Secure Cross-Platform Hybrid Mobile Enterprise Voice Agent

David Jaramillo
CIO Lab – Mobile Innovations
IBM
Boca Raton, FL, USA
djaramil@us.ibm.com

Viney Ugave
Department of Electrical and Computer Engineering
Colorado State University
Fort Collins, CO 80523
vinzzz@rams.colostate.edu

Robert Smart
Emerging Technology Services
IBM
Hursley, UK
smartrob@uk.ibm.com

Sudeep Pasricha
Department of Electrical and Computer Engineering
Colorado State University
Fort Collins, CO 80523
sudeep@colostate.edu

Abstract— Voice is a very compelling natural interface for mobile computing devices. However, mobile operating system speech solutions (e.g., Siri, Google Voice Search) are platform-specific, making their integration within an enterprise environment difficult due to the lack of uniformity in the user experience as well as concerns over privacy and confidentiality. Hybrid apps can overcome these enterprise-specific challenges by allowing for fast deployment of functionality that is accessible across mobile platforms and enabling secure containerization of resources. But such hybrid apps often encounter significant performance slowdown due to the overheads associated with cross-platform compatibility and security. In this paper, we introduce a new secure mobile voice agent for enterprise voice interface applications on mobile devices that overcomes these issues. The proposed voice agent architecture enables confidentiality, enterprise integration, good cross-platform performance, and easy portability to multiple mobile platforms (such as Android, iOS, Windows Mobile, and Blackberry). Our approach avoids the need to manage multiple native speech engines across mobile OS platforms, reducing the time to deploy cross-platform voice interface solutions, and ensuring a cohesive user experience. In addition to describing the architecture and design of our mobile voice agent, we also provide a comparative analysis of its performance on various contemporary mobile computing platforms.

Keywords—mobile computing; voice agent; hybrid mobile apps; enterprise computing; Android; IOS

I. INTRODUCTION

The success and failure of enterprises depends on the sophistication, reliability and trustworthiness of their software systems and processes to produce precise and consistent results. Enterprises also stress on optimizing process costs to obtain increased profits. These processes often carry sensitive data and thereforesecurity is one of the major concerns for enterprises. Compromising any sensitive data could have a damaging effect and result in loss of revenue as well as customer confidence. With the advent of mobile computing systems, enterprises face a challenge to extend access to their highly reliable secured systems out on to mobile devices.

Contemporary mobile devices provide a convenient mechanism for people to interact with information and with each other on the go. Today's smartphones have truly revolutionized mobile computing. These smartphones have higher processor speeds and more memory than desktop PCs had ten years ago. Smartphones have changed the way people consume information by allowing for easy capture and sharing of data between users, thereby creating more of a social experience compared to conventional PC applications. Enterprises are also incorporating smartphones because of these benefits and have been relying on mobile device management (MDM) [14] strategies to secure, monitor, manage and support mobile devices. But the availability of diverse mobile platforms and the performance and energy bottlenecks of mobile devices make it arduous to implement enterprise applicationsefficiently on to today's mobile systems.

Cross-platform compatibility in particular is problematic from an enterprise perspective. Every mobile platform has its own software development kit(SDK) and its own programming language. The mobile market, which is largely dominated by Android and iOS devices, uses Java and Objective C programming languages for development, which are very distinctive. Significant programming effort is required to replicate and port appsacross mobile platforms. Moreover every platform has its own API and features. The result is that the time spent and cost of creating platform specific apps is very high. In addition the look and feel from one platform to another is different due to platform specific UI controls. To address all of these issues the only solution is to users frameworks that aid the creation of cross platform apps. Numerous solutions have surfaced in recent years – Sencha Touch [15], jQuery mobile [16], PhoneGap [17], IBM Worklight [18] to name but a few.

Mobile systems also provide natural interfaces like touch and speech to interact with mobile devices;and speech in particular is an alluring natural interface. These multiple interfaces have given birth to multimodal systems, which provide a more flexible experience by offering a greater range of interaction than unimodal systems [9]. Natural interfaces help to bridge the gap between physical and digital worlds. Voice enabled applications are now widely seen on

978-1-4799-6585-4/14/$31.00 ©2014 IEEE

smartphones due to the improvement in speech recognition technology, network and processing power in recent years.

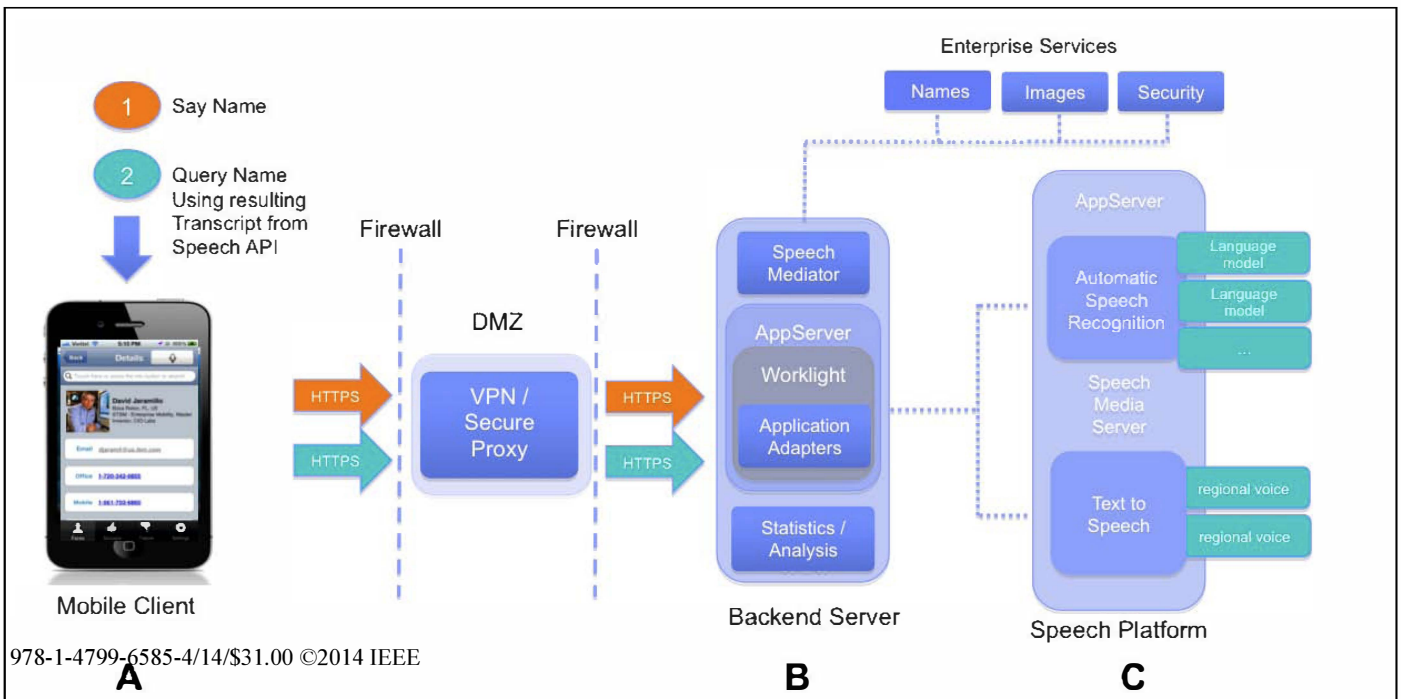
Speech addresses some of the drawbacks of a touch or keyboard interface on a small form factor device, but introduces some serious security and privacy concerns. For an enterprise app sending voice data to third party servers can be a big confidentiality risk. But this does nothave to imply that enterprises should not enable speech interfaces in their applications, considering the enormous usability benefits of integrating natural voice activated interfaces. *All these challenges motivated us to create a highly specialized and secure voice agent using a hybrid app architecture that is cross platform*. Hybrid apps help to overcome cross platform challenges by utilizing HTML, JavaScript and CSS as common development tools across platforms.*We present the enterprise architecture of our voice agent in this paper, along with a comparative study on multiple mobile platforms to determine which components are potential bottlenecks in the performance of the voice agent.*

II. Related Work

There is a great interest in business-to-business (B2B) and business-to-employee (B2E) mobile apps [2] due to the benefits of providing employees with 'anytime, anywhere' access to business relevant data and processes. Studies suggest that most enterprise mobile apps are lightweight and their major function is similar to a query engine that queries the backend enterprise servers [1][2]. This allows for enterprises to keep data secure by storing the data at enterprise servers using standard enterprise security policies. *But if this data was to be stored on mobile devices or third party servers then this creates a big loophole in the security infrastructure setup by companies.* If enterprises were to use existing voice agents like Google Voice Searchor Apple's Siri, they would be highly vulnerableto security threats. Therefore enterprise voice interface solutions need to be customized and cannot rely on commodity solutions from Google and Apple.

Voice recognition systems have been in existence for a long time. Early prototypes were being used for creating assistive systems /guidance apps for visually impaired people [10]. With improving accuracy and adaptability, voice systems are today being extended to various embedded devices, smartphones, etc. [11]. The major obstacles identified in these speech engines are background noise and speaker variation [11]. The word success rate (WSR) in quiet environments has beenshown to be higher than in noisy environments. Techniques such as automatic gain control(AGC) have improved WSR in noisy environments [12]. Modern mobile voice agents like Google Voice Search have also evolved to handle multiple dialects very well[13]. *All these developments suggest that voice recognition systems have become more accurate and overcome many of the challenges of the past; a key challenge now is to support a cross platform voice system that integrates seamlessly with existing enterprise apps.*

In recent years, hybrid apps, which are part web app and part native app, have emerged to provide more access to a mobile platform's full capabilities than web apps, yet they are actually cheaper to develop than native apps [4]. Hybrid apps, for example created with Apache Cordova [5] (formerly known as Phone-Gap), essentially package a Website within a native wrapper that provides access to device features. Hybrid apps that are cross platform by far support the highest number of mobile platforms [6]. In hybrid apps, HTML, CSS and JavaScript are used to build the user interface. This allows for a good enough user experience. In spite of all the great advantages of hybrid apps there are a few shortcomings. Display and interaction speeds are not always equivalent to what the users are accustomed to while using native apps [4][6]. Also, not all advanced device specific features are accessible using hybrid apps. Moreover there are some performance related issues with hybrid apps, for example Facebook went from being a hybrid app to native app [7] mainly due to slow screen refresh amongst other issues. But this change from hybrid to native for Facebook resulted in an increased development cost for the app, in addition the deployment of updates across multiple platforms required more time. Hybrid apps have proven to offer a way of providing secure, cost efficient and agile solutions for enterprises, but one must consider any tradeoff in performance when it comes to adding sophisticated functions. *This paper proposes and implements a framework for a secure enterprise mobile voice agent, which is a hybrid app, and explores performanceissues via comparative studies that highlight components which are time expensiveas well as the variables that can be tuned to improve the speed of our hybrid app.*



978-1-4799-6585-4/14/$31.00 ©2014 IEEE

Figure 1 Voice Agent Architecture

III. PROPOSED FRAMEWORK

Creating a secure cross platform mobile enterprise voice agent, in the form of a hybrid app involves a system architecture that is built usingtried and tested principles of secure enterprise architecture. The architectural overview diagram is shown in figure 1. The basic operation of the voice agent is that the mobile client is responsible for recording the audio from the user and securely sending it to the backend server, whichin turn sends the audio to the speech media server in order to perform automatic speech recognition (ASR) [19] also known as speech to text (S2T). The backend server then takes the resulting transcript and performs a REST query to the appropriate enterprise service to obtain the query result. A further request is made to a text to speech engine (TTS) in order to obtain an audio summary of the query result. Finally the text to speech audio and the application output are sent back to the mobile client. We will now discuss the different parts of the architecture in detail:

A. Mobile Client

The mobile client is a hybrid app built using IBM Worklight. It contains HTML, JavaScript and native code plugins that utilize numerous libraries to create a secure solution with a rich user experience. In order to create a user-friendly interface experience that is close to that of the native UI there are a number of libraries available like jQuery mobile, dojo mobile, etc. We chose to use the Dojo Mobile toolkit [20]. Dojo Mobile has numerouswidgets that allow for the creation of a unified UI across multiple platforms including iOS, Android and Blackberry. Using JavaScript and the Apache Cordova plugin interface we can access device specific native API's to control the device microphone. For voice input we developed two different voice interaction methods, one that asks the user to manually start and stop speech recording by toggling a button and another that uses Voice Activity Detection (VAD) to automatically detect when the user has started and finished speaking. In order to facilitate VAD we use a proprietary native library that detects periods of silence after voice activity and automatically ends voice recording. Recorded audio is compressed using the speex audio compression format, which is an open source codec especially designed for speech. In order to optimize the audio upload time we stream the audio using HTTPS[21] chunked transfer in parallel to the audio recording. The chunked audio transfer is sent to a RESTful [22] service end pointand in response we receive the JSON [23] output back from the server. All the communication to and from the mobile client has to pass through a firewall.

B. Backend Server

The backend server (BES) is exposed to the external network using a secure authenticated proxy server located inside a demilitarized zone (DMZ) that separates the Internet from the company intranet.The BES is responsible for numerous functions and has a number of components. It is essentially a set of Linux servers running CentOS that hosts a cluster of application servers, Worklight servers, and a database server. We use Websphere Application Server (WAS) as our application server. WAS hosts both the Worklight application and a further Speech Mediator application that consists of a set of REST services to handle the user speech queries. The Worklight server provides authentication and authorization services to the mobile client as well as utility services such as logging. We also create a backend application with a web based administration panel totrack and present metrics on usage. This panel also provides the facility to alter various configuration options of the application, allowing us to change parameters such as the speech model used without requiring a client update. The Speech Mediatorapplication was created using the Play framework and interacts with the speech engine to obtain the transcript of a user voice query. The Worklight application has services (*adapters* in Worklight terminology) to log data and forward requests to the Speech Mediator application. The Speech Mediator application also interacts with other enterprise services.

C. Speech Platform

The speech platform provides both anASR service that does S2T and also provides TTS capability. The speech platform is a separately hosted service and is accessible using RESTAPI calls. There are numerous speech recognition services available like Nuance Dragon [24] speech engine, etc. We use a proprietary speech engine for our purpose. The ASR service further offers language models for different languages, dialects, or domains. In particular the voice agent solution uses general language models that have been extended with business and technical data extracted from the targeted enterprise services. This use of domain specific models increases the accuracy of the ASR. Likewise the TTS engine can generate the TTS audio with regional voice for the desired language. The ability to choose different models is exposed through the speech platform REST API along with parameters for audio encoding, result formatting etc.

D. Enterprise Service

Depending on the functional purpose of the mobile app,an enterprise can interface various services using the backend application as a mediator. Common services would be to search for resources on company servers using different phrases, or to translate sentences to different languages, etc. Our particular mobile voice agent instancequeries two pre-existing enterprise applications: one that provides a searchable directory of employees and a second that contains employee sourced question and answer data that has been shared through internal social networking. The application service is architected such that additional services can be added on with minimal development effort. Both enterprise applications expose their services via a published REST API.

E. Security Measures

In most cases mobile apps deployed by an enterprise have to fulfill a set of security criteria before rollout. They also come under scrutiny of security review teams. In the case of this mobile voice agent there are several levels of protection both at the client and server. Firstly, access to the app install is controlled through an enrolment process only accessible to employees. The app is not made available in public app stores but instead can be installed over the air (OTA) via an access controlled web page. Enrolled employees are placed on an access control list (ACL) that allows them to access the app. Once the app is installed and executed,a user must enter their credentials to gain access to the main app functionality. Authentication is carried out via Worklight services against the company LDAP directory [25]. Authorization takes two forms, the first is that the employee user id is on the applications ACL, and the same ACL used to control access tothe install site. The second form of authorization is to validate that the device on which the app has been installed upon is managed by the company mobile device management

978-1-4799-6585-4/14/$31.00 ©2014 IEEE

(MDM) system. This check is performed at the server-side. The device's unique identifier (uid) is passed along with user credentials and verified using a common REST security service that exposes MDM data.In addition to the authentication and authorization processes, data obtained from using the app services is not stored on the device. The data is kept entirely at the server-side where it can be managed under standard security practices. All requests made to the server system are performed over a secure connection verified by certificates.

IV. Experiment and Results

In order to understandthe performance of our voice agent architecture, we performed several experiments that are discussed in this section.

TABLE 1. List of Devices

Device	Device Specifications	
	Platform OS	*CPU/Memory (RAM)**
Samsung S4	Android 4.3	Quad-core 1.6 GHz Cortex-A15 & Quad-core 1.2 GHz Cortex-A7/ 2 GB
IPhone 5	iOS 7.0.4	Dual-core 1.3 GHz Swift /1 GB
IPhone 5s	iOS 7.0.4	Dual-core 1.3 GHz Cyclone (ARM v8-based)/1 GB
Samsung Note 3	Android 4.2	Quad-core 1.9 GHz Cortex-A15 & quad-core 1.3 GHz Cortex-A7 (N9000)/3 GB

*Random Access Memory

A. Experiment Setup

Our experiments involved logging and timing every component of the hybrid voice agent architecture for various inputs. The experimentshad a number of other variables:they werecarried out on two different OS platforms(Android and iOS) and also on different devices. Table I gives the details of the devices used. We used a variety of devices to show how our voice agent performs across OS platforms and mobile devices. We used a database of 60 voice samples that were on an average 1-2 seconds long, as the inputs. Themain metrics measuredbyour experiments are summarized as follows:

1) Total Transaction Time: As seen in figure 2, total transaction time is the time taken to complete one complete request from the point when an audio query is initiated on the mobile device to the time when a response to the audio query is received at the mobile device.

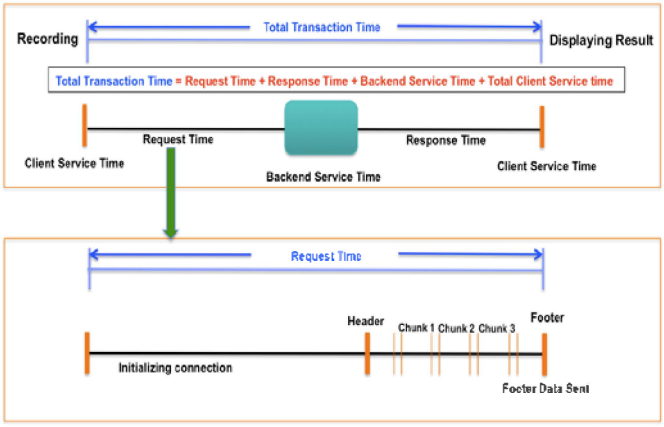


Figure 2. Time Statistics

2) *Client Service Time:*This is the time taken by the mobile client i.e. the native and JavaScript code that is responsible for the audio conditioning; this includes functions such as AGC for noise reduction, recording, compression and creation of the https request.

3) Request Time: This is the time taken to stream the audio to the backend server and complete the https request. This time involves initializing the https request and buffering chunks of data to the backend server as shown in figure 2.

4) Response Time: This is the time needed for the backend server to transmit the query result and TTS audio back to the client.

5) Backend Service Time: This timeincludes a lot of factors. The bulk of the time consists of the time to query the speech engine to get the S2T and TTS output and the time to process the HTTP request to query the enterprise services.

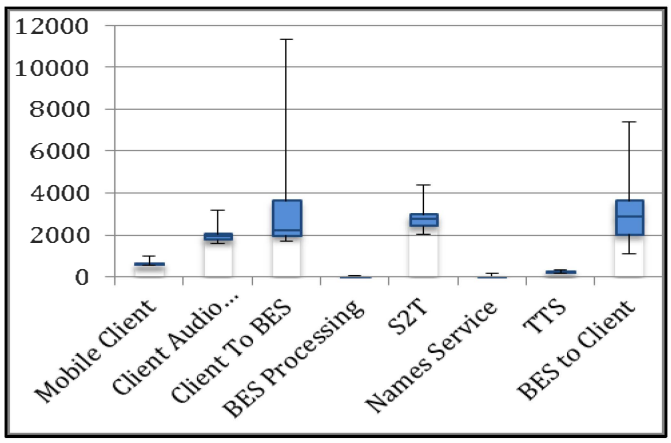


Figure 3. Performance Results on IOS Client

B. Results

Figure 3 shows the performance results of running our voice agent on an iPhone 5 iOS client. The metrics measured in the figure are as discussedabove. The S2T, TTS, Names Service(NS) and BES Processing constitute the Backend Service Time. The Mobile Client and Client Audio Compression constitute the Client Service Time. The Client to BES and BES to Client is the request and response time respectively. All the times are shown in miliseconds (ms).

After examining figure 2 the main observation is thatthe *majority of time is spent in transmission of data from the backend server to the mobile client and vice versa.* The average transmission time is between 2 to 3 seconds. This could be a result of time spent in the secured transmission of data that has to pass through the security layers necessary for enterprise security policies. But the advantage of this phase is that the data is kept secured on the company servers instead of on some third party server. The experiment was conducted on a stable wifi connection, but we still observe a large positive error ,which we believe is due to network conditions affecting the data transmission.

Next we observe that the average time for the hybrid mobile client to complete all its functions (audio recording, conditioning and compression) is between 2 to 2.5 seconds. Out of this time,the voice library that is used to compress the

978-1-4799-6585-4/14/$31.00 ©2014 IEEE

recorded audio consumes around 2 seconds. This clearly suggests that the time taken by the hybrid client is relatively small (~0.5 seconds). The library that we have created uses the speex engine to compress the audio data. This library implementation is done natively on the desired platform. The time taken by the library can vary based on the device platform and architecture. Moreover the library can be optimized to carry out its functions even faster. Device specific native API's are observed to carry out these functions faster than our proprietary library.

The time taken by the speech engine (S2T and TTS)can be observed to be generally high, around 4000 ms. The S2T time is relatively higher than the TTS time. This is mainly because the speech engine has to decompress the audio file and then process the audio to synthesize the text. We believe that the time taken by the speech engine can be further reduced by optimizing it or replacing it with a better speech engine.

The average time taken by the enterprise service, that is in our case the names service, is very low (13 ms). The average time taken by our backend application that is hosted on the BES is around 4 ms. It is responsible for processing the https request and logging all the queries. The BES is composed of a fast hardware and hence the processing times are significantly reduced.

After studying this statistics, weextended our study on multiple OS platforms and different devices, to determine how timings differ across these platforms and devices. The following subsections summarize our results.

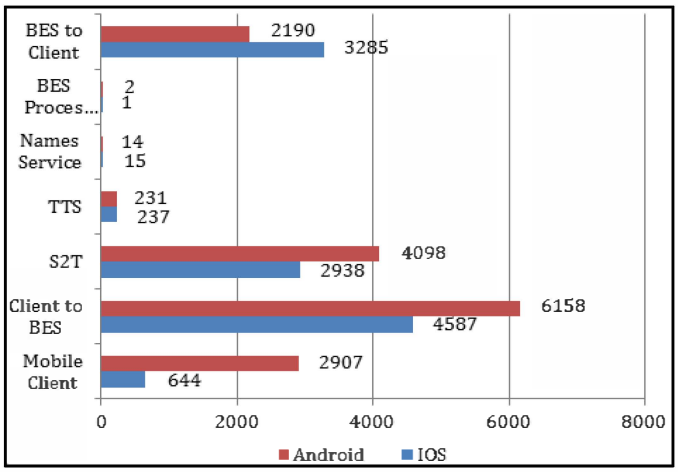


Figure 4. Android(S4) vs IOS(iPhone 5)

1) Android vs iOS

For this study, we send concurrent requests on both the devices and record the timing metrics as earlier. The devices used were Samsung S4 for Android and iPhone 5 for IOS. The results are as shown in figure 4.It can be observed that the *time taken by ourhybrid voice agent on Android is almost three times less than the iOS client*. The Samsung S4 device has an eight core CPU of which four cores are used at a time while the iPhone has a dual core CPU. Moreover the S4 has more RAM than the iPhone device. These architectural differences greatly reduce the execution time of the hybrid voice agent on the Android device. The time to transmit data is however higher on the Android client. We investigated the audio length for the same audio samples on both the clients.

TABLE II: RECORDED AUDIO LENGTH

Platform	Recorded byte size
IOS	1529
Android	2203

As seen in Table II the *audio length of the same audio on iOS is much less compared to that on Android.* This is because of the superior microphone hardware on iPhoneand because the voice activity detection library works better on iOS. The silence detection is also done sooner on iOSthan in Android. Due to these reasons, the S2T time too is also increased on the speech engine server. All other times are comparable on both platforms.

We were also interested in determining if it was the device hardware or the platform OS that was the reason for the boost in performance on the Android client, so we compared two Android and iOS devices separately and analyzed the results.

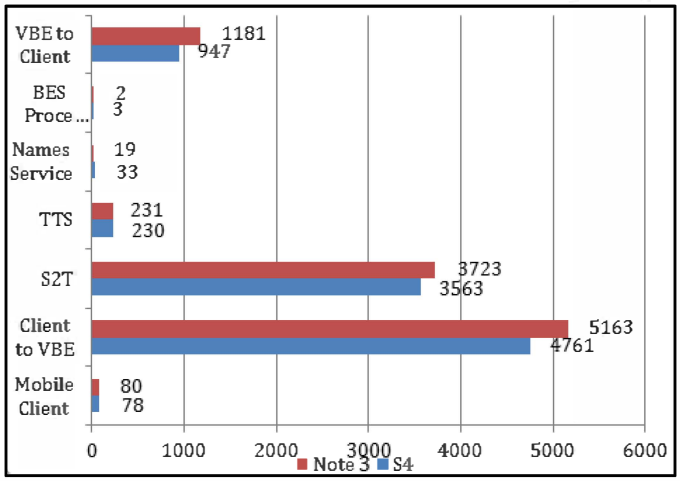


Figure 5. Samsung S4 vs Note 3

2) Android vs Android

To perform this comparison we used two powerful Android devices: Samsung S4 which is the latest S series device from Samsung and the Note 3 which is a "phablet" variant from Samsung. Both pose the same processing power but the Note 3 has more RAM. The results of our study are shown in figure 5. It is not surprising to see that the hybrid voice agent execution time is almost the same on both devices. Both are running the same OS version and have almost the same processing capability. We observed that the average upload time (and download time) on Note 3 is higher than on S4. We also found that the S2T for Note 3 is higher compared to S4 as well. These results can be explained by the higher average recorded byte size on Note 3, which increased upload and S2T times. The voice library was however better at detecting voice activity on Note 3.

3) iOS vs iOS

To compare two iOS devices we used the latest iPhone that is iPhone 5s and its predecessor iPhone 5. Both have the same iOS version. But the iPhone 5s usesa 64 bit CPU. The results

978-1-4799-6585-4/14/$31.00 ©2014 IEEE

of this study are shown in Figure 6. Inspite of the superior hardware, the hybrid voice agent execution time on the iPhone 5s is higher than iPhone 5. This clearly suggests that the architecture doesn't have a larger role to play when it comes to improving the hybrid voice agent app performance. The OS certainly has to be optimized to accommodate hybrid apps. iOS takes four times longer than Android for the hybrid portion of the voice agent app to complete execution. This clearly suggests that Android is optimized for hybrid apps while iOS is not. The upload and download times on both the iOS devices do not seem to vary much. The difference in upload times (~200ms) on iOS devices is less than that on Android devices(~2000ms). This could be a result of the superior microphone on iOS that detects voice activity better.

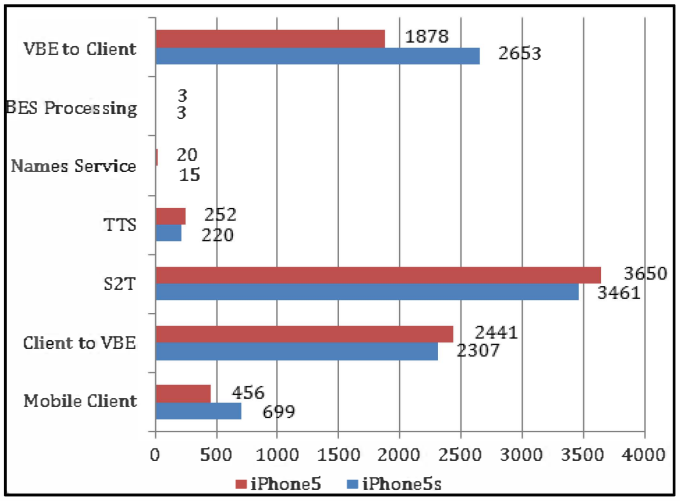


Figure 6. iPhone 5s vs 5

V. CONCLUSION

The increasing prevalence of personally owned smartphones that are almost always network connected, together with secure and fast access to an incredible amount of business data from backend servers hasfostered significant growth in enterprise mobile computing in recent years. Speech is a natural addition to the enterprise mobile capability feature set and presents a whole new way to search and interact with business data. We have created a cross platform mobile voice agent that overcomes the hurdles imposed by differing mobile operating system platform, hardware, languages and interfaces by using a hybrid development approach. In conducting a performance review of the agent we have identified bottlenecks in the execution flow and highlighted the strengths and weaknesses on the state-of-the-art mobile platforms tested.

VI. REFERENCES

[1] Wasserman A.I. - Software Engineering Issues for Mobile Application Development. Proc. 2nd Workshop on Software Engineering for Mobile Application Development MobiCase'11 (2011)

[2] IDC Predictions 2013: Competing on the 3rd Platform. www.idc.com/research/Predictions13/downloadable/238044.pdf (Accessed in August 2013)..

[3] Ambler,Scott W, "Agile Enterprise Architecture", Agile Enterprise Architecture: Beyond, Enterprise Data Modelling, 2006.

[4] Mobile:WebApps, Native apps and Hybrid apps, http://www.nngroup.com/articles/mobile-native-apps/

[5] Apache Cordova, http://cordova.apache.org/

[6] H. Heitkotter, S. Hanschke, and T. A. Majchrzak. Comparing cross-platform development approaches for mobile applications. In Proc. 8th WEBIST, 2012

[7] Facebook's Zuckerberg:'The biggest mistake we've made as a company is betting on HTML5 over native.', http://venturebeat.com/2012/09/11/facebooks-zuckerberg-the-biggest-mistake-weve-made-as-a-company-is-betting-on-html5-over-native/

[8] Challenges of Mobile Computing, http://software.intel.com/sites/manageability/AMT_Implementation_and_Reference_Guide/default.htm?turl=WordDocuments%2Fchallengesofmobilecomputing.htm

[9] Sharon Oviatt. 2000. Multimodal system processing in mobile environments. In *Proceedings of the 13th annual ACM symposium on User interface software and technology* (UIST '00). ACM, New York, NY, USA, 21-30. DOI=10.1145/354401.354408 http://doi.acm.org/10.1145/354401.354408

[10] Halimah, B.Z.; Azlina, A.; Behrang, P.; Choo, W.O., "Voice recognition system for the visually impaired: Virtual cognitive approach," *Information Technology, 2008. ITSim 2008. International Symposium on* , vol.2, no., pp.1,6, 26-28 Aug. 2008

[11] Khalil, R.T.; Khalifeh, A.F.; Darabkh, K.A., "Mobile-free driving with Android phones: System design and performance evaluation," *Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on* , vol., no., pp.1,6, 20-23 March 2012

[12] Nagata, Y.; Fujioka, T.; Abe, M., "Speech enhancement based on auto gain control," *Audio, Speech, and Language Processing, IEEE Transactions on* , vol.14, no.1, pp.177,190, Jan. 2006

[13] Fadi Biadsy and Pedro J. Moreno and Martin Jansche,"Google's Cross-Dialect Arabic Voice Search",IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012)

[14] Liu, L.; Moulic, R.; Shea, D., "Cloud Service Portal for Mobile Device Management," *e-Business Engineering (ICEBE), 2010 IEEE 7th International Conference on* , vol., no., pp.474,478, 10-12 Nov. 2010

[15] Zhanlin Ji; Xueji Zhang; Ganchev, I.; O'Droma, M., "Development of a Sencha-Touch mTest Mobile App for a mLearning System," *Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on* , vol., no., pp.210,211, 15-18 July 2013

[16] Sin, D.; Lawson, E.; Kannoorpatti, K., "Mobile Web Apps - The Non-programmer's Alternative to Native Applications," *Human System Interactions (HSI), 2012 5th International Conference on* , vol., no., pp.8,15, 6-8 June 2012

[17] Li Tian; Huaichang Du; Long Tang; Ye Xu, "The discussion of cross-platform mobile application based on Phonegap," *Software Engineering and Service Science (ICSESS), 2013 4th IEEE International Conference on* , vol., no., pp.652,655, 23-25 May 2013

[18] IBM Worklight, http://www-03.ibm.com/software/products/en/worklight/

[19] O'Shaughnessy, D., "Interacting with computers by voice: automatic speech recognition and synthesis," *Proceedings of the IEEE* , vol.91, no.9, pp.1272,1305, Sept. 2003

[20] Dojo Mobile Toolkit, http://dojotoolkit.org/features/mobile

[21] HTTP Secure, http://en.wikipedia.org/wiki/HTTP_Secure

[22] Xinyang Feng; Jianjing Shen; Ying Fan, "REST: An alternative to RPC for Web services architecture," *Future Information Networks, 2009. ICFIN 2009. First International Conference on* , vol., no., pp.7,10, 14-17 Oct. 2009

[23] Javascript Object Notation, http://en.wikipedia.org/wiki/JSON

[24] Nuance Dragon, http://www.nuance.com/dragon/index.htm

[25] LDAP Directory, http://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol

978-1-4799-6585-4/14/$31.00 ©2014 IEEE