that make it easy for the developers to sync data between Salesforce databases and their mobile apps. It provides the means for getting and posting data to a server endpoint, caching data on a device, and reading cached data. For sync operations, SmartSync predefines cache policies for fine-tuning interactions between cached data and server data in offline and online scenarios. A set of SmartSync convenience methods automate common network activities, such as fetching sObject metadata, fetching a list of most recently used objects, and building SOQL and SOSL queries.

ForceTk. JavaScript is a popular programming language for building Web-based applications. The most common use of JavaScript is client-side in a Web browser, for implementing pages with enhanced, responsive user interfaces. Client-side JavaScript is also useful for calling XMLHttpRequest methods (GET, POST, etc.) to work with data managed by the remote Web server. ForceTK provides a convenient, thin JavaScript abstraction of the Force.com REST API, making the API more accessible to JavaScript code running in Visualforce, in hybrid mobile apps, and elsewhere.

3.5 Cordova/PhoneGap

Apache Cordova is an open-source mobile development framework. It allows us to use standard web technologies such as HTML5, CSS3, and JavaScript for cross-platform development, avoiding each mobile platforms' native development language. Applications execute within wrappers targeted to each platform, and rely on standards-compliant API bindings to access each device's sensors, data, and network status.

4 Best Practices

Though there are many different approaches, one can follow to develop the apps for efficient offline support. Above Sect. 3, we have used the concept from sales force and develop a mobile application using Hybrid approach. However, some of the best practices in general would be considered.

4.1 Less is More

Significantly important to be selective about what the users really should view in a mobile app. For instance, instead of displaying all the data, display the selective of the information that needs to be displayed and is useful.

4.2 Limit Data Usage

A cloud based mobile application generally displays data that it retrieves from the cloud. This could mean a lot of Ajax calls; which ultimately will be heavy on the data usage. But, since we have the availability of a secured offline storage, we can store the data and display the stored data when synchronization undertakes.

4.3 Limit Stored Data

It is recommended to limit the data storage to avoid over utilization of device memory. If excessive amounts of data needs to be stored then, a limit for the number of items or records received from the cloud must be maintained and stored in the secure offline storage. Additionally, we can also query records with only the required fields instead of all the fields. This would not only optimize memory consumption but will also help in all performance of the application and reduce data usage.

4.4 Make Sure Long Strings Don't Break the Layout

Naturally, we are discussing about the data itself, but the surrounding UI might also suffer. For example, consider a command bar with buttons. When the text on the buttons is translated, they might grow large and overflow the bar.

4.5 Make the UI "Lazy Load"

Make the UI lazy load some of the info and consider using pagination. For example, it might take too long to load at once thousands of items - make sure the UI doesn't look frozen or broken while it loads, or that you load one page and then load the rest while the user can also start interacting with the UI.

5 Conclusion

The cloud is one of the more significant shifts that computing has gone through. As we move towards the cloud, we will discover a new service-based world, and being the developers, we will have to serve the collaboration and security needs of all customers – from those working in small team to those working in huge enterprises. With the knowledge and features discussed in this paper, we can deliver flexible and powerful applications that can be managed on the cloud and will provide the customer with both offline and real-time updates.

References

1. Kendrick, J.: Mobile technology: the amazing impact on our lives. ZDNet Mobile Technology, 30 April 2013
2. Armano, D.: The future isn't about moible: its about mobiliety. Harvard Bus. Rev., 18 July, 2012
3. Guntur, S.: Cloud mobility: the next big disruptive technology. HCL Technology, 18 July 2013
4. Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., Ghalsasi, A.: Cloud computing: business perspective. Sci. Directo Decis. Support Syst. **51**(1), 176–189 (2011)
5. Kim, K.-H., Lee, S.-J., Congdon, P.: On cloud-centric network architecture for multi-dimensional mobility. In: Proceedings of the SIGCOM, 17 August 2012
6. Kadlec, T.: Implementing Responsive Design: Building Sites for an Anywhere, Everywhere Web, 1st edn. New Riders, Berkeley (2012)
7. Patra, S.K.: Data visualization for hybrid application: the challenges in choosing an optimal library for line chart. Int. J. Emerg. Sci. Eng. (IJESE) **2**(10), 12–15 (2014)
8. http://www.salesforce.com/in/. Accessed 12 November 2015
9. https://jquerymobile.com/
10. https://www.sencha.com/products/extjs/
11. https://angularjs.org/