ISSN (e): 2250 – 3005 || *Volume, 05* || *Issue, 06* || *June – 2015* ||
International Journal of Computational Engineering Research (IJCER)

Offline and Online Bank Data Synchronization System

Anil Jaiswal[1], Rahul Sonawane[1], Mrs. Sangeeta Oswal[2], Mrs. Geocy Shejy[2].
[1]*Post-Graduate Student in the Department of MCA, VESIT Mumbai, India.*
[2]*Professor in the Department of MCA, VESIT Mumbai, India.*

Abstract:
Mobile applications mostly being developed are either for a local or client-server. Though the applications in the future will be developed with cloud in mind, i.e. it will be a native application, the heavy processing and storage is done on the native and later uploaded to cloud, it will deliver only required parts and requested data at runtime and able to run offline. In order to better understand how to facilitate the building of mobile cloud-based applications, we have implemented existing work in mobile computing through the prism of cloud computing principles. We provide an overview of the system, in particular, models of mobile cloud applications. Also, synchronization of offline data with online database will be done.

Keywords:*cloud computing, mobile device, remote execution, database synchronization, offloading.*

I. INTRODUCTION

Offline and Online Bank Data Synchronization System is used by credit societies to save the money of people with small businesses, or e.g. shopkeepers, vegetable vendors. These people deposit money at the credit society. The cash collector is the person who is working under the credit society. The cash collector collects the money from these people.Cash collector enters the transaction data on his tablet. We are storing our whole data on local storage at first and it will upload the data to Microsoft's cloud, Azure, as soon as the system gets online. Azure services through a global network of Microsoft-managed datacenters [2].

II. LITERATURE SURVEY

2.1. OData Service:

Open Data Service (OData) is a RESTful data access protocol initially defined by Microsoft. Versions 1.0, 2.0, and 3.0 are released under the Promise. It is the data API for Microsoft Azure. SAP NetWeaver Gateway provides OData access to SAP Business Suite and SAP Business Warehouse. IBM WebSphere extreme Scale REST data service can be accessed by any HTTP client using OData. OData client implementations include Microsoft SharePoint 2010, WCF Data Services and Windward Reports. OData is built on the AtomPub protocol and XML where the Atom structure is the envelope that contains the data returned from each OData request. An OData request uses the REST model for all requests. Each REST command is a POST, GET, PUT, PATCH, or DELETE HTTP request (mapping to CRUD) where the specifics of the command are in the URL.

- GET: Get a collection of entities (as a feed document) or a single entity (as an entry document).
- POST: Create a new entity from an entry document.
- PUT: Update an existing entity with an entry document.
- PATCH: Update an existing entity with a partial entry document.
- DELETE: Remove an entity.

Any platform that provides support for HTTP and XML is enough to form HTTP requests to interact with AtomPub. The OData specification defines how AtomPub is used to standardize a typed, resource-oriented CRUD interface for manipulating data sources.

2.2. Cloud Technology:

Cloud computing is a general term for anything that involves delivering hosted services over the Internet. These services are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). The name cloud was inspired by the symbol that's often used to represent the Internet in flowcharts and diagrams.Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network. At the foundation of cloud computing is the broader concept of converged infrastructure and shared services. Cloud computing, or "the cloud", focuses on maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand.

***2.3. Traditional approach v/s current approach*:**

In traditional approach, the paper receipt of the transaction was given by the agent and at the end of the day he used to submit the details of the number of transactions carried out that day as well as the money collected from the customers manually to the bank manager.Now days, each agent is allocated with the TRIMAX machine. Agent collects the money from the customers and enters the transactions into the machine. At the end of the day, machine database is synchronized with the bank database. In this way the data gets submitted in the bank.With our system in place, the transaction data locally stored in tablet will be getting uploaded on cloud to report to the bank manager. Hence, the data in the system will be directly compared with the data collected by the Cash collector. So if there is any difference in the data or the cash collected then the Cash collector is answerable for it.For traditional system, the procedure of daily collection is totally dependent upon the cash collector's shoulders. Cash collectors helps in collecting money from the customers and also maintaining the respective account balance sheet, so in most cases it is seen that, there is a mismatch between the bank ledgers and the cash collector's ledgers. This system is vulnerable to vandalism and fraud's. Co-operative society as well as vendor has to believe in cash collector in any condition. So, it is very much easy for cash collector to do fraudulent work and misuse of money. Also cash collectors give customers thermally printed receipt, whose ink gets easily\ rubs off if you rub it quickly, or it will get faded in 2 to 3 days, thus the customers won't have any proof of his investment. So, it is one of the main reasons why co-operative society's or co-operative bank goes down in.

III. BANK APPLICATION ARCHITECTURE:

***3.1. Bank application architecture overview**:*

Bank Application is combination of offline mobile application and online web Application. Mobile Application is used by the Agent to enter the transaction data. That data will be at first locally stored and whenever the system gets online it transfers the data over to cloud storage. Online web application is used by the bank manager, he will be able access to data on cloud storage to check the details of the agent transactions and other information.

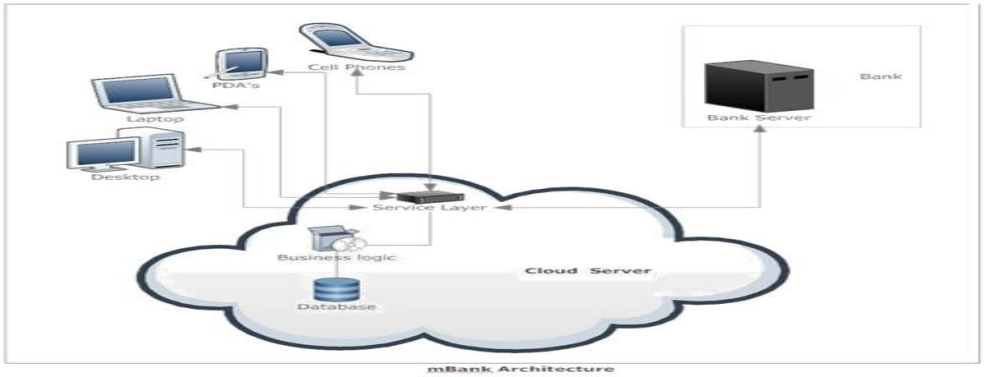

Fig 1 *Bank Application Architecture*

So it Consist of 3 layer architecture

1. Client Side
2. Cloud Server
3. Bank Server

1. Client Side: It can be accessed on any device may that be Computer. Laptop or Mobile with internet connectivity i.e. it is platform independent. There are two types of client one is agent and other one is bank manager.

2. Cloud Server: It is basically the intermediate communication medium between Client and Bank Server. Whatever data entered by the agents through their tablets is synchronised with the cloud server.as well as it will also contain the business logic for this sake.

3. Bank Server: Bank server is taking all the data from cloud storage and updating its data on cloud.

To get the full advantage of potential of cloud computing we need to consider the capabilities and constraints of existing architectures. Most of the applications available for modern mobile devices fall into the following two categories.

3.2. Offline Application.

They act as fat client that processes the presentation and business logic layer locally on mobile devices with data downloaded from backend systems. Only at a certain period, the client and backend system are synchronized. A fat client is a networked application with most resources available locally, rather than distributed over a network as is the case with a thin client. Offline applications, also often called native applications, offer:

- good integration with device functionality and access to its features
- performance optimized for specific hardware and multitasking
- always available capabilities, even without network connectivity

On the other hand, the native applications have many disadvantages:

- no portability to other platforms complex code
- increased time to market

3.3. Online Application.

An online application assumes that the connection between mobile devices and backend systems is available most of the time. Smartphones are popular due to the power and utility of their applications, but there are problems such as cross-platform issues. Here web technologies can overcome them, i.e. applications based on web technology are
powerful alternative to native applications. Online mobile applications have the potential to overcome some of the disadvantages of offline applications because they are:

- multi-platform
- directly accessible from anywhere
- knowledge of Web technologies is widespread among
- developers, greatly minimizing the learning curve required

3.4. Issues with Offline and Online Mobile Applications

Current applications are statically partitioned, i.e. most of the execution and application logic happens either on the device or on backend systems. However, mobile clients could face wide variations and rapid changes in network conditions and local resource availability when accessing remote data and services. As a result, one partitioning model does not satisfy all application types and devices. In order to enable applications and systems to continue to operate in such dynamic environments, mobile cloud applications must react with dynamic adjustment of the computing functionality between the mobile device and cloud depending on circumstances. In other words, the computation of clients and cloud has to be adaptive in response to the changes in mobile environments [1].

So combining the both the systems and taking their pros and neglecting cons this bank architecture is being built.

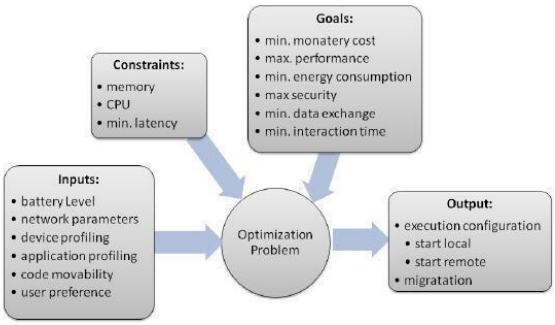

Fig 2 *Optimization issues*

IV. BENEFITS OF THE SYSTEM

1. This entire system will have web based background for maintaining depositional transactional history.
2. Human efforts will reduce in maintaining the details and entering the daily collections into the system.
3. The system provides security to the customer by providing login id and password.
4. Thus system is fraudulent and trustworthy. Also make the system paperless.

V. RESULT

The wire frame of the system is given below:

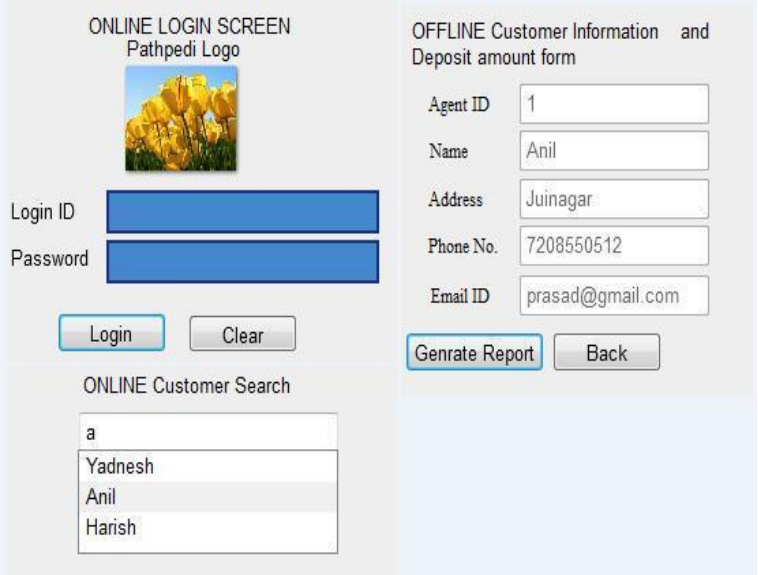

Fig 3 *Wire Frame of the system.*

VI. CONCLUSION

In this paper, we have covered several representative mobile cloud approaches. Other related work exists, but the purpose of this paper is to give an overview of the wide spectrum of mobile cloud computing possibilities. None of the existing approaches meets completely the requirements of mobile clouds. Native (offline) and web-based (online) applications are the two extremes of mobile applications. We believe that the full potential of mobile cloud applications lies in between these two extremes, while dynamically shifting the responsibilities between mobile device and cloud. Several approaches have shown how to achieve it, e.g., by replicating whole device software image or offloading parts of the application. The offloading can happen to some remote data centre, nearby computer or cluster of computers, or even to nearby mobile devices. Moreover, due to the unstable mobile environments, many factors need to be incorporated in a cost model, and fast predictive optimizing algorithms decide upon the best application execution. To simplify the development a convenient, but effective, programming abstraction is required.

REFERENCES

[1] D. Kristol, HTTP State Management Mechanism, February 1997.
[2] 2012 Sixth International Conference on Innovative Mobile and Internet Ser-vices in "Ubiquitous Computing".
[3] Android Wireless Application Development: Volume IAndroid Essentials, By Lauren Darcey, Shane Conder.
[4] http://azure.microsoft.com/en-us/