Concept of the Personal Devices Content Management Using Modular Architecture and Evaluation Based Design

Miroslav Behan and Ondrej Krejcar

University of Hradec Kralove, FIM, Department of Information Technologies,
Rokitanskeho 62, Hradec Kralove, 500 03,
Czech Republic Miroslav
Behan@uhk.cz, Ondrej.Krejcar@ASJournal.eu

Abstract. Personal devices as smart phones, smart watches, and smart glasses will be considered as natural cyberspace interface. The content will be valuable more without fix relation to single device. Therefore increasing amount of personal devices will evolve open standards for content exchange over network and through different application providers. The personal productivity will be more or less influenced by comprehensive form of management over multiple device's types or platforms. The proposed concept is considered as modular where possible changes of system key functionalities are various modules dependable on behavior or type of person.

Keywords: Mobile, Device, Management, Smart, Sensor, Content.

1 Introduction

The increasing mobile devices computing power influences development of smart environment solutions and propose more human productivity possibilities. We acknowledged that nowadays the current market with mobile devices is more and more fragmented and therefore we propose personal device content management as alternative to overflowed types and kinds of devices. Also remote device control area in approach to management device groups by remote interface are known as challenge nowadays. We would like to present concept of personal device content management as a future vision of multi device environment approach.

Personal content delivery based information system to develop is complex task where are different resources over different platforms. The beneficial approach in development such a kind of private cloud service or application is in modular scalable architecture where separate parts are built and tested separately and are customizable within system.

As a device we define all that devices, which are able to connect to network resp. to the internet sources using an online or offline mode. The multi device environment from single person point of view naturally underline future realistic scenario where user would own or have to manage more than one device. We

P.C. Vinh et al. (Eds.): ICCASA 2012, LNICST 109, pp. 151–159, 2013.

acknowledge multi-vendor environment and multi-platform environment as Android [8], iOS, Mango and others. Currently many users have more than one device which would be as an interface to cyberspace or which would be as an extension to visualize electronic world. The scenario would be about connectivity with cyberspace and user in conventional way more and more. The basic idea of multiple device management is based on simplify and user friendly environment, where the same user interface (UI) for different devices or type of devices from multiple vendors or manufactures, is presented. What could happen when user reclaims the same type of device interface for instance mobile phones where the same functionality and content basically is? The user has to know as many device interfaces as many types of platforms exist. What if exists one customize able device interface which accessible the most common features for devices. Is that would be good experience in evaluation of human productivity? What if user could be independent on platforms and type of devices and in case of device crash or device lost, it could be easily recovered by one button click? Even more when user realizes that there is a possibility to multiply devices with same content and when expecting precise content from one device it would be accessed from different device. Of course it is all about capability of devices which could in future leads more and more in massive usage of mobile device smart solutions and mobile device could be as natural connection between the rests others types of devices (e.g. car, fridge or boat).

The Apple platform provides for developers fundamental and well prepared design support with framework named COCOA, which is basically using Object-C as programmatic language. There are others extensions from point of developers view where Java or others scripting languages could be used. The mobile devices are using as operation systems the iOS and the most convenient way for developing application is at using common system calls as application interfaces, application services and core services [2]. The advantages of apple platform are basically comprehensive, publishable and distributive application channel over internet.

Another plus of this platform architecture [10] is one vendor device based solution where the certainty of proper system calls and their behavior is well defined and supported. As well as device hardware access in terms of mobile device development the screen resolution where is ratio between height and width constantly 5:3 could be announced another beneficial aspect in rapid application development (RAD). Apple platform establish fundamentals of mobile application eco-system environment. The increase of usability of mobile device is enormous. Identity of application is consists of small image and short term expression with remote update framework possibility known as application market.

The mobile platform as Android is due to self inter open ability suitable for 3rd party solutions where partial problems are solved [3]. The security and stability of system which is based on open source concept is outstanding [4]. A device types which are currently running under Android platform are well-known smartphones and tablets but other types of devices are suitable caused by advantages of platform for instance laptop, netbooks, smart books, e-book readers, smart TVs, wristwatches, headphones, car players, smart glasses, vehicle navigating systems, refrigerators, home automation systems, games consoles, mirrors, cameras or portable media

players. The architecture of Java based platform fully provides multithreading environment where gathering precise data form sensors are required. The architecture [5] allows services which are running on background to provider or consume external services or content dependable on version [6].

2 Problem Definition of Remote Device Management

In this chapter we summarized problematic areas in device management in consideration of possible remote use. We focused on cross device features which are main base upon which the management provides measurement, controlling and maintenance over sensors or content of device. Other point of view would consider the platforms aspect which are supportive to some key benefits in remote device management. At last the focus would consider the network access and its capabilities in terms of usability and sustainable processing. For better overview we outline ideas expressed in mind map on following figure (see Fig.1)

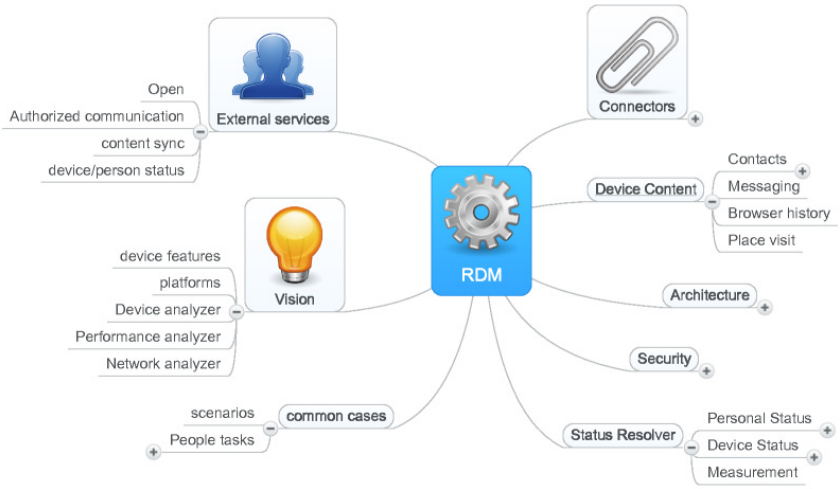

Fig. 1. Mind map – Remote Device Management

2.1 Content Problematic

The core feature of remote device management is multiply, sync or backup device content. Content would be any kind of information but in terms of device management we defined content as end-user data which are important to keep on mobile device caused by daily usage bases or offline mode of device connectivity. As main we recognized Contacts, Photos, Messages or History of communication. Such content would be synchronized over end-user owned devices or be backed up automatically before some of device is lost. Management of content would be more

convenient on desktop device rather than small screen device. On the other hand in time or in location tasks are required event from nonconforming user interfaces for specific tasks of content management.

2.2 Sensor Problematic

The sensors which are informational providers over types of devices would be more in future used due to cost-effective available solutions for daily tasks. As a correct way of gathering sensor data is by using allowed platform system calls where access is authorized by end-user. The other way would be over device management provided by platform or by overriding manufactured firmware by dedicated customized distribution of open platforms which would be available as an open source. We are focusing on allowed 3rd party sensor access which is available through framework API for example in android sensors for smart and ambient environments [Table 1] [12].

The measurement requires at least separate thread to perform precise measured result therefore architecture suites to producer and consumer concept. In case of remote consumer the results wouldn't be influenced by dilation of time of transport or transaction. With consideration of network latency the result would be notified or expected in correct time form. We recognized two groups of sensors where one of them are real-time changed and other group is consisting of state full or long-time change sensors.

We consider as contributory following article [11] where the domain of sensor data gathering is well defined. The informational system as extension would provide user status resolution over sets of gathered sensor data; where sleeping, sitting, running, walking or driving have informational value in current context point of view. Also the environment context is valuable in terms of user productivity for instance vacation, work or distance movement.

3 Concept of Modular Architecture

This article describe solution concept of remote device management focused mainly on server side architecture, where a modular architecture is used [17], [18]. Designed concept is variable in terms of technology use case. We suppose to use as development framework all Java based technology. The reason why is hidden in effectiveness in productivity, scalability and reuse available open source components. The following figure (See Fig. 2) highlighted important parts of architecture which are required for specific needs especially from network connectivity characteristics [1], [22]. The informational system consists of three main parts which are remote client part, core system part and end-user interface part. We starts describe remote part where all possible devices potentially could be connected to system. The devices are highly fragmented hardware area due to competition of manufactures and vendors about end-user goodwill. We recognize basically two sets of devices from system point of view.

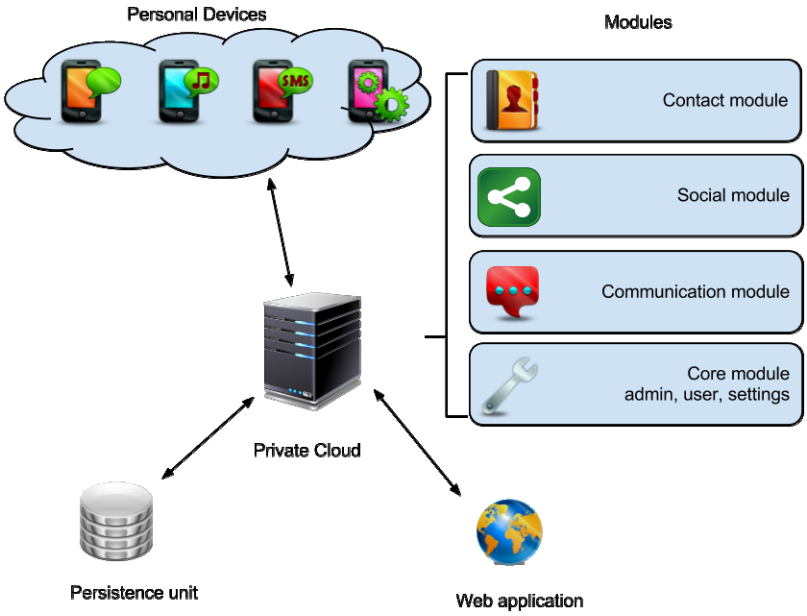

Fig. 2. System architecture – Remote Device Management

The first are all mobile devices which have to care about power supply management and without power saving management the unnecessarily draining battery would lead to uselessness of developed application. The second group of devices is all with an independent power supply where we for instance we classify cars because of their external engine power supplier [19]. For these reasons each of group would behave differently in terms of kind of connectivity mode. The devices which could be connected to system over any kind of network and could provide peer-peer internet connectivity instantly or for exact amount of time we called as active devices. Others we called as passive where which would not be all time online or connected to dedicated server. The passive mode would respect user's defined network connectivity due to cost effectiveness or power management. The connectivity of devices and their statuses accordingly to active or passive device policy are presented in [Table 2][12].

Explicitly defined states of connectivity wouldn't get rid of some cases where the uncertain behavior of network connectivity could appear. The problem would be eliminated by control queuing management based on messages. The messages of events and data would be handled on client side as well as on server side in correct time frame where is explicitly defined. In case of network unavailability the queue substitute consumer and when network status changed and reconnects the message queue processed all FIFO messages in time order. The sentinel measurement data which are with low level importance and are not supposed for real-time processing are lastly consumed. From technical point of view the messages are sent from client to

server by user datagram protocol (UDP) in such low level importance cases and where high measurement precision is not required but is preferred speed of delivery or amount of transmitted data. Otherwise the transport control protocol (TCP) is used for confident types of measurement or data delivery of content [20]. The connection server is consists of datagram resolver and socket transport object resolver where socket resolver due to usage with sensor monitoring and also with command executor is practically core functionality of connection server. The socket resolver either client or server are providing communication between device and system. The main responsibilities are maintenance of connection and transportation of data objects between both sides. Object are transport over serialization Java technology where content is fast serialized or de-serialize on binary code and send over network.

The second part called as system part basically handles core features of informational system from device monitoring, device command execution and device management with predefined device or user policies up to system external data storing, system content providing and system authorized access ability for all kind of requests from front or background processing. Also as a connection server we would use Java programing language for implementation due to object inter-server exchange. What and when would be transported decide application server and client application service. Device execution commands are initiated also by application server where are authorizations of requests dispatched from end-user actions or device routines with associated permissions. Application server is responsible for all other types of requests from device or end-user. The calls consist of group with data visualization calls, group of background routine calls based on time triggering settings and maintenance group or administration group of calls. All calls are related to specific tasks or concrete device or group of devices therefore also security and authorizations is part of responsibility of application server. Last component is data management of informational system based on Java Persistence API (JPA) technology which is being used due to extraordinary developing capability and time development saving. The data objects are defined in Java classes and relations between entities are expressed as member of concrete class with specific annotation which specific cardinality and type of relation. The objects are transformed to database thought persistent commands and after commitment are saved to hard file on disk.

The last part of system is focused on web content delivery and interactions with end-user. The web content is hosted on external web server as a cloud solution where the user identity and cloud services could be used. The web client is connected to web cloud services either to application server over secure channel. Web cloud is used due to implicit network traffic monitoring tool and cost-effective load balancing for web clients with minimal impact to maintenance. Basically is used as secured fast traffic response container for web client which mainly communicate with application server in global world scale where continental redistribution is a case. Web client itself implements data visualization, requests posting and corresponding response handling. The client is based on Hypertext Markup Language (HTLM) version 5 and JavaScript (JS) concept and communication with application server is over transmission protocol (TCP) by Web Socket technology due to convenient and fast responsive way than

classical Asynchronous JavaScript and XML (AJAX) technology. The Web Socket technology provides persist communication channel over well-known port 80 with advantages of socket connectivity. Therefore the reaction time of committed commands in live online mode increases usability entire system where round trip to server is multiple times faster than common XmlHttpRequest (XHR).

4 Evaluation

The evaluation of the proposed concept of system is based on partial implementation where modules are representing key functionality in terms of personal content. The first contact management module using as resource for data minding embedded contact lists of device, social connectors which are accessible over OAuth [21], [22] and other external services based on Restful architecture. The merging process of identities over resources is done on client side and by uniform content provider where data are scored over informational massiveness, social relevancy and frequency of usage.

Table 1. Evaluation of modules functionalities

Module name	Functionality description	Status
Personal	All personal details which are provided for identification, autocomplete and distribute functionalities.	Not implemented yet
Contacts	Contact management module for gathering available resources and merging identities from different providers into comprehensive visual form with smart attributes behavior.	Implemented Android > 2.0
Messages	Communication over message base delivery with fast and full text search history where the smart attributes are included in conversation as location and context is provided.	Not implemented yet
Files	Files synchronization capabilities over USB cable access where device provide memory stick.	Not implemented yet
Apps	Application management and related data are able to be resynchronized	Not implemented yet
Images	Images management with resynchronization capabilities, where optimal filter will be used [23]	Not implemented yet

The evaluation is considered as part of development where increasing quality of result is beneficial. There are numerous evaluation techniques which will be used when concept of system is fully implemented where testers and real users are involved in process of evaluation and usability measurement.

5 Conclusions

The concept of modular personal content devices management is proposed, problematic related to device content is defined and technology principals are designed with evaluation of design. The further discovery would revile capabilities over different platforms with usability of key functionalities for uniform comprehensive device management. Future works will implement proposed concept and test productivity with real users. Benefits of concept are more significant in deployment within real scenario. We considered as mandatory provide uniform access to personal content as private cloud service which would be device or platform independent and accessible anywhere anytime.

Acknowledgment. The work and the contribution were partially supported by the project (1) “SMEW – Smart Environments at Workplaces”, the Grant Agency of the Czech Republic, GACR P403/10/1310; (2) specific research project "Smart Solutions in Ambient Intelligent Environments", University of Hradec Kralove under the project SP/2012/6.

References

1. Conder, S., Darcey, L.: Android Wireless Application Development. Addison-Wesley (2009) ISBN 978-0-321-62709-4
2. Hall, S.P., Anderson, E.: Operating Systems for Mobile Computing. Journal of Computing Sciences in Colleges (2009) ISSN:1937-4771
3. Murphy, M.L.: Android Programming Tutorials. CommonsWare (2009) ISBN 978-0-9816780-2-3
4. Yamakami, T.: Foundation-based Mobile Platform Software Engineering: Implications to Converge to Open Source Software. In: ACM International Conference Proceeding Series, vol. 403 (2009) ISBN:978-1-60558-710-3
5. Android. What is Android? `http://developer.android.com/guide/basics/what-is-android.html` (retrieved May 23, 2012)
6. Android, Platform versions, `http://developer.android.com/resources/dashboard/platform-versions.html` (retrieved May 23, 2012)
7. Open Handset Alliance Overview, `http://www.openhandsetalliance.com/oha_overview.html`(retrieved May 23, 2012)
8. Wikipedia, Android OS, `http://en.wikipedia.org/wiki/Android_operating_system` (retrieved May 23, 2012)
9. Android Developer Site, Sensors, `http://developer.android.com/reference/android/hardware/Sensor.html` (retrieved May 23, 2012)
10. Apple Developer Site, iOS, `https://developer.apple.com/library/ios/#documentation/` (retrieved May 24, 2012)
11. Blazquez, G.G., Berlanga, A., Molina, J.M.: InContexto: multisensor architecture to obtain people context from smartphones. International Journal of Distributed Sensor Networks 2012, Article ID 758789, 15 pages (2012), doi:10.1155/2012/758789

12. Behan, M., Krejcar, O.: The Concept of the Remote Devices Content Management. Journal of Computer Networks and Communications 2012, Article ID 194284, 7 (2012), doi:10.1155/2012/194284
13. Vybiral, D., Augustynek, M., Penhaker, M.: Devices for position detection. Journal of Vibroengineering 13(3), 531–535 (2011)
14. Espada, J.P., Crespo, R.G., Martınez, O.S., Pelayo, B.C., Bustelo, G., Lovelle, J.M.C.: Extensible architecture for context-aware mobile web applications. Expert Systems with Applications 39(10), 9686–9694 (2012)
15. Kasik, V., Penhaker, M., Novak, V., Bridzik, R., Krawiec, J.: User interactive biomedical data web services application. Communications in Computer and Information Science 171, 223–237 (2011)
16. Penhaker, M., Krejcar, O., Kasik, V., Snášel, V.: Cloud Computing Environments for Biomedical Data Services. In: Yin, H., Costa, J.A.F., Barreto, G. (eds.) IDEAL 2012. LNCS, vol. 7435, pp. 336–343. Springer, Heidelberg (2012)
17. Pittera, T., D'Errico, M.: Multi-purpose modular plug and play architecture for space systems: Design, integration and testing. Acta Astronautica 69(7-8), 629–643 (2011), doi:10.1016/j.actaastro.2011.04.002
18. Sojka, M., Pisa, P., Faggioli, D., Cucinotta, T., Checconi, F., Hanzalek, Z., Lipari, G.: Modular software architecture for flexible reservation mechanisms on heterogeneous resources. Journal of Systems Architecture 57(4), 366–382 (2011), doi:10.1016/j.sysarc.2011.02.005
19. Rodriguez-Ascariz, J.M., Boquete-Vazquez, L.: Transforming PC Power Supplies Into Smart Car Battery Conditioners. IEEE Transactions on Education 54(3), 366–373 (2011), doi:10.1109/TE.2010.2059704
20. Horalek, J., Sobeslav, V.: Datanetworking Aspects of Power Substation Automation. In: International Conference on Communication and Management in Technological Innovation and Academic Globalization, Puerto De La Cruz, Spain, November 30-December 02, pp. 147–153 (2010)
21. Tomaszuk, D., Rybiński, H.: OAuth+UAO: A Distributed Identification Mechanism for Triplestores. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part I. LNCS, vol. 6922, pp. 275–284. Springer, Heidelberg (2011)
22. Jung, J.J., Ou, C.M., Nguyen, N.T., Kim, C.G.: Advances on agent-based network management. Journal of Network and Computer Applications 35(6), Special Issue: SI, 631–632 (2010), doi:10.1016/j.jnca.2010.07.009
23. Sharmin, N., Brad, R.: Optimal Filter Estimation for Lucas-Kanade Optical Flow. Journal Sensors 12(9), 12694–12709 (2012)