International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

An Exploration of Web Services and Virtualization Mechanization Fostered In Communication as a Service

Tatenda T. Gotora[1], Kudakwashe Zvarevashe[2]

[1]M Tech Scholar, Department of Software Engineering, Jawaharlal Nehru Technological University, Kukatpally, Hyderabad, Telengana 500085, India

[2]M. Tech Scholar, Department of CSE, Jawaharlal Nehru Technological University, Kukatpally, Hyderabad, Telengana 500085, India

Abstract: ***Basic principle of economics, reiterates the increase in demand of a product with low supply increases price. Likewise the growing demand of mobile clients is not being met by the communications industry, hence Operational Expenses (OPEX) is high. Web 2.0 brought about significant flexibility in the web services interoperability and standardized communication protocols. The open source nature of most of the web service APIs used over the Web has made it easier to build simple to sophisticated Mobile applications for Business-to-Business (B2B), Machine-to-Machine (M2M) and an integration of the both. More so virtualization has not only spearheaded cloud computing paradigms but also facilitated novel virtual technologies especially in Communication. This paper seeks to explore various Web Services (WSs) and Virtualization Technologies used to achieve Communication as a Service (CaaS). Also a proposed hybrid solution to be adopted in the Cloud has been described briefly.***

Keywords: OPEX**,** Mobile applications, Virtualization, Web Services, Communication as a Service

1. Introduction

A web service is a service offered on the Internet for a particular task and can provide base or supplementary functionality. Web Services have for a long time been an integration communication methodology and a force to reckon with in transferring information over the Web. During the Web 1.0 era the most dominant technologies was static HTML pages which provided less user interaction and restricted clients to being content consumers. The advent of Web 2.0 brought XML, dynamic HTML, AJAX and later other server or client side scripting languages. The Web becomes a platform as suggested by most researchers like O'Reilly (2005). This encouraged more user interaction, social networking, blogging and the duality nature of users acting as both consumers and producers of content. Data exchange was achieved using XML HttpRequest (XHR) and SOAP between server and client [1]. Innovations by various third party software experts has increased the availability of open APIs which can be used to link or consume the Web Services and Mash ups over the net. The modern day data exchange is carried using JSON and Rest APIs which not only uses traditional Http methods (Get, Post, Delete and more) but also will be having Representation state.

Virtualization is the ability of a single physical platform being able to host and provide multiple Operating Systems functionalities and resources to the user. Also can be considered [2] as the separation of request or resource for service from the underlying physical delivery of that service. The thrust is to manage a distributed, virtualized environment and to ensure that the resources (physical and virtual) stored in the virtualized database are provided to suit user needs.

This paper will investigate the distinct researches which have been carried out on web services and virtualization technologies which have been adopted for CaaS. Beginning with the service oriented communication framework achieved through Web Services Session Initiation Protocol (WIP) in section 2. Subsequently look into WIPdroid, a mobile distributed communication platform based on WIP in section 3. RESTful Web Services integration in Android is explained in Section 4, followed by Session Initiation Protocol in Section 5. Smart phone Virtualization via Cloud is explored in part 6 and part 7 introduces the concept of Phone Call Virtualization. A proposed hybrid framework comes into play in section 8 and the conclusion wraps it up in part 9.

2. Web Services Session Initiation Protocol (WIP)

Web Services platform independence nature have been the key to the drastic change in the communication paradigms. They have enabled integration and interoperability of various technologies and platforms to be accessed anywhere and on demand over the Web. Web services are shifting from mere service integration methodologies to communication frameworks, as shown by WIP [3] (Web Service Initiation Protocol). It is a web services based service-oriented communication paradigm for real-time communication and unified communication services over IP as shown in Figure 1.

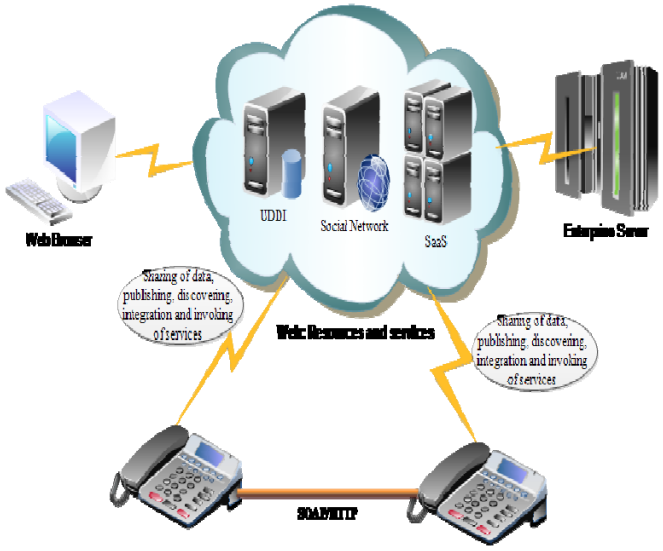

Figure 1: Service Oriented Communication via IP

Communication in WIP is entirely service-oriented as establishment is achieved over IP using a single-stack of web services. WIP endpoints set up calls by exchanging SOAP messages and allow seamless exchange of services between end points. New applications can be added in WIP as services in a declarative and interoperable manner promoting possibility of communication to be offered as a service. The communication follows a Peer-to-Peer presence protocol which makes use of Presentity and Watcher model. The Presentity being the entity which avails its information and resources. The Watcher sends requests and uses the Presentity resources. It eliminates the need for a mandatory Web Server like in Session Initiation Protocol (SIP) and SIMPLE VoIP applications. The messaging security is achieved by using WS-S (Web Service-Security) which employs PKI Standards. The sender and receiver sends or accesses the Private Key and Public Key through the WSSE-I (Web Service Security Interceptor) an independent Party. The WIP URL end points are made obtainable on a search engine like Google via Restful API web services and also offer Web interface for each endpoint on a UDDI Registry as shown on Figure 1.

Limitations of WIP:

- SOAP cannot be used along to offer services but needs the UDDI interface or other web services like RESTful to be made searchable as a URI.
- Complex to integrate many services at once but uses a Peer to Peer Protocol to provide services between end points.

Advantages of WIP:

- Uses WS-Session to keep track of a client usage and uses PKI standards in WSSE-I which makes it hard to intercept the SOAP message.
- Uses Peer to peer communication which eliminates the need for eliminates the need for a mandatory Web Server like in Session Initiation Protocol (SIP) and SIMPLE VoIP applications.

3. WIPdroid

WIPdroid [4] is a dual mode mobile technology which avails a WIP End point communication via IP and another on the mobile environment, that is, Android. It seeks to provide real-time communication on a mobile platform which is web service based while incorporating a distributed computing nature. A WIP Switch is used in WIPdroid architecture and integrated as a supplementary analogous process with Android through XML over TCP/IP connection. This was due to the incompleteness issues in the earlier version of Android and enabled WIPdroid to be developed in parallel with the evolving Android. The WIP based VoIP communication in WIPdroid provides a common real-time communication mode for all WIPdroid endpoints. Physical phones implantation through e.g. GSM, CDMA or SIP phones will be bond to an end point so that it can exist in a mobile environment. This makes WIPdroid handset/PDA a dual-mode phone. A phone type and mobile environment dependent one and one from WIP which is independent of the physical phone to WIPdroid binding, as it only needs a data channel connection to the Internet.

3.1 WIPdroid Architecture

The WIPdroid [4] mobile platform is illustrated in Fig 3 and shows the WIP switch representing the WIP end point. The switch will manage all web services and traffic flow to and from the Android using SOAP/ XML and HTTP. The red blocks represent Android modules which are supposed to provide a GUI to enable user to have full control of WIP end point.

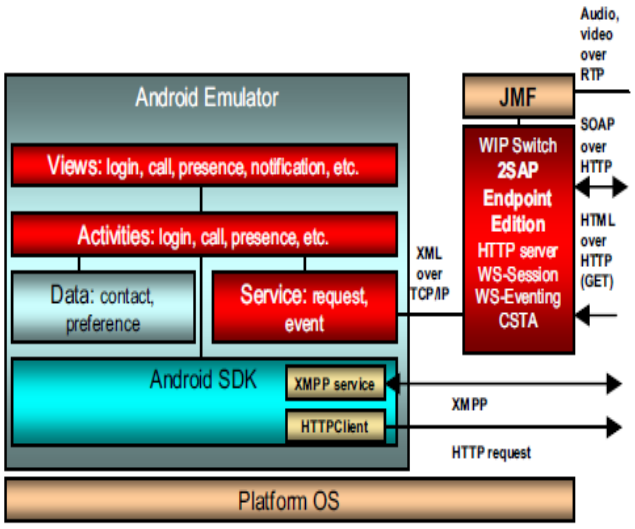

Figure 2: WIPdroid Architecture

Two-way web services application proxy (2SAP) [4] is a web services SOA enablement middleware, meant to support unconventional web services capabilities dire to distributed computing and service-oriented communication. These include two-way full duplex web services interactions, stateful transactions, asynchronous response, and event notification. 2SAP is based on a meta-services/base services design style for web services management and services invocation. Base services are particular to the domain and

meta- services are supplementary like Session handling and require base services to provide full functional services. The Empirical implementation used 2SAP Endpoint Edition which is low memory footprint and the Android APIs were used to provide GUI using Views and Endpoint commands were linked to Activity modules. The Android application was launched on the Android SDK and functions like calling and sending notifications to other WIP phones could be carried out.

Limitations of WIPdroid:

- Message overload and programming complication just like all other SOA implementations.
- Requires a small footprint of 2SAP Endpoint Edition to cater for the Android platform.
- It is dependent on SOAP and HTML which is difficult to offer integrated services over IP.

Advantages of WIPdroid:

- Offers a mobile platform for real-time and distributed computing.
- Uses the traditional XML and HTTP protocols to exchange SOAP messages.
- Services can be invoked from client side using a rich Android GUI and end point can be easily controlled from there.
- Services are discoverable via URI HTTP Request.

4. Restful API chemistry with Android

Representational State Transfer (REST) [5] is an architecture which came into being in 2000 and huge received user confidence. This was due to the simplicity and lightweight nature is exhibited. First proposed by Roy Fielding with the hope of overcoming Service oriented Architecture (SOA) limitations. The web services which implement this framework are called RESTful web services and they follow the following ideologies:

- Identification of Resources: Everything is defined and referred as resources [6]. A resource is a software artifact supporting specific data, provided by the resources. Each resource has a unique URI in HTTP which makes it accessible worldwide if deployed on a server. For example: http://www.mysite.com/country represents a list of URIs for Country Data. To get “India” details in countries we make Http Post to http://www.mysite.com/country/india
- Manipulation of Resources through Representations: HTTP protocol is used to implement REST web services. This allows the same interface to be used to design every resource by a fixed set of HTTP traditional methods (GET, POST, DELETE etc.).
- Self-Descriptive Messages: Resources are self-descriptive in nature, implying a request will possess all sufficient knowledge on how to do a task and give a response in multiple representation such HTML, XML, JSON.
- HATEOAS: Hypermedia as the Engine of Application State (HATEOAS) is a vital notion in REST. Each response message needs to have a hyperlink to navigate to the next request message so that neither client nor server needs to store information about the state. This permits notification of altered state to client application at the server via hypermedia.

The three elements such as URI, HTTP methods and Media types (XML or JSON) are uniform contract design elements for designing REST. The RESTful architecture integrates multiple Web Services APIs and generates mashups. A composition algorithm was applied to integrate a set of RESTful Web services, which helped a farmer to select a crop based on location. The Android mobile device was the target platform and provided GUI to interact with Services offered by server. Fig 3 shows the framework implemented to develop the RESTful Web Service integration with Android. The farmer/client communicated via the mobile device sending URI POST and IMEI information for identification of user to GIS. A farmer may require to find out information about the crop type associated with a given geographical location. Also go on further to want to know the market at their disposal.

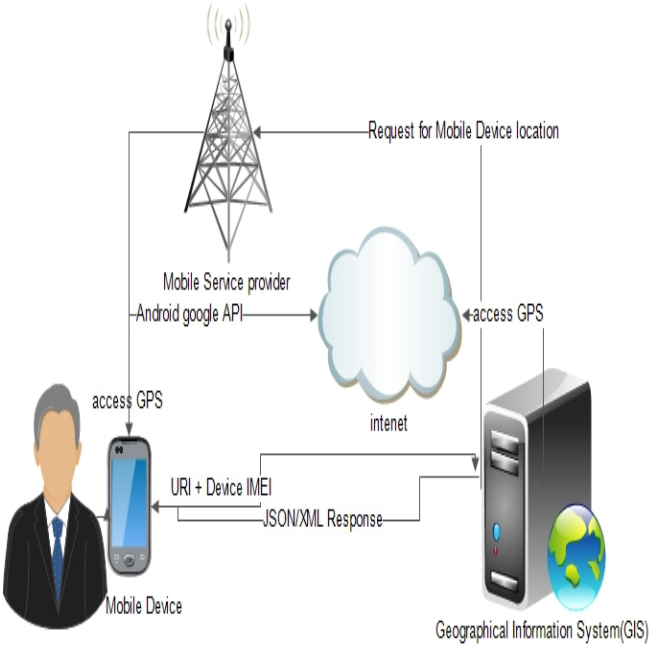

Figure 3: Restful Web Services integration with Android

Advantages of using RESTful Web Services over SOAP

- Reliable Availability: Can be searched for or accessed via a simple URI by clients unlike SOAP requires WSDL interface definition and interpretation to be accessed on the Web.
- Expandability: This is due to the ability to naturally support caching and parallelization/partitioning on URIs. Also supports a more effective and simpler way to provide load balancing based on URI partitioning. While SOAP interfaces uses adhoc partitioning. This makes RESTful WSs [7] the best option to take in building asynchronous distributed computing Web application which can be adopted for any platform.
- Mash Ups: RESTful WSs have the capability to integrate to build simple to complex Web applications and most companies like Amazon [6] use more of REST than SOAP

on their Sites. RESTful WSs users have grown to 85% over the years due to the easier access of a pool of user required Mash Ups.

5. Session Initiation Protocol (SIP) for Real Time Communication

The Next Generation Networks (NGN) proposed during the early 2000s sought to [9] [10] bring about a fusion of Telecommunications and IT services. It required new innovations which allowed a packet-based network system which could transmit all types of data while keeping a clear separation between services layer and the underlying physical network layers. SIP emerged as a solution which could offer change. SIP [9] is a signaling, messaging protocol developed by the Internet Engineering Task Force (IETF). It is capable of setting up, modifying and tearing down multimedia sessions, requesting and sending presence and instant messages. Alike Hypertext Transfer Protocol (HTTP) SIP works in the application layer of the OSI communications model.

The architecture of SIP defines and uses few components:

- User Agent Client (UAC)
- User Agent Server (UAS)
- User Agent (UA)
- Proxy server
- Redirect Server
- Location Server

Although SIP [8] is easy to use it is neither XML nor a Web Services and thus requires special SIP operations to provide services on a real-time basis and store communication session. A real-time web service communication model based on the SIP was proposed and implemented as shown on Fig 4. It consists of 8 major modules which are described as follows:

Web Service Adapter Module, [8] is responsible for encoding and decoding the SOAP messages. Conversion and formatting of SOAP message to and from the clients.

Message Dispatching Module is responsible to provide an interface for forwarding unprocessed messages to corresponding internal interface while coming various modules.

Call Messages Process Module is responsible for the call control state maintenance. It comprises of two sub categories; one state machine management and call state machines. The former creates and destroys state machines and both forward the processing request messages from the business logic and the SIP stack adapter module. The state machine sub provides maintenance to the call control state and reports the call event in the real call set up.

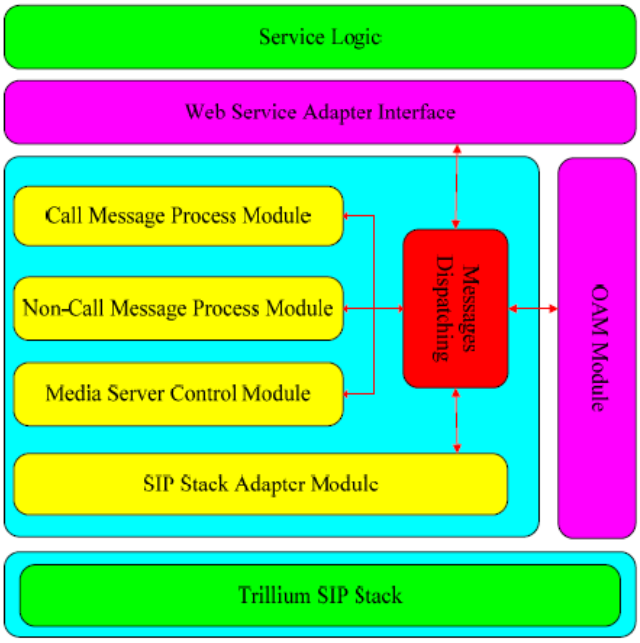

Figure 4: Real Time Web Service Communication Model based on SIP

Non-Call Message Process Module handles all non-call related messages and network messages, like service registration.
Media Server Control Module, handles all the media related request messages to media server. MSML (Media Server Markup Language) allows services to be dynamically applied and changed via control agent during the lifetime of the SIP session.

OAM Module forwards the messages to the OAM process module and to complete the messages statistic. These messages contain configuration messages, heartbeat message and log message. It is not a purely message process module and the instance which provided the OAM module can be obtained via messages dispatching module.

SIP Stack Adapter Module serves as the interface for sending and receiving SIP message to /from the message dispatching module to SIP Stack module while providing the necessary translations.

Trillium SIP Stack is a complete implementation of the SIP protocol stack for use in a SIP enabled device and in NGN. It provides its own reliability and also TCP/UDP enables SIP messages to be multicast.

Advantages of using SIP
- Easy to implement in the Application Layer of TCP/UDP.
- Is a loosely coupled protocol.
- Can be identified as a URI in HTTP.

Disadvantages of SIP
- It is not scalable like RESTful WSs.
- Requires a special Framework to be adopted in Real Time or Distributed Computing.

6. Smartphone Virtualization

Mobile devices have for long been viewed as mere dumb calling and messaging terminals. Thanks to the ferocious development in Smart philosophy in mobiles, Web technologies and telecommunication Infrastructure it has led to the consideration of virtualizing the constraint System on Chip (SOC) [12] architecture. The smartphone has a limited memory, processing power and power. Most researches in the Internet of Things is majorly focusing expanding the wireless and connectivity of the new WebTops [13] to the Web at anywhere and anytime. Smartphone virtualization is a concept meant to avail another virtual platform to perform tasks on an on demand mode. According to Intel Research Berkeley [11] the smartphone can host its expensive, exotic applications on an execution engine. Thereby augmenting the smartphone's capabilities by seamlessly off-loading some tasks to a nearby computer, where they are executed in a cloned whole-system image of the device, reintegrating the results in the smartphone's execution upon completion. This helps improve the processing speed and introduces restore points for backup.

6.1 Types of Augmentation

Primary functionality outsourcing: [11] Computation hungry applications such as speech processing, video indexing, and super-resolution are automatically split, with GUI and other less processing activities being retained on the phone. The high-power, expensive computation is off-loaded to the infrastructure, synchronously. This is similar to client- server architecture with server offering services.

- Background augmentation: [11] deals with operations that do not require to interact with users in a short time scale, such as scanning the file system for viruses and crawling news web pages.
- Mainline augmentation: resides between primary functionality outsourcing and background augmentation. Here the user may opt to run a particular application in a wrapped fashion, altering the method of its execution but not its semantics. For example fault tolerance.
- Hardware augmentation: this seeks to compensate for fundamental weaknesses of the smartphone platform, such as memory caps or other constraints, and hardware peculiarities.
- Augmentation through multiplicity: is unique because it has multiple copies of the system image executed in different ways. This will promote parallel processing of applications.
- These various augmentation categories implemented on a cloud (laptop, desktop, server nodes) on a clone (virtual phone, VP) and upon completion the results are combined and sent back to phone via Wi fi as shown in fig 5.

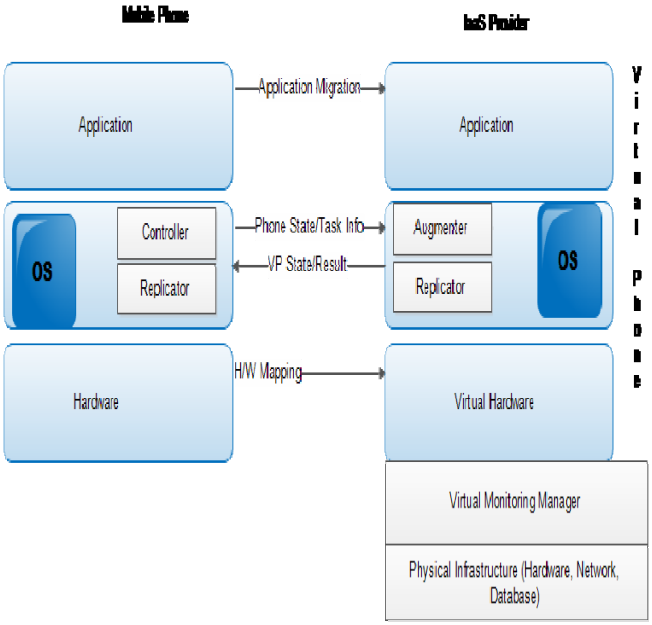

Figure 5: Smartphone virtualization Architecture

The Replicator synchronizes the changes or state updates in phone software and clone (Virtual Phone). The Controller is responsible to invoke augmentation execution and integrating the results back to the smartphone, while liaising with replicator for state update. The Augmenter running on VP takes care of local processes, and returns a result to the phone.

Advantages of Smartphone Virtualization [11], [14]

- Increased processing power.
- Extends battery life.
- On demand facility offered.

This sets motion for virtualization of components found in a mobile device like the SIM card and Memory Card as well.

7. Phone Call Virtualization

Raising demand in mobile users and various requirements has forced both the Academic and Industry professionals to deliberate outside the box .One might not find all user requirements needed in one particular Telecommunication service provider for which one would be strained to carry more than one mobile or multi- SIM usage. Over the years big companies like Samsung, Huawei and Micromax mobile device manufacturers have opted for Dual SIM architecture and it serves the purpose. Although at times the two SIM network coverage does interfere, employing a well – woven virtualization technology called [2] phone call virtualization technology could be a better and cheaper option. Along with this, a supportive cloud service named Phone call as a Service is employed (PCaaS).

The PCaaS [2] uses a cloud to store the SIMs in a centralized or distributed database and at the same time provide an all one-stop platform. The cloud provides various Communication Service Providers (CSPs) an opportunity to offer their services to customers on a dynamic loading and

unloading on to their devices on demand. A client can either obtain the SIMs synchronously or asynchronously .Each client sends a request and gets a unique Session ID allocated when connection is accepted. If it's a direct request the connection gets terminated as soon as response is obtained while an indirect one a notification or redirect commands are also availed to the client. For instance, a message notification or call is loaded to the client's other SIM on the cloud comes while a call is in progress on the active one the alert manager will be raised. PCaaS can be incorporated can easily into the current underlying communication infrastructure using a powerful API but it requires altering of the Client Phone OS to include the virtual SIM services. This raises tight coupling to a particular mobile platform and most likely there will be SIM loading delays for communication to be established.

Limitations of PCaaS:

- Supports a finite number of Service Providers unlike other Cloud Computing Services.
- Virtual SIMs can't be accessed simultaneously but relies on the notification algorithm and alert mechanism employed at the Cloud.
- Does not fully support real time and distributed computing as it requires enhanced communication services.
- Benefits of using PCaaS:[2]
- PCaaS integrates social networking, cloud computing, and phones together.
- Provides the choice of network providers thus reducing the capital expenditure
- Reliable and resilient services are provided by multiple secure sim hoards which are loaded in to the cloud which are highly scalable.
- Ease upgrading process which is availed in the form of new services.
- Client phones use an efficient notification system.
- Session ID is used to identify and provide authentication to a user.

8. Proposed Hybrid Web Service Based Communication Model for Telephony Services

Each Web Service Communication Model based on the SOAP seems to have a glitch in achieving asynchronous real time communication. At the same time RESTful web services make use of Open APIs to build dynamic and complex web applications but require a trusted Server to contain your URI resources. The security of Open APIs is still flaccid and can be easily comprised. Fusing both the good aspects of SOAP and REST Web Services will be the key to achieving distributed computing communication as a service while incorporating the Virtualization of SIM cards and the networking infrastructure. This will bring abought a flexible and extensible telephony services which are offered on demand and with less OPEX (Operational Expenses) and CAPEX (Capital Expenses) being used by CSPs.

9. Conclusion and Future Works

Technology is advancing likewise communication is following suit, to try and meet the growing user needs. Web services are shifting from mere service integration methodologies to communication frameworks. WIP is a greater example of a SOA Web service based paradigm offered via IP and uses SOAP to transmit messages between end points. The UDDI registry enables a Web Interface for the end points while making them discoverable or after incorporating then with RESTful Web services. Services can be added in a dynamic manner and this promoted the raise of WIPdroid a WIP architecture being adopted on an android platform to provide real-time communication with a distributed computing nature. A WIP Switch implements a WIP End point which has a 2SAP supporting unconventional web services capabilities and the switch can be controlled by the user from the Android GUI views. Like all SOA applications it has message overloading and programming complexity when unifying SOAP applications. The REST framework tries to eradicate SOAP shortfalls through its simplicity and lightweight nature. RESTful Web Services are well suited for the asynchronous real time and distributed web applications as they are easy to identify, scalable, self-describing URI and use HATEOS. SIPS is just as ease to implement and discovered using URI like RESTful WSs but are not as scalable and require a special framework to be adopted for real time communication. Virtualization is moving to the smart phones and can be applied through augmentation techniques while tasks are loading to a clone phone on a Cloud. This will increase processing power and increase multitasking. Moreover a Phone call can be offered as a service as the advent of virtual SIMs comes into play. This will remove the hustle of carrying to multiple phones or SIMs as one can simply use one mobile device to load multiple SIM on demand while being offered various services. For future works much has to be carried out in the analysis of the integration of Web Services and Android in SIM card virtualization.

References

[1] Balachander Krishnamurthy, Graham Cormode. "Key differences between Web 1.0 and Web 2.0", First Monday, Volume 13 Number 6, 2 June 2008
[2] R Monica1, Dinesha H.A, V.K.Agrawal, Cloud Computing – Phone Call as a Service: A Concept, 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp 236-242
[3] Wu Chou, Li Li, Communication over IP Based on Web Services , 2008 IEEE International Conference on Web Services, pp 409-416
[4] Wu Chou, Li Li, "WIPdroid – A Two-way Web Services and Real-time Communication Enabled ", 2008 IEEE International Conference on Services Computing, pp 205-212
[5] Mohammed Husain Bohara, Madhuresh Mishra, Sanjay Chaudhary, "RESTful Web Service Integration Using Android Platform", 4th ICCCNT 2013, IEEE – 31661
[6] Ye Zhou, Yang Ji, "Design of rest APIs for the exposure of IMS capabilities towards web services", Proceedings of ICCTA2011
[7] Frederik Orellana, Marko Niinimaki, "Distributed Computing with RESTful Web Services", 2012 Seventh International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pp 103-110

[8] Cheng Bo, Guo Jie, Meng Xiangwu, Chen Junliang , "SIP Based Real-Time Web Services Communication Model", 2008 ISECS International Colloquium on Computing, Communication, Control, and Management, pp 439-443
[9] International Telecommunications Union, online http://www.itu.int/ITU-T/studygroups/com13/ngn2004/working_definition.html, (accessed 22 July 2014)
[10] Michal Marzec, Bartosz Sakowicz, Piotr Mazur, Andrzej Napieralski, "Application of the SIP Protocol in Telecommunication, Based on a Image Transmitting Application", TCSET'2010, February 23-27, 2010, Lviv-Slavske, Ukraine, pp 163-165
[11] Byung-Gon Chun, Petros Maniatis ,"Augmented Smartphone Applications Through Clone Cloud Execution", Intel Research Berkeley
[12] Xiaoyi Chen, "Smartphone Virtualization: Status and Challenges"
[13] Tatenda Trust Gotora, Kudakwashe Zvarevashe, Pranav Nandan, "A Survey on the Security Fight against Ransomware and Trojans in Android", International Journal of Innovative Research in Computer and Communication Engineering Vol. 2 Issue 5 May 2014, pp 4115-4123.
[14] Shih-Hao Hung, Chi-Sheng Shih, Jeng-Peng Shieh, Chen-Pang Lee, and Yi-Hsiang Huang, "An Online Migration Environment for Executing Mobile Applications on the Cloud", 2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, pp 20-27

Author Profile

Tatenda Trust Gotora: Attained his BSc degree in Computer Science at MSU, Zimbabwe in 2011. He is currently doing M Tech SE final year at JNTUH, India. He is a HIT staff development research fellow. His research interests are in the area of mobile computing, big data, information security and web services.

Kudakwashe Zvarevashe: Attained his BSc degree in Information Systems at MSU, Zimbabwe in 2010. He is currently doing M Tech IT final year at JNTUH, India. He is a HIT staff development research fellow. His research interests are in the area of big data, Information security, cloud computing and web services.