Enterprise App Store for Cross Platforms

Nathan M. Balakrishnan, Mohd Azrul Hadi Abd Aziz, and Fauzi Mansoor Mohd Ibrahim

***Abstract*—As mobile devices becoming necessity and popular way for performing certain tasks, it has also impacted on expectation of IT services at enterprises. Employees and entities related to the enterprises are expecting IT departments to adhere to this trend by moving forward with consumer based IT services as conditioned in mobile world. IT heads need to respond to these demands with careful strategies where the solution should cater for various mobile platforms and at the same time without impacting the investment already made on desktops/notebooks by not breaking support on later users. The strategy also should include lower cost of mobile apps development by adopting write once and run anywhere approach. In this paper we discuss a strategy and approach for above stated problem by implementing a prototype model ofenterprise app store. This model is variant of existing app stores where it distributes apps which are mini web based applications that rendered by web client on browsers on various platforms. Our goal is to provide enterprise app store services to seamless mobile and computing devices without multiplying cost.**

***Index Terms*—Enterprise app store, mobile enterprise application platform, IT services consumerization.**

I. Introduction

The ever increasing demand for smart phones and tablets has been catalyst for flooding of mobile apps. Mobile apps play a main role in the mobile environment or infrastructure to expedite the growth of the mobile device usage.References [1]-[4] predicted that from 2009 to 2014 a growth rate of 677% on mobile apps deployment and distribution through current channels like App Store, Play Store and Market Place. Lately these developments have created excitements in the business world as well that the enterprises have developed growing interest in mobile apps as tools for their operations [2], [4]. Imagine if all business functions encapsulated as mini applications that could be downloaded from enterprise app store using role based authorization can be created, after careful analysis of the applications that would enable the highest worker productivity and adoption. [5] Likened it as glue that connects end users to all significant steps in any process. Employees as well as customers, suppliers and business partners would use the mobile apps that make most sense to them, based on which ones apply to their situation and make them most productive. Also it enhances the reach of corporate IT beyond the borders of the company [6]. Business and technology analysts (e.g., Gartner, IDC) have ranked this approach among the top strategic IT trends [7]. As part of the initiatives towards using mobile infrastructure at the workplace and also as services for their customers enterprises are beginning to adopt “Bring Your Own Device” BYOD strategy [5], [8]. Wikipedia describes BYOD as policy implementation for permitting employees to bring their personal owned mobile devices (laptops, tablets, and smart phones) to their workplace and to use those devices to access privileged company information and applications [9].

However enterprises are not just targeting at developing mobile apps alone for their operations but rather implementing their own private app store as their distribution channel as suggested in [10]. Implementing private corporate or enterprise app store gives the enterprises the advantages of developing mobile apps of their own standard thus not limited to the established standards by Apple, Google or others. More over having own app store empowers enterprises with absolute control on type of apps to distribute, apps distribution and its lifecycle management. It also enables them greater flexibility for their data access with better protection. Even though there are a lot of benefits on having private app store, however implementing it impose certain challenges. First employees, vendor, suppliers and customer of an enterprise are not limited to use one type of mobile device alone. As such it is important that the single app store is capable for cross mobile device apps distribution. Second it is essential that the apps distributed through this app store also executable on desktop and notebooks as well. This is vital because unlike public app stores where consumers are expected to use mobile devices users at enterprises are bound to use desktops or notebooks to perform certain tasks. Moreover enterprises also want to maximize their huge investment already made on those machines. Third the apps distributed through this app store should adhere to write once and run everywhere strategy meaning the same version of app should be executable on cross devices including desktop or notebook. This is extremely crucial factor for cost saving. Enterprise can save cost on the apps development where apps that developed by developer with particular skill set can be executable on multiple device. This also saves enterprises on complexity of maintaining apps compatibility issue for different devices. An app store that provides solution for the above raised cross platform and cost issue would be ideal for enterprise level implementation. In this paper we are proposing a new model of web based app store for enterprise that is truly cross platform compatible. Section II of this paper will discuss on related works on this area followed by discussion on the work done in Section III. In Section IV we will discuss on our prototype model implementation meanwhile Section V will be on conclusion.

II. Literature Review

Gartner report [11] indicates that by 2018 25% of large organizations will have strategies to make their corporate

Manuscript received August 18, 2014; revised October 28, 2014.

The authors are with the Software Development Lab Kulim High Tech Park, MIMOS Berhad 57000 Kuala Lumpur, Malaysia (e-mail: nathan.bkrishnan@mimos.my).

DOI: 10.7763/IJFCC.2015.V4.374

computing environment similar to consumer computing experience. This indicates the rise of IT consumerization trend. IT consumerization refers to the blending of personal and business use of technology devices and applications. This blending of personal and business technology will have significant impact on corporate IT departments, which traditionally issue and control the technology that employees use to do their jobs.

To accommodate this demand some vendors are beginning to provide app store services at enterprises as solution [12]. The service is having internal app store [8] as mobile apps distribution channel rather than using public channels such as Apple's App or Google's Play store. However the apps are type of native which are built for device specific using SDKs provided by prominent player in this industry such as Apple, Google, Microsoft and Blackberry. The distribution channel is usually a web based management console that approves and give access rights based on user's roles to the enterprise. On the other hand, users need to install technologies such as "Hypervisor" or "Containers" on their devices to integrate to the store for downloading and installing the apps.

While this approach seems convincing it imposes certain challenges to the enterprises. As this solution requires the development of target device specific apps, the enterprise would need to develop and maintain different versions of same app for as much as mobile devices available out there. This will be an overhead for the enterprises as they need to acquire and maintain resources (developers specific for different mobile platforms) to develop apps for different devices. Moreover there will be possible apps compatibility issues if the mobile devices OS are upgraded. This situation is further complicated with different tools used for development, build systems, APIs, and devices with different capabilities for each platform. And off course these apps would not be able to be executed on desktop and notebooks.

Another work [13] proposed architecture of combining Client/Server (CS) and Browser/Server (BS) architectures to form Widget/Server (WS) architecture for widget based application framework. However the paper doesn't describe how this architecture can be matured to support implementation of enterprise app store. The essentials of mobile apps related to manufacturing environment are presented in [14]. This paper highlights the importance of enterprise app store for distributing apps in a unified way across mobile device without discussing the mechanism.

III. Cross Platform App Store for Enterprise

Our core work is on developing model of app store that distributes apps for enterprise that are written once and executed at mobile devices as well as desktop or notebooks. Generally apps are refer to applications for novel mobile consumer devices with a touch screen, especially smart phones and tablet PCs, however until now, there is no clear common definition of apps and for that reason apps can take broader sense where apps stand for any form of application on a computing device, no matter whether on a smart phone, desktop or notebook [14]. As such the model should be based on a technology that is commonly available in all the mobile devices and notebooks. The only thing these devices' have in common is that they all shipped with browsers. Exploiting this advantage to create an app store that distributes web based mini applications or apps that are render-able on web client launched at users' devices is the main focus of enterprise app store model.

With our experience with public app stores such as Apple store or Google store, we can conclude that app stores comprises of three main components. They are:

1) A central repository as channel for developers to distribute or sell their application and for users to consume them. In some cases where an extensive apps security and vulnerability checking mechanism is needed, a workflow for approval process is integrated with centralized repository system.
2) A mobile device that runs an OS at users' end that enables them to download and execute the apps.
3) The mobile app which is a form of executable program that can be executed on mobile device it is intended for. A programmable module (e.g. SDK is made available for developers to develop apps according to standards and limitations as required by the store owner.

Generally our app store model follows above described common components structure with extension to authentication and authorization services as depicted in the Fig. 1 below:

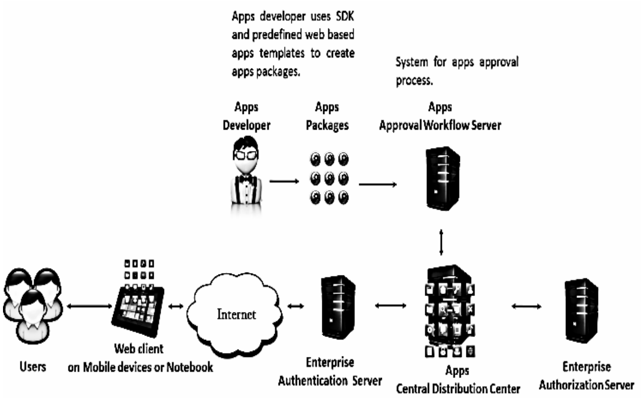

Apps developer uses SDK and predefined web based apps templates to create apps packages.
System for apps approval process.
Apps Developer
Apps Packages
Apps Approval Workflow Server
Internet
Users
Web client on Mobile devices or Notebook
Enterprise Authentication Server
Apps Central Distribution Center
Enterprise Authorization Server

Fig. 1. Architecture of web based enterprise app store.

Authentication and authorization components enable users to download apps based on their role and tasks they perform in the enterprise or with relation to the enterprise. Perhaps the most important feature in our enterprise store model is the web client. As iOS on IPhone and IPad and Android on supported phone needed to execute and manage the apps that are built targeted for them, our web client is essential to render the apps which are mini web application built on standards and formats that we have established.

The web client is a web application built using HTML5 and JavaScript web technologies. The strong drivers to use HTML5 is that it has been upgraded for some important technical and performance limitations of previous-generation web application technology as well as cross-platform compatibility for enterprises to future-proof their mobile environments [15]. The platform and device independence capability enables it to be loaded on various browsers at notebook and on micro browser on tablet and mobile phones. Moreover HTML5 give advantage for developing sophisticated applications with improved user interface and has a strong developer base. Employing responsive design and SPA (Single Page Application) design techniques the web client meets UX (User Experience) requirements for executing apps on small screen devices as well as provides the feel of native application like interaction. It also leverages on browser storage capability together with JavaScript based persistence technologies for back – end synchronization and offline processing support.

The web client can be accessed by navigating the browser to specified URL. Through the web client users' will be able to select and install apps as required as shown in the Fig. 2 below.

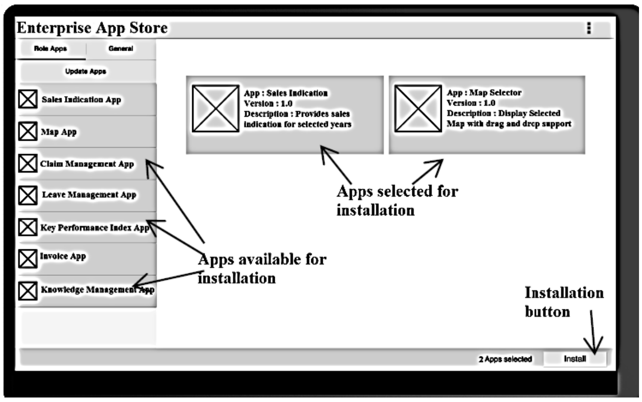


Fig. 2. Wireframe structure of app installation feature in the web client.

Only apps related to their roles will be displayed for user's selection. Since this is web application, no actual installation will happen at the user's device but rather creating reference between the apps and user account. Once user have selected and installed the apps, users can add multiple apps including adding the same instance of app more than once and interact with them through workspace as shown in Fig. 3.

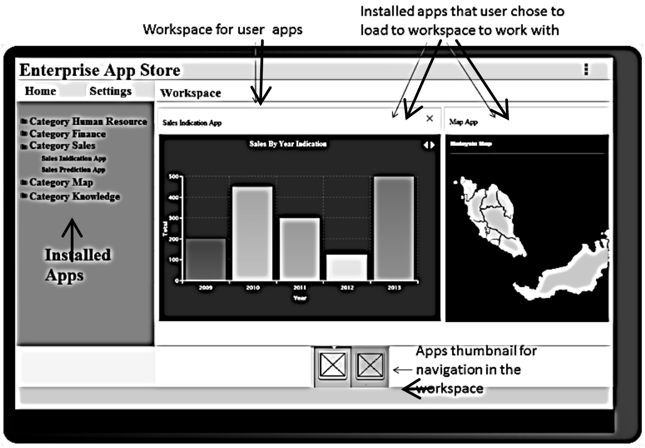


Fig. 3. Wireframe structure of workspace feature in the web client.

This is a significant difference compared to current approach of loading of only one app at a time on the mobile devices. This feature encourages user to work with multiple apps at the same time. It also provides opportunity for app developers to develop apps having features for interaction with other app in the workspace through gestures like drag and drop. Combining capabilities of different apps to perform task would make users to feel more guided and supported. Workspace can be saved for reuse and users can create as many workspaces with different combinations of apps added to them to suite their need to perform different kind of task attached to their role in the enterprise.

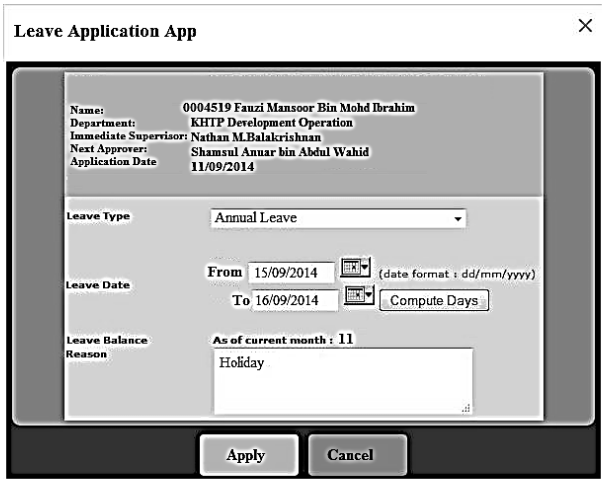
Management and maintenance functionalities are provided as part of setting screen (not shown here). The functionalities included are uninstalling apps, installing new apps, updating installed apps, creating and deleting workspaces, setting favorite apps and others.

The app as described above is a mini web application shown in Fig. 3. Web apps are an important precursor to introduce BYOD for mobile workers [15]. Basically the app is a HTML5, JavaScript, CSS and configuration file assembly that will easily fit itself once installed to and detach from the SPA structure of the web client. Thus it can be developed independently using our SDK that provides basic HTML5 and CSS structure with required JavaScript components. Content inside the app can be anything that supports operations of the enterprise. The data source for apps can be REST or SOAP based web services which are integrated to LOB application data sources.

IV. Enterprise App Store Model Prototype Evaluation

A. Prototype Implementation

In order to evaluate the cross platform supportability of the model, a prototype of enterprise app store has been implemented. Leave application and its approval apps as shown in Fig. 4 and Fig. 5 are developed as part of the prototype to test the ability for users to download, install and execute the apps in different devices. An enterprise app store is deployed with the two apps made available to all the users.


Leave Application App ×

Name: 0004519 Fauzi Mansoor Bin Mohd Ibrahim
Department: KHTP Development Operation
Immediate Supervisor: Nathan M.Balakrishnan
Next Approver: Shamsul Anuar bin Abdul Wahid
Application Date 11/09/2014

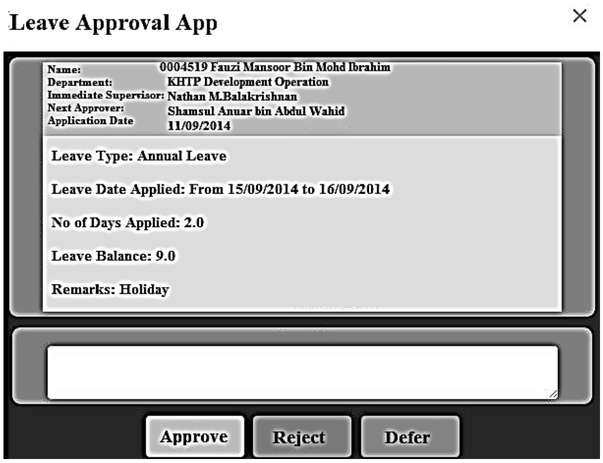
Leave Type: Annual Leave

Leave Date: From 15/09/2014 (date format : dd/mm/yyyy)
To 16/09/2014 Compute Days

Leave Balance: As of current month : 11
Reason: Holiday

Apply Cancel

Fig. 4. Leave application app.


Leave Approval App ×

Name: 0004519 Fauzi Mansoor Bin Mohd Ibrahim
Department: KHTP Development Operation
Immediate Supervisor: Nathan M.Balakrishnan
Next Approver: Shamsul Anuar bin Abdul Wahid
Application Date 11/09/2014

Leave Type: Annual Leave

Leave Date Applied: From 15/09/2014 to 16/09/2014

No of Days Applied: 2.0

Leave Balance: 9.0

Remarks: Holiday

Approve Reject Defer

Fig. 5. Leave approval app.

Three different categories of users on two different roles are identified to test the prototype model. The roles are leave applicants and approvers who are divided into notebook users, tablet users and mobile phone users. The model and its specifications of the devices used for evaluation are given in the tables below.

Table I below shows specification for selected devices in notebook category. The devices are Mac Book Pro and Asus notebook. For Mac Book Pro, Chrome, Firefox and Safari browsers are used for testing while for Asus notebook Chrome, Firefox and IE browsers are used.

TABLE I: NOTEBOOK SPECIFICATION

Notebook			
Device		Mac Book Pro, 13-inch with Retina Display	Asus - 15.6" Notebook
OS		OS X Mavericks	Windows 8.1
Processor		2.6GHz 2.6GHz dual-core Intel Core i5 processor (Turbo Boost up to 3.1GHz) with 3MB shared L3 cache	2.16GHz (with Intel® Burst up to 2.41GHz)
Storage		128GB PCIe-based flash storage	4GB Memory - 500GB Hard Drive
Wireless Speed		≈ 4Mbps	≈ 4 Mbps
Resolution		Native resolution: 2560 by 1600 pixels (Retina); scaled resolutions: 1680 by 1050, 1440 by 900, and 1024 by 640 pixels	1366 x 768
Browsers	Chrome	√	√
	Firefox	√	√
	Safari	√	N/A
	IE	N/A	√

TABLE II: TABLET SPECIFICATION

Tablets				
Device		Google's Nexus 7	Windows Surface Pro 3, 12" ClearType Full HD Display	Apple iPad 3 Wi-Fi
OS		Android 4.2 (Jelly Bean)	Windows 8.1 Pro	iOS 5.1
Processor		NVIDIA® Tegra® 3 quad-core processor	64GB/Intel i3 version: 4th generation Intel Core i3-4020Y 1.50 GHz with Intel HD Graphics 4200	Dual-core 1 GHz Cortex-A9
Storage		16GB internal storage	64GB or 128GB version with 4GB RAM	16 GB, 1 GB RAM
Wireless Speed		≈ 4 Mbps	≈ 4 Mbps	≈ 4 Mbps
Resolution		1280x800 (216ppi)	2160 x 1440 Aspect Ratio: 3:2	1536 x 2048 pixels, 9.7 inches (~264 ppi pixel density)
Browser	Chrome	√	√	√
	Firefox	√	√	N/A
	Safari	N/A	N/A	√
	IE	N/A	√	N/A

Table II shows specification for tablet category. The tablets are Google Nexus 7, Windows Surface Pro and Apple IPad 3. For Google Nexus 7only Chrome and Firefox browsers are used. Test on Windows Surface Pro includes Chrome, Firefox and IE browsers while for Apple IPad 3Chrome and Safari browsers are used for testing.

Table III shows testing devices for mobile phone category. Samsung Galaxy S5, Nokia Lumia 1520 and Apple iPhone 5S phone are selected. For Samsung Galaxy S5, Chrome and Firefox browsers are used for testing. Only IE browser is used for Nokia Lumia 1520 testing and for Apple iPhone 5S Chrome and Safari browser are used.

The leave applicants are requested to download and install "Leave Application" app and apply leave through the app. On the other hand, the approvers will use the "Leave Approval" app to approve the applied leaves.

TABLE III: MOBIE PHONE SPECIFICATION

Mobile Phones				
Device		Samsung Galaxy S5	Nokia Lumia 1520	Apple iPhone 5S
OS		Android OS, v4.4.2 (KitKat)	Microsoft Windows Phone 8	iOS 7
Processor		Quad-core 2.5 GHz Krait 400	Quad-core 2.2 GHz Krait 400	Dual-core 1.3 GHz Cyclone (ARM v8-based)
Storage		16 GB, 2 GB RAM	16 GB, 2 GB RAM	16 GB, 1 GB RAM DDR3
Wireless Speed		4 Mbps	4 Mbps	4 Mbps
Resolution		1080 x 1920 pixels, 5.1 inches (~432 ppi pixel density)	1080 x 1920 pixels, 6.0 inches (~367 ppi pixel density)	640 x 1136 pixels, 4.0 inches (~326 ppi pixel density)
Browser	Chrome	√	N/A	√
	Firefox	√	N/A	N/A
	Safari	N/A	N/A	√
	IE	N/A	√	N/A

B. Evaluation Result Analysis

As expected the app store prototype works well on all three device categories. The users of all three categories able to download install and execute the apps on their devices. Based on analysis of users feed backs, generally highest quality of service is achieved on desktops and notebooks followed by tablets and with acceptable delay on mobile phones. This is mainly because capability of the browsers and the processing speed of the devices. Responsive nature of the prototype through the responsive design employed provides the user of the three categories seamless user experience in using the store.

The result of the evaluation clearly indicates the objective of the app store to support cross platform is achieved. Thus this model can be further enhanced for real implementation where office operations are encapsulated in the form of apps described in the model and distributing them through the store for enterprise employees and others related to the enterprise. This would definitely enhance enterprises ability to prepare itself for emerging trend such as IT consumeraization.

V. CONCLUSION

It is reported in [16] by 2015 tablets shipments will reach around 50% of laptop shipment. This trend gives impression that mobile devices will be more employed in workplaces. In line with this trend, [17] is recommending IT departments in enterprise need to align strategies to change from business computing characteristics to consumer computing characteristics for more flexible work environment that better accommodates a rapidly changing workplace. This signals messages that enterprises should be getting ready to implement app stores to effectively positioning them to respond to demands of more consumerization of IT services. However enterprise should be careful in their implementation strategies as their app store services need to provision across mobile devices without breaking support for desktop or notebook users and at same time not multiplying cost for providing the services. In this paper we have presented our model of developing a web based enterprise app store which can meet above expectations. It comprises of three main components which are centralized apps repository for apps distribution, a web client that renders the apps enabling users to install and use and manage the apps and the apps which are mini web applications that can be developed independently using supplied SDK. The whole app store components are built on web technologies such as HTML and JavaScript enabling it to be executed on browsers on mobile devices and desktop/notebooks.

References

[1] Distimo, *Full Year 2010*. Distimo database, 2011.

[2] F. Sullivan. (2009). *An Insight into the U.S. Smartphone Application Storefront Market*. Frost and Sullivan. [Online]. Available: http://www.researchandmarkets.com/reports/1052873/an_insight_into_the_u_s_smartphone_application

[3] S. Ellison, "Market analysis-worldwide and U.S. mobile applications, storefronts, and developer 2010 – 2014 forecast and year-end 2010 vendor shares: the "appification" of everything," *International Data Corporation*, 2010.

[4] *Google's Android Becomes the World's Leading Smart Phone Platform - Canalys Reveals Smart Phone Market Exceeded 100 Million Units in Q4 2010*, 2010.

[5] O. Leif, D. Ken, and K. Stephen, *Strategic Road Map for Endpoints Using Mobility as the Key Driver*, 2014.

[6] S. Wenzel, "App Store Models for Enterprise Software: A Comparative Case Study of Public versus Internal Enterprise App Stores," in *Proc. the 5th International Conference on Software Business*, 2014, pp. 227-242.

[7] C. Pettey, *Gartner Identifies the Top 10 Strategic Technology Trends for 2013*, 2012.

[8] D. Jaramillo, "Cooperative solutions for bring your own device (BYOD)," *IBM J. Res. and Dev*., vol. 57, no. 6, 2013.

[9] Bring Your Own Device. [Online]. Available: http://en.wikipedia.org/wiki/Bring_your_own_device

[10] G. Andrea, S. Katarina, and D. Bastiaan, "Mobile enterprise applications - current state and future directions," in *Proc. the 45th International Conference on System Sciences*, 2012, pp. 1363-1372.

[11] M. Cain and M. Gotta, *CIO Survey Uncovers the Need to Focus on Digital Workplaces and Engaged Workforces*, 2014.

[12] U. Pratap and R. K. Srivastava, "Transforming business with mobile enterprise apps," *International Journal of Engineering and Computer Science*, vol. 2, pp. 2057-2066, June 2013.

[13] X. Zhiqing, W. Si, Y. Heqi, W. Zhenyu, C. Hao, Z. Chunhong, and J. Yang, "A new architecture of web applications –The widget/server architecture," in *Proc. the 2nd IEEE International Conference on Network Infrastructure and Digital Content*, 2010, pp. 866-869.

[14] G. Christoph, S. Silcher, E. Westkamper, and B. Mitschang, "Leveraging apps in manufacturing. a framework for app technology in the enterprise," in *Proc. the 46th CIRP Conference on Manufacturing Systems*, 2013, pp. 664-449.

[15] Intermec. (2013). *Does HTML5 Make Sense for Mobile Enterprise Applications?* [Online]. Available: http://www.intermec.com/public-files/white-papers/en/wp-HTML5-mobile-enterprise-applications.pdf

[16] C. David, *Top 10 Strategic Technology Trends for 2013*, 2012.

[17] W. C. Matthew and G. Mike, *Create a Digital Workplace to Respond to Critical Changes in the Workforce*, 2014.

Nathan M. Balakrishnan is attached to MIMOS Bhd, Software Development Lab of ICT division as a staff engineer. In MIMOS he is responsible for knowledge management platform development as well as various software development projects. Currently he is also exploring mobile based systems for enterprise operation support. He has over 14 years of experience in software development field.

He has progressed to architect and design systems as well as managing development team to deliver quality products to customer. Nathan has vast experience working on enterprise level applications. His skill-set includes development experience using Microsoft .NET framework, C#, ASP.NET, WCF, WPF, JavaScript and NodeJSplatfrom. He holds degrees both master and bachelor (Hons) in the field of computer science. He has published paper on enterprise mash-up systems and filed patent in the field of enterprise knowledge management.

Mohd Azrul Hadi Abd Aziz is a staff engineer at MIMOS Berhad. He is currently attached to Software Development Lab. His primary role is as a senior developer for Knowledge Management platform. Mohd Azrul Hadi has 13 years of vast development experiences in system development ranging from simple thin client, mobile application, RFID systems to wide enterprise level systems.

He has in-depth experiences on the following technologies which includes Microsoft .NET framework, C#, ASP.NET, Silverlight, WCF, WPF, Entity Framework, JSON, HTML and JavaScript. He holds a bachelor of information technology (Hons) degree. He has co-authored paper on enterprise mash-up systems and filed patents in the field of enterprise knowledge management.

Fauzi Mansoor Mohd Ibrahim is a software developer at MIMOS. He works on various software development projects. His development skillset includes C#, WPF, HTML and JavaScript. He has a degree in computer science and currently pursuing master degree in software engineering.