ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 282 - 287 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

Cloud based Structure Approach of Content-As-A-Service for Supplier Impartial of Mobile Gadgets

[#1]Maddali M. V. M. Kumar, [#2]Rajesh Ghanta
[#1]Assistant Professor, Department of MCA,St. Ann's College of Engineering & Technology, Chirala, Andhra Pradesh
[#2]Assistant Professor, Department of CSE, Vizag Institute of Technology, Visakhapatnam, Andhra Pradesh
[#1]mvmkumar.mtech@gmail.com, [#2]rajesh.ghanta@yahoo.co.in

Abstract - These days, mobile gadgets have turn into a commodity, people use several gadgets for various purposes. While people carry only some of their gadgets with them, people still want to access content originating from any device. To overcome this issue, content is often uploaded into a hosting service available in the cloud.However, using a cloud-based hosting can alienate the control and ownership of the content, in particular in the context of mobile gadgets that upload their content into a cloud-based service. In this paper, Structure approach the development of a cloud computing service for mobile gadgets from a different angle. The main premise of Structure approach is to maintain the content in the device where it was first formed.The resulting design leads to a mobile device cloud,where gadgets, collected with the content and possessions they host, are preserved as outstanding cloud citizens. The proof-of-concept implementation, presented in the paper, is based on regular web protocols,and the core design can be configured for various contexts,such as individuals that have several mobile gadgets, the use of social communities interested in sharing content,or companies.

Keywords: Mobilecloud,mobiledata,mobilegadgets,contentmanagement, CMS

INTRODUCTION

Within the last two decades, owning mobile gadgets has transformed from a novelty to a commodity. Among other things, the commoditization means that the number of gadgets is exceeding the number of people who use them. Thus, when considering the level of individuals, multiple different mobile gadgets are owned and actively used for different purposes by a single user. Furthermore, the mobile gadgets have become an important application platform and a gateway to the Web. The increasing CPU speeds, memory capacity, network bandwidth and "all-you-can-eat" monthly network service plans are rapidly making mobile web usage and mobile software applications more alluring for an average consumer than conventional personal computers.

This commoditization of mobile gadgets has led to numerous usage patterns for different gadgets but a notable trend is to create content using a mobile device. For instance, it is possible that a general- purpose device is used for casual photographing along the way on a trip, whereas a more sophisticated camera is used when more professional photographing takes place. Similarly, people often use smartphones during business hours, whereas a small and robust device is used while performing a physical exercise. What is common to almost all of these gadgets is that while in use, content created locally to the device keeps accumulating in them however, the content is seldom considered solely as a private property of a particular device, or even a user. The ability to seamlessly and conveniently share content between the different gadgets is becoming increasingly important, for different purposes using more and more of different types of connected gadgets. While the complexity and features of individual gadgets varies, it is not uncommon that they all produce shareable content, which is meaningful to view also from other kinds of gadgets.

Furthermore, gadgets are increasingly web enabled, and can therefore upload all of their content to the web, if necessary.For sharing purpose, various hosting services exist where the content can be uploaded and shared with friends and relatives. However, relying on a service provider raises trust and usability issues. The service providers may gain access, as well as numerous privileges, on all the content of an individual. Furthermore, once the content has been uploaded to a service, changing the service provider for another can be difficult, in particular when considering the removal of the content from the service. A user has also to make selection what content is actually shared and to whom since, e.g., a large number of photos can be taken but only few are worth sharing. In general, sharing requires explicit effort from the user including even selecting the specific service, or even multiple ones, depending on content type, and for setting privacy attributes knowing the virtual identities of friends, which can alter from service to service. In the end, it might be easier for the user not to share the content or send it by email.

In this paper, describe a Structure for an approach for managing the content created in and distributed between numerous mobile gadgets: a mobile device cloud, or a mobile cloud for short. In contrast to current web hosting and cloud computing services and content management systems, the main premise is to store the created content on the gadgets. Instead of a central server into which everything is uploaded, each device provides its own content to others as a service when requested. This distributed hosting of content is managed by a server that maintains information about the users, gadgets and privileges associated to the content, for instance. Furthermore, the server is able to perform caching, so that mobile gadgets will not be overloaded if the same content

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 282 - 287 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

is accessed all over again. Since the content remains on the gadgets, there is no infrastructure that would lock the user to only one dealer, as is the case with many presently available content management systems. Central issues for the approach include also quality attributes such as usability, security and trust. However, the Structure defines principles for a general architecture where, e.g., security is not dealt with beyond rudimentary access control. A mobile device cloud established this way forms a natural extension to the existing cloud infrastructures, simply making mobile gadgets, together with their resources, first-class cloud citizens.

Furthermore, considering the content as a service enables extensions to other kinds of services. First, existing services, such as Facebook (http://www.facebook.com/), Twitter (http://www.twitter.com), or emails for that matter, can be added in the system as additional gadgets, making their content available within the same framework as well. Second, in addition to content, other device resources that cannot be uploaded to anywhere, such as camera and microphone, can be made available in the device cloud for the creation of compound services over multiple gadgets.The corresponding proof-of-concept system, which include both server-side and device-side components, has been introduced at the level of implementation detail in more technical articles in a piecemeal fashion. However, no comprehensive presentation on the fundamental research goals or lessons learned in the process has been produced in a condensed form before this paper.

EXISTING CONTENT MANAGEMENT SYSTEMS

As the number of gadgets people actively use, also the amount of content included in them increases, calling for improved content management facilities. Although the term content is central and commonly used in connection with the web, it is somewhat unestablished what the term precisely means. The closest definitive terminology, used by Society of Motion Pictures and Television Engineers (SMPTE) and the European Broadcasting Union (EBU), divides content to two parts, essence and metadata. Of these, essence is the core part of the content, defined in (Mauthe and Thomas 2004) as the raw programme material itself. In contrast, metadata contains information about the essence and its representations. Examples of metadata include title of the content, a location related to the content, and the format of the content.With the above definitions, a system that is used to manage content - including both the essence and associated metadata - is referred to as content management system. Essential features of any content management system have been studied by numerous authors (e.g. Spedalieri et al 2007, Curtis et al1999, Cranor et al 2003), and they include aspects, such as that the content must be stored in a location that is safe and easy to access regardless of time and place, the system must handle all phases of the life cycle of the content, and navigation and searching the content must be easy and natural for the end user.

For mobile content, there are presently roughly two commonly used ways to manage content that build on the facilities of the web. One is to synchronize everything to the web in a device or manufacturer specific fashion, and the other is to use a service that manages the content for you.When using device synchronization services, one in essence creates a backup of the device using, for instance, software provided by the device manufacturer. Then, assuming that the device malfunctions or a new device is purchased, the data can be restored from the service and installed into the new device with relative ease. However, in the worst case this creates dealer lock-in, and inherently is not collaborative enabling sharing between different gadgets and users. In cases where the approach can be extended over a single manufacturer, services are often cumbersome to use. For example, Apple's iCloud (http://www.apple.com/icloud/) supports dealer-specific gadgets and generic web browsers, but enables almost any imaginable content, including photos, videos, and music just to name a few content types, which makes the service a hybrid service that can be considered as a backup and a sharing mechanism, in particular between different gadgets of an individual user. However, sharing between users and between different dealers' gadgets is not convenient and fully supported.

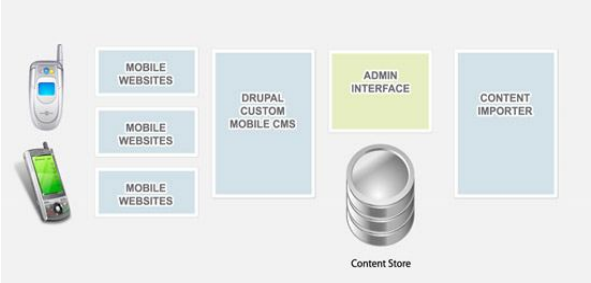

Fig. 1: Content Management Design

Another way to manage content is to upload it to a specific service for further use, be it personal orcollaborative with other users. Numerous different cloud computing systems exist that enable this, with different characteristics. In general, uploading requires user intervention, relies on pushing data without knowing does anyone ever wish to access the data, and raises data ownership issues. Moreover, if only some data is uploaded, content management must also be performed within the device itself. Two different breeds of service-based contentmanagement systems exist. The service can be available for generic content in the form of files for instance, or tailored for certain type of content. The former offers a simple model to allow a variety of content types - for example DropBox relies on a file-system and the service can be accessed from a large variety of gadgets. However, content management operations are relatively limited. In contrast, services that specialize tocertain type of content - such as Flickr which specializes on images - can provide rich operations and metadata on the supported content at the expense of the generality of the content.

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 282 - 287 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

When using one or several of these services, content is uploaded and copied to multiple locations. As a consequence, users may start losing their touch to the content, and the effort of managing content becomes quite a burden. For instance, if a user wants to delete some content she may need to browse through multiple services by removing the copied content, and still she cannot be sure if the content was actually removed from these services. Also different pricing models and conditions of each service may bring additional hassle. For example in the case of Flickr, where the amount of free image uploads is limited, users may temporarily loose access to some of their content if their Pro account expires. Also the free account may be deleted if it is not used for 90 days.
None of the above approaches can be considered overly flexible or universal from the users' perspective. In addition, while the original vision of cloud computing has been about openness and interoperability, both of which can be considered dealer-impartial, in reality further complications are sometimes introduced because companies that provide services sometimes assume control over the content that has been uploaded to their cloud system. To improve the service, we propose a more flexible approach, starting from the presumption that no unnecessary uploading of content takes place. Instead, as the essential characteristic, the content is always kept in the source device, unless a backup copy is wanted or caching is needed. As will be shown in the following, this in essence makes mobile gadgets first-class cloud citizens

STRUCTUREFORAMOBILEGADGETCLOUD

Inthefollowing,wefirstlyoutlinegenericarchitecturalprincipl esinadealer-impartialandimplementation free fashion. Secondly, we describe our prototype implementation relaying commonly used open technologies.

a. Architecture&Design

Within a cloud of mobile gadgets,direct device-to-device connectivity may lead to complex, potentially cyclic interactions,which might further cause inefficient use of network and computational resources.This can be avoided byintroducing acentralized elementthatmediates betweengadgets. Reflecting theterminologyofourproof-of-concept implementation, this element is addressed as the VisualREST server (see Figure 2).Thus, the architectureillustratedinFigure2 consistsoftwokeyelements: centralserver and mobile gadgets.The interfaces betweenthesearealsoessentialtokeepuniform,open,andflexi ble.Thecentral server hasseveral keycharacteristics. The server contains themetadata of thecontent createdandresidingonthegadgets,butnotnecessarilyanyessen ceofthecontent.The server provides a search overthemetadataofthecontent.Finally,theservermanages and authenticates users, gadgets of users, and access control of the content. Additionally, the server can sometimes host the content essence itself, in addition to the content metadata, in case a mobile device is not willing to do so, or in order to provide caching to prevent overload if the same content is accessed all over again.

Themainresponsibilityofthemobilegadgetsistohosttheircon tentas aservicefor othergadgets.Thus mobile gadgetshavedualroles. Firstly, gadgetsactasservice providers: they store created content essence locally, upload the content metadata to the server, and provide the essence on request. Secondly, gadgetsactasclients foruser: theysearch forcontent metadataandaccess thecontent essenceregardlesswheretheessenceislocated.The architecture in Figure 2 can be utilized in different ways and have different scope. As one example, the central element can be a service offered to numerous users by a single (potentially global) cloud computing service provider. As another example, the central element can be a user-specific computer that manages the content of a single user's gadgets. This setting determines the scope of the mobile gadgets that belong to the same cloud, and determines how content in the gadgets accessed.

Fig. 2: Architecture for managing content in a cloud of mobile gadgets

b. Implementation & Deployment

The most obvious piece of code, VisualREST server, implements the server in Figure 2. VisualREST interacts with both client gadgets that access the content as well as with the content providing gadgets. Furthermore, additional code has been included in the VisualREST server for storing data that has been backed up or cached in the server, and for accessing certain social media content, to gain an access to all relevant content. As a result, these sites appear to the user of the system as further gadgets, whose content can be exposed through the same system.On the device side, each mobile device that hosts content needs a piece of code that is responsible for communicating with the VisualREST server and for maintaining the content that is included in each of the gadgets. The way changes in the content are noticed can be device specific; in our experiment we monitor changes in the file system. Similarly, each client device needs a piece of software for accessing the content. In the simplest case, this application can be downloaded on the fly from the VisualREST server as a web page, but in order to create more compelling applications, it is often a

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 282 - 287 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

necessity to create purpose-built clients for individual tasks, especially when using mobile gadgets. To complement the web-based system, we have implemented a middleware application running in a Nokia N900 device. The implementation of the client software is representing cloud content in terms of file systems, which means that legacy applications that can treat files can access cloud content without modifications.

The interaction between elements in the architecture relies on standard protocols following RESTful principles (Fielding and Taylor 2002). Consequently, the server can be scaled by simply introducing more computing resources as needed. HTTP is used for all the communication between the client and the server. The request mechanism for certain content from a device is based on XMPP (http://xmpp.org/). If XMPP is not available, the system will do its best with HTTP only, although this will most likely lead to delays in the delivery of the content.The implementation details of the system have already been laid out in our previous articles, and for those the reader is referred to (Ghosh et al. 2011, Mäkitalo et al. 2011). Next, we will focus on what we have learned from the design and evaluation of the system. The individual pieces of software are available for download as open source (https://github.com/hpeltola/VisualREST and http://www.soberit.hut.fi/cloudsoftware).

CONVERSATION ABOUT DESIGN DECISIONS

The most obvious lesson learned is the greater autonomy created by the ability to share without uploading data to the cloud. There is no need to commit to the gadgets from a certain dealer, to the use of a certain service, and if desired, even a personal system can be created. The system was designed using standard protocols and already existing software components, and custom software was constructed only when no suitable component existed. Furthermore, the ease of deployment was considered all the way during the design, and we believe that the system manifests the minimal yet fundamental deployment steps that are needed to remain dealer-impartial. Of course, the VisualREST server's functions could be offered as a commercial service.On the technical side, the design and implementation have given us unique insight on the design of systems building on the opportunity of mobile data as a service. Since there is no single control point in the system - the only potential control point is the VisualREST server, which is available as open source and the system can be scaled to various user-specific needs - it remains dealerimpartial, and gadgets and computers from different manufacturers can be used.

The downside of the approach is that there is apparent lack of convenience from users' perspective, because each device hosting content needs special software, which at this point must be installed. Given appropriate standardization effort, the software could be universally preinstalled in a portion of mobile phones, but this would require support from a wide industrial consortium of dealers to succeed. However, over time, we expect that web servers will find their way in mobile gadgets as well, which is a key element in the transition of mobile realm towards the open web - a key element in designing dealer-impartial systems in many other domains already today. This in turn might simplify the creation of the system with minimal installation by the user.Apart from the code in mobile gadgets hosting the content, the situation is more promising. To begin with, the VisualREST server itself can be flexibly deployed. For instance, it can be provided as a cloud-based service, as a separate hardware element readily available from a store, or as a piece of downloadable software installable to existing computer. In addition, increasingly compelling applications can be composed with web technologies only, enabling an approach where the VisualREST server would provide the different content-consuming applications as a service. Furthermore, since there still are some problematic issues for on-the-fly updates of web pages, it is also possible to create native apps that will make the content even more appealing, assuming that additional programs can be installed.

In addition to the use of content located in mobile gadgets, the VisualREST server can also be used as a gateway to services available in the web. For example, making Facebook or Flickr content - or emails from a certain account for that matter - has been experimented as a part of the system. The actual designs associated with these functions reflect their counterparts designed for mobile data, treating each source of data as a virtual device, but for practical reasons a lot of caching takes place. Similar systems have also been designed for balancing between the capabilities and capacity of cloud computing facilities and mobile gadgets. Some of them also contain additional features, such as offloading tasks to the cloud (Lagerspetz and Tarkoma, 2011).

Finally, there are some fresh approaches for managing mobile data that are related to this work. Proximiter, is a content management system based on a cloud of mobile gadgets (Xing et al. 2009). Unlike in our approach, Proximiter uses the device proximity as the driver for design. In this design, gadgets that are nearby can form an implicit mobile cloud in which it is possible to exchange content. Furthermore, operations are geared towards exchanging content in a more interactive fashion, and therefore they are in general simpler than in VisualREST. Generalizing the Proximiter approach results in the concept of cloudlets that build on infrastructure that is available locally (Satyanarayanan et al. 2009). However unlike with cloudlets, where an assumption is made that mobile gadgets' resources are always considerably inferior to fixed computers, we expect that mobile gadgets will be increasingly powerful and capable of performing hosting tasks. Consequently building on their facilities will be an increasingly feasible option.

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 282 - 287 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

CONCLUSION

While already existing facilities allow uploading data to the Web, the existing systems do not always provide the user total control over the data. The two commonly used way to manage mobile data - to synchronize everything to the web in a device specific fashion, or to insert data to a content-specific cloud- may lead to losing some of the Data to the dealer offering the service or even tie the user to a certainprovider. In this paper, we described another approach, where a dealer-impartial cloud of mobile gadgets is used to deliver mobile data as a service, resulting in a new type of a content management system. All through the design, special attention has been paid to protect the system from single-dealer lock-in. Consequently, the implementation can be best described as providing mobile data as a service, where mobile gadgets, together with the possessions they host, are treated as outstanding cloud citizens.

The resulting system is built using standard web protocols to support interoperability, which will act as protection against single-dealer lock-in. The main downside of the approach is that some pieces of software still need to be installed, when the system is deployed. Each mobile device hosting the content must provide support for accessing content in it, and since this is not in place as a standard feature, additional installations are a necessity. In addition, centralized functions, such as search, must be supported with cloud-based services. In the proposed design, these functions have been implemented in a fashion that scales from personal (or family) use in a single computer to a full-fledged cloud-based service that can serve numerous users. A key contributor to the scalability side is the use of ReSTful design, which automatically provides certain properties. Regarding the gadgets that are used to access the data, special-purpose, installable clients or plain web pages can be used. The former enables the design of top-notch user experience, whereas the latter reduces the installations to the bare minimum, because the web pages can be loaded from the central server.

Over time, as mobile gadgetsbecome more and more capable, we expect that the situation also advances in the domain ofproviding mobile data as a service. In addition, newkinds ofnetworks, providing fasterconnectivity, make it increasingly feasible toimplement systems asdescribed inthepaper.However,inordertoreallycreateasingle,dealerim partialsystemformanaging in-device mobile data, standardization is obviously needed.As many major telephony operators and mobile device manufacturers considerdealer-basedservices positively,we seethat thebest way to approach standardization is in terms of the open web,which would most likely lead to a solution that could be generalized to be universally applicable in the web.Weareexcited aboutsuchopportunity, andlook forwardtothebenefitsthiscouldimplyinthelongrun.

REFERENCES

[1.] CharlesD. Cranor,R. Ethington, Amit Sehgal, David Shur, Cormac Sreenan, and Jacobus E.vander Merwe. *Design and implementation of a distributed content management system, Proceedings o fthe13th International Workshop on Network and Operating Systems Support for Digital Audio and Video, (NOSSDAV'03)*,pages 4-11, Monterey, CA, USA, ACM,2003.

[2.] Katherine Curtis, PaulFoster and Fred Stentiford, *Meta-data-The Key to Content Management Services. In Proceedings of theMeta-Data'99,The 3rd IEEE Meta-Data Conference*, Bethesa, Maryland,USA, 1999.

[3.] RoyT.Fielding and Rickhard N.Taylor. Principled *Design of the Modern Web Architecture ACM Transactions on Internet Technology (TOIT)*,pages115–150Volume 2Issue 2,May 2002.

[4.] Subhamoy Ghosh, Juha Savolainen, Mikko Raatikainen,and Tomi Männistö.*Cloudifying User-Created Content for Existing Applications in Mobile Gadgets, In Proceedings of 2011 IEEE 35th Annual Computer Software and Applications Conference*, pp.510-515, (COMPSAC'11) IEEE Computer Society,2 011.

[5.] EemilLagerspetz and Sasu Tarkoma. *Mobile Search and the Cloud: The Benefits of Offloading*. *In Proceedings of 2011 IEEE International Conference Pervasive Computing and Communications Workshop*,pp.117-122.IEEE 2011.March21-25, 2011, Seattle,USA.

[6.] Andreas Mauthe and Peter Thomas.*Professional Content Management Systems:Handling Digital Media Assets*.JohnWiley&Sons,2004.

[7.] Niko Mäkitalo,Heikki Peltola,Joonas Salo, and Tuomas Turto.*Visual REST:A Content Management System for Cloud Computing Environment, In Proceedings of the 37th EURO MICRO Conference on Software Engineering and Advanced Applications (SEAA'11)*,pp.183-187, Aug.30 2011-Sept.2,Oulu, Finland, IEEE Computer Society,2011.

[8.] Mahadev Satyanarayanan, Paramvir Bahl, Ramon Caceres, and Nigel Davies.*The Case for VM-based Cloudlets in Mobile Computing*.pp.14-13,IEEE Pervasive Computing,Vol.8 Issue4,Oct-Dec.2009.

[9.] AntoniettaSpedalieri,AlbertSinfreu,GiuseppeSisto,PabloCesar,IshanVaishnaviandDario Melpignano.*MultimediaContentManagementSupport inNextGenerationServicePlatforms.Proceedings ofthe3rdInternationalConferenceonMobileMultimediaCommunications*,14:1-14:7,(MobiMedia'07), Nafpaktos,Greece,ICST,2007.

[10.] BoXing,Karim Seada, and Nelini Venkata subramanian.Proximiter: *Enabling Mobile Proximity-Based Content Sharing on Portable Gadgets.Pervasive Computing and Communications,2009. PerCom 2009. IEEE International Conference* ,pp.1-3, March2009

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 282 - 287 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

AUTHORS

Mr. Maddali M. V. M. Kumar received his Master of Technology in Computer Science & Engineering from Jawaharlal Nehru Technological University Kakinada. Present he is working as an Assistant Professor in Department of MCA, St. Ann's College of Engineering & Technology, Andhra Pradesh and having 1.8 Years Industrial and 6.5 Years Teaching Experience, He is a Life Member in CSI & ISTE. His research focuses on the Mobile & Cloud Computing and Network Security.

Mr. Ghanta Rajesh received his Master of Technology in Computer Science & Engineering from Jawaharlal Nehru Technological University Kakinada. Present he is working as an Assistant Professor in Department of CSE, Vizag Institute of Technology, Andhra Pradesh and having 3 Years Teaching Experience, He is a Life Member in ISTE. His research focuses on the Mobile, Cloud Computing and Information Security.