International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 01 Issue 01
February 2014

Mobile Augmentation Process and CMA Approaches

Fahmida Begum
Assistant Professor, Dr.K.V.Subba Reddy Institute of MCA, Kurnool.

Abstract:
Cloud-based Mobile Augmentation (CMA) approaches have gained remarkable ground from academia and industry. CMA is the state-of-the-art mobile augmentation model that employs resource-rich clouds to increase, enhance, and optimize computing capabilities of mobile devices aiming at execution of resource-intensive mobile applications. Augmented mobile devices envision to perform extensive computations and to store big data beyond their intrinsic capabilities with least footprint and vulnerability. Researchers utilize varied cloudbased computing resources (e.g., distant clouds and nearby mobile nodes) to meet various computing requirements of mobile users. However, employing cloud-based computing resources is not a straightforward panacea. Comprehending critical factors (e.g., current state of mobile client and remote resources) that impact on augmentation process and optimum selection of cloudbased resource types are some challenges that hinder CMA adaptability. This paper comprehensively surveys the mobile augmentation domain and presents taxonomy of CMA approaches. The objectives of this study is to highlight the effects of remote resources on the quality and reliability of augmentation processes and discuss the challenges and opportunities of employing varied cloud-based resources in augmenting mobile devices. We present augmentation definition, motivation, and taxonomy of augmentation types, including traditional and cloud-based. We critically analyze the state-of-the-art CMA approaches and classify them into four groups of distant fixed, proximate fixed, proximate mobile, and hybrid to present a taxonomy. Vital decision making and performance limitation factors that influence on the adoption of CMA approaches are introduced and an exemplary decision making flowchart for future CMA approaches are presented. Impacts of CMA approaches on mobile computing is discussed and open challenges are presented as the future research directions.

Keywords
cloud computing, authentication, mobile computing

I. INTRODUCTION

Mobile devices (e.g., smartphone, tablet pcs, etc) are increasingly becoming an essential part of human life as the most effective and convenient communication tools not bounded by time and place. Mobile users accumulate rich experience of various services from mobile applications (e.g., iPhone apps, Google apps, etc), which run on the devices and/or on remote servers via wireless networks. The rapid progress of mobile computing (MC) [1] becomes a powerful trend in the development of IT technology as well as commerce and industry fields. However, the mobile devices are facing many challenges in their resources (e.g., battery life, storage, and bandwidth) and communications (e.g., mobility and security) [2]. The limited resources significantly impede the improvement of service qualities. Smartphone's, tablets, PDAs, wearable devices are playing significant roles in human lives due to their faultless features and productivity. The easiness of user friendly interface, high definition graphics, customized application instalment, compact size, portability, multi-card SIM facility, multi-mode service operation, multi-band connectivity, high responsiveness, multimedia application support, etc features of mobile devices have fascinated the attentions of users worldwide. The demand of mobile phones is growing day by day. The survey states that in 2014, more than 1.1 billion phones will be shipped worldwide and it is expected to rise over 1.5 billion in 2017 as predicted by International Data Corporation Market Research Company [1]. In-spite of development in computer and communication technology, mobile devices can't reach their full potential for the execution of high-end applications. These include computer vision applications, augmented reality applications, face and object recognition applications, natural language processing applications, file indexing in mobile sys-tem applications, virus scanning applications, optical character recognition applications, image and video processing applications, health monitoring applications, 3D gaming applications, etc. These applications require high processing speed, large memory and battery backup to execute efficiently and accurately. However, these application requirements are not fulfilled by mobile devices. Therefore, to improve the resource scarcity of mobile phones, the computation offloading provides the best solution in mobile cloud computing.

II. MOBILE AUGMENTATION AND TYPES

A. Mobile Augmentation

Mobile augmentation in brief is the process of increasing, enhancing and optimizing computing capabilities of mobile devices by using various feasible approaches that can be hardware as well as software [7]. Mobile device is any battery-operating computing entity which is able to interact with user and execute transactions, store data and communicate using wireless technology. Augmentation approaches include hardware and software. Hardware approaches include manufacturing high-end physical components such as CPU, memory, Storage and battery. Software approaches include computation offloading, resource aware computing, remote service request, remote data storage, wireless communication and fidelity adaptation.

B. Mobile Augmentation types

There are two major types of mobile augmentation approaches.

1) Hardware: In hardware approach Smartphone's are exploited with powerful resources, multicore CPU with high clock speed, large mobile screens, long lasting battery. However augmenting hardware is hindered by several obstacles. Smartphone's handiness decrease by generating powerful processor, large storage and big screen. It also generates additional heat, increase size and weight of the Smartphone. Using long lasting battery in small mobile is not suitable with current technology [7]. The enlargement of Smartphone resources contribute to faster battery drainage and short battery life. Also all this hardware enhancement increase the cost of mobile tremendously.

2) Software: Software-oriented mobile augmentation approaches are classified into six groups, namely remote execution, remote storage, multi-tier programming, live cloud-streaming, resource-aware computing, and fidelity adaptation.

a) Remote Execution: In this resource-hungry components of mobile applications are mi-grated to the cloud. It conserves local re-source like battery, memory, and storage of mobile devices, but also enables execution of intense processing applications in Smart-phone.

b) Remote Storage: Remote storage is the process of expanding storage capability of mo-bile devices using remote storage resources. It enables maintaining applications and data outside the mobile devices and provides re-mote access to them.

c) Resource aware computing: In resource-aware computing efforts especially resource requirements of mobile applications are de-creased utilizing the application-level re-source management methods using application management software such as compiler and OS and lightweight protocols. Resource conservation is performed via efficient selection of available execution approaches and technologies. Ex. In energy constraint one should choose 2G over 3G since 2G consumes less energy.

d) Live Cloud Streaming: In live cloud streaming applications, entire processing take place in the cloud and results are streaming to the mobile devices. However, usability of cloud-streaming is hindered by latency, network bandwidth, portability, and network traffic cost.

e) Fidelity adaptation: Fidelity adaptation is an alternative solution to augment mobile de-vices in the absence of remote resources and online connectivity. In this method local resources are conserved by decreasing quality of application execution ex. YouTube. In YouTube choice of quality is asked to user.

f) Multi-tier Programming: The main idea in this type of mobile applications is to reduce the client-side computing workload and develop the applications with less native resource requirements. Computationally intensive components of the applications are executed outside the device, whereas the user interface and native codes e.g., accessing to the device camera remain inside the device for execution.

III. RELEVANCE IN CURRENT SCENARIO

Mobile phones are set to become the universal interface to online services and cloud computing applications. However, using them for this purpose today is limited

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 01 Issue 01
February 2014

to two configurations: applications either run on the phone or run on the server and are remotely accessed by the phone. These two options do not allow for a customized and flexible service interaction, limiting the possibilities for performance optimization as well. Mobile devices are resource sensitive. Some applications on mobile consume large computation power of mobile decrease its performance which can include various factors like battery, memory etc. So offloading some task from mobile to cloud server can enhance the performance of the mobile. But it is not always feasible to upload all the application on cloud. So we will be developing the cost model which will calculate the offloading cost of the application and according to the cost it will decide whether to upload or not.

IV.IMPACTS OF CMA ON MOBILE COMPUTING

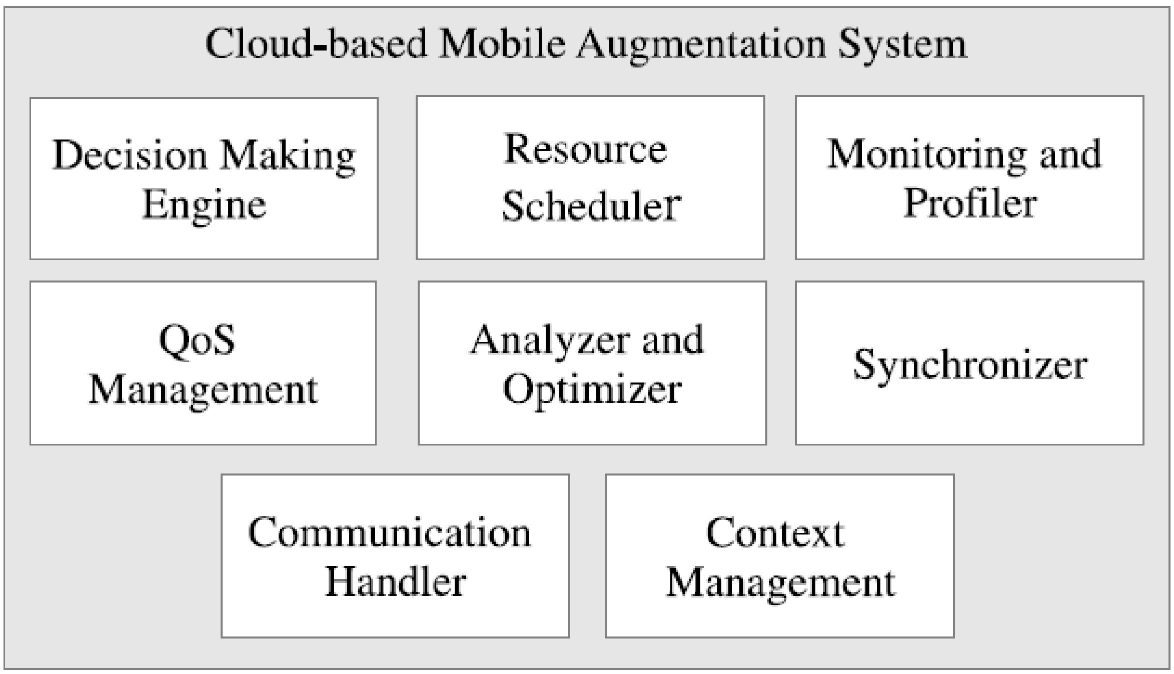

Figure 1: Major Building Blocks of an CMA System

This Section discusses the advantages and disadvantages of performing a CMA process on mobile computing that
are summarized. We aim to demonstrate how CMA approaches mitigate deficiencies of mobile computing. In this Section the terms 'cloud resources' and 'cloud infrastructures' refer to any type of cloud-based resources and infrastructures.

A. Advantages

In this part, eight major benefits of utilizing cloud resources in mobile augmentation processes are introduced.

1) Empowered Processing: Empowering processing is the state of virtually increased transaction execution per second and extended main memory leveraging CMA approaches. In computing-intensive mobile applications, the hosting device either does not have enough processing power and memory or cannot provide required energy. A common solution is to offload the application —in whole or part —to a reliable, powerful resource with least energy and time cost. In computation offloading, the complex, CPU- and memory-intensive components of a standalone application are migrated to the cloud. Consequently, the mobile devices can virtually perform and actually deliver the results of heavy transactions beyond their native capabilities. Although surrogates in traditional augmentation approaches [8]–[10], [12] could increase computing capabilities of mobile devices, excessive overhead of arbitrary service interruption and denial could shadow augmentation benefits [19].Cloud resources guarantee highest possible

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 01 Issue 01
February 2014

resource availability and reliability. Leveraging CMA approaches, application developers build mobile application with no consideration on available native resources of mobile devices and mobile users dismiss their devices' inabilities. Hence, computing- and memory-intensive mobile applications like content-based image retrieval applications (enable mobile users to retrieve an image from the database) can be executed on smartphones without excess efforts. However, a flexible and generic CMA approach that can enhance plethora of mobile devices with least configuration, processing overhead, and latency is a vital need in excessively diverse mobile computing domain. Such diversity is mainly due to the rapid development of smartphones and Tablets, and sharp rise in their hardware, platform, API, feature, and network heterogeneity in the absence of early standardization.

2) Prolonged Battery: Long-lasting battery can be considered as one the most significant achievements of CMA approaches for large number of mobile users. Smartphone manufacturers have already utilized high speed, multi-core ARM processors (e.g., Cortex-A57 Processor17) being able to perform daily computing needs of mobile end-users. However, such giant processing entities consume large energy and quickly drain the battery that irks end-users. CMA solutions can noticeably save energy by migrating heavy and energy-intensive computing to the cloud for execution. Although energy efficiency is one of the most important challenges of current CMA systems, several efforts such as are endeavoring to comprehend the energy implications of exploiting cloud-based resources from mobile devices and shrinking their energy overhead. In traditional cyber foraging or surrogate computing approaches, energy is saved by computation offloading, but several issues such as lack of mobility support and resource elasticity can neutralize the benefits of energy-hungry task offloading.

3) Expanded Storage: Infinite cloud storage accessible from smartphones enables users to utilize large number of applications and digital data on device. Hence, they are not obliged to frequently install and remove popular applications and data due to the space limit. Online connectivity is essential to access cloud storage. In such online storage systems, data are manually or automatically updated to the online storage for maintaining the consistency of the online storage system. Storing applications in cloud storage provides the opportunity to update the code without consuming any mobile I/O transactions which enhances user experience and improves the smartphones' energy efficiency —because I/O transactions are energy-hungry tasks.

4) Increased Data Safety: CMA efforts can bring the benefit of data safety to the mobile users. Naturally, stored data on mobile devices are susceptible to loss, robbery, physical damage, and device malfunction. Storing sensitive and personal data such as online banking information, online credentials, and customer related information on such a risky storage significantly degrades the quality of user experience and hinders usability of mobile devices. Due to the scarce computing resources, especially energy in mobile devices, performing complex and secure encryption provisions is not feasible. Hence, by storing data in a reliable cloud storage, users ensure data availability and safety regardless of time, place, and unforeseen mishaps. Threats such as device robbery or physical damage to the mobile devices will effect on the tangible value of the device rather than intangible value of the data.

5) Ubiquitous Data Access and Content Sharing: Cloud infrastructures play a vital role in enhancing data access quality. Storing data in cloud resources enables mobile users to access their digital contents anytime, anywhere, from any device. Hence, the impact of temporal, geographical, and physical differences is noticeably decreased that enriches user experience. Moreover, cloud storages facilitate data sharing and contribution among authorized users. Every file and folder in cloud, usually has a protected unique access link that can be obtained by the owner to share them among legitimate users. Network traffic is hence, shrunk because data is accumulated in a central server accessible to unlimited users from various machines. Cloud can significantly enhance data transfer among different mobile devices. One of the most irksome user's impediments is to transfer data from current mobile device to a new handset. Apart from its temporal cost, porting data from one device to another, especially to a heterogeneous device is a risky practice that puts data is in the risk of corruption and loss of integrity. Stored data on Cloud remain safe and can be synchronized to any number of mobile devices with minimum risk. However, a reliable data access control mechanism is required to adjust user permissions.

6) Protected Offloaded Content: Cloud storage solutions aim to protect remote codes and data while ensure user's privacy. This is one of the most important gains of replacing surrogates with cloud resources. Cloud servers deploy virtualization technology to isolate the guest environment from other guests and also from their permanent software stack. Moreover, cloud vendors deploy strict security and privacy policies to not only ensure confidentiality of user content, but also to protect their properties and business. Implement internal security provisions particularly the stateof-the-art biometric security systems to protect their physical infrastructures and avoid unauthorized access. Employing complex content encryption, frequent patching, and continuous virus signature update inside the company premise or seeking technical services from a trusted third party are other examples of security provisions undertaken in cloud to further protect cloud storage.

7) Enriched User Interface: As described in part II-B4, visualization shortcomings of mobile devices diminish user experience and hinder smartphones' usability. However, cloud resources can be exploited to perform intensive 2D or 3D screen rendering. The final screen image can be prepared based on the smartphone screen size and streamed to the device. Consequently, screen adaptation also is achieved when cloud side processing engine automatically alter the presentation technique to match screen image with the device screen size.

8) Enhanced Application Generation: Cloud resources and cloud-based application development frameworks similar to µCloud and CMH, facilitate application generations in heterogeneous mobile environment. Once a cloud component is built, it can be utilized to develop various distributed mobile applications for large number of dissimilar mobile devices. In the presence of cloud components, application programmer needs to develop native mobile components and integrate them with relevant, prefabricated cloud components to develop a complex application. When a mobile-cloud application is developed for Android device, by slightly changing native components the application is transited to new OS like iOS18 and Symbian19 which significantly save time and money.

B. Disadvantages

Despite of many advantageous aspects of cloud services, their success is hindered by several drawbacks and shortcomings that are discussed as follows.

1) Dependency to High Performance Networking Infrastructure: CMA approaches demand converged wired and wireless networking infrastructures and technologies to fulfill intersystem communication requirements. In wireless domain, CMAs need high performance, robust, reliable, high bandwidth wireless communication to realize the vision of computing anywhere, anytime, from any-device. In wired communication, fast reliable communications ground is essential to facilitate live migration of heavy data and computations to a regional cloud-based resources near the mobile users. Efforts such as next generation wireless networks and the open mobile infrastructure with Open Wireless Architecture (OWA) by Sieneon [127] contribute toward enhancing the networking infrastructures' performance in MCC.

2) Excessive Communication Overhead and Traffic: Mobile data traffic is significantly growing by ever-increasing mobile user demands for exploiting cloud-based computational resources. Data storage/retrieval, application offloading, and live VM migration are example of CMA operations that drastically increase traffic leading to excessive congestion and packet loss. Thus, managing such overwhelming traffic and congestion via wireless medium becomes challenging, especially when offloading mobile data are distributed among helping nodes to commute to/from the cloud. Consequently, application functionality and performance decrease leading to user experience degradation.

3) Unauthorized Access to Offloaded Data: Since cloud clients have no control over their remote data, users contents are in risk of being accessed and altered by unauthorized parties. Migrating sensitive codes as well as financial and enterprise data to publicly accessible cloud resources decreases users privacy, especially enterprise users. Moreover, storing business data in the cloud is likely increasing the chance of leakage to the competitor firm. Hence, users, especially enterprise users hesitate to leverage cloud services to augment their smartphones.

4) Application Development Complexity: The excessive complexity created by the heterogeneous cloud environment increases environment's dynamism and complicates mobile application development. Mobile application developers are required to acquire extensive knowledge of cloud platforms (i.e., cloud OSs, programming languages, and data structures) to integrate cloud infrastructures to the plethora of mobile devices. Understanding and alleviating such complexity impose temporal and financial costs on application developers and decrease success of CMA-based mobile applications.

5) Paid Infrastructures: Unlike the free surrogate resources, utilizing cloud infrastructure levies financial charges to the end-users. Mobile users pay for consumed infrastructures according to the SLA negotiated with cloud vendor. In certain scenarios, users prefer local execution or application termination because of monetary cloud infrastructures cost. However, user payment is an incentive for cloud vendors to maintain their services and deliver reliable, robust, and secure services to the mobile users. In addition, cloud vendors often charge mobile users twice; once for offloading contents to the cloud and once again when users decide to transfer their cloud data to another cloud vendors to utilize more appropriate service (e.g., monetary and QoS (Quality of Service) aspects).

6) Inconsistent Cloud Policies and Restrictions: One of the challenges in utilizing cloud resources for augmenting mobile devices is the possibility of changes in policies and restrictions imposed by the cloud vendors. Cloud service providers apply certain policies to restrain service quality to a desired level by applying specific limitations via their intermediate applications like Google App Engine bulk loader20. Services are controlled and balanced while accurate bills will be provided based on utilized resources. Also, service provisioning, controlling, balancing, and billing are often matched with the requirements of desktop clients rather than mobile users. Considering the great differences in wired and wireless communications, disregarding mobility and resource limitations of mobile users in design and maintenance of cloud can significantly impact on feasibility of CMA approaches. Hence, it is essential to amend restriction rules and policies to meet MCC users requirements and realize intense mobile computing on the go.

7) Service Negotiation and Control: While cloud users are required to negotiate and comply with the cloud terms and conditions for a certain period of time, often cloud agreements are nonnegotiable and policies might change over the time. Moreover, there is no control over the cloud performance and commitments in the absence of a controlling authority or a trusted third party. Hence, CMA services are always volatile to the service quality of cloud vendors.

	Distant immobile cloud	Proximate immobile computing entities	Proximate mobile computing entities	Hybrid
Architecture	Distributed			
Ownership	Service provider	Public	Individual	Hybrid
Environment	Vendor Premise	Business Center	Urban Area	Hybrid
Availability	High	Medium	Medium	High
Scalability	High	Medium	Medium	High
Sensing Capabilities	Medium	Low	High	High
Utilization Cost	Pay-As-You-Use			
Computing Heterogeneity	High	Medium	High	High
Computing Flexibility	High	Medium	High	High
Power Efficiency	High	Medium	Medium	High
Execution Performance	High	Medium	Medium	High
Security and Trust	High	Moderate	Low	High
Utilization Rate	High			
Execution Platform	VM	VM	Physical/VM	Physical/VM
Resource Intensity	High	Moderate	Moderate	Rich
Complexity	Low	Moderate	Moderate	High
Communication Technology	3G/Wi-Fi	Wi-Fi	Wi-Fi	3G/Wi-Fi
Mobility	NA	NA	High	High
Communication Latency	High	Low	Low	Moderate
Execution Latency	Low	Medium	Medium	Low
Maintenance Complexity	Low	Medium	Medium	High

Table 1: Comparison of Cloud based server

V. CMA Approaches

In this section, the architecture of the proposed CMA framework is introduced. For this purpose, a general technology-independent architecture is defined first, which can be applied to arbitrary technologies. To further develop this general architecture towards a concrete implementation, a technology-specific architecture is derived afterwards. Derivation of the technology-specific architecture is based on an elaborate selection of currently available technologies

A.General Architecture

This architecture comprises a central repository that stores complete application packages. Each application package contains all necessary files and dependencies required to run a specific application. The central repository is shared among all potential external resources. It is up to the resource provider to only support a certain subset of applications. To enable the referencing of application packages at run-time, a couple of identifiers are used. At compile time, each application part considered for offloading is assigned with an identifier that is unique in the context of the application package. In addition, each package is assigned with a globally unique identifier as well.

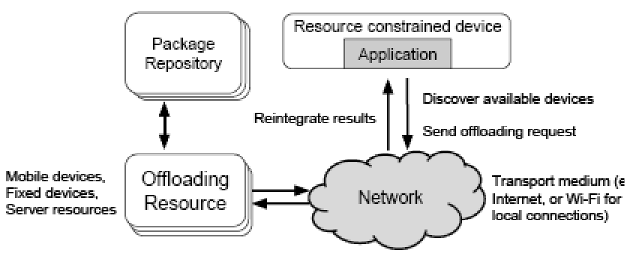

Figure 2: Highlevel Architecture

During run-time, a resource-constrained device maintains a list of available offloading resources. If the CMA framework decides that a particular part of an application needs to be offloaded, a connection to a suitable offloading resource is established and the offloading process is initiated. Once an offloading resource receives a request, it downloads the complete package from the package repository using the package ID, identifies the part to execute, executes it, and sends back the results to the resource-constrained device, where the results are finally reintegrated into the application. We are aiming at a lightweight approach operating on the source level of the application. This architecture basically enables two approaches. Either the application developer provides implementations in different programming languages for various platforms, or a single implementation utilizing cross-platform frameworks or common programming languages.

B. Technological Considerations

The proposed general architecture is completely technology-independent. To develop this architecture towards a concrete solution, an appropriate technology must be selected to further enhance the architecture's various building blocks. When selecting an appropriate technology, the heterogeneous landscape of current mobile platforms and programming environments must be taken into account. Concretely, to minimize the development effort a single
programming language with support for all platforms is desirable. Currently the only language that is supported by
all platforms and that can equally be executed on all mobile operating systems is JavaScript. We are totally aware that JavaScript is not a panacea and introduces other problems, but it forms a common basis. JavaScript is the common anchor and technological basis for a new class of programming languages. The compilers of these programming languages produce highly optimized JavaScript code to be executed in the browser. Representative examples are Coffeescript [7] or TypeScript [8]. The Dart [9] programming language, which is heavily supported by Google, even goes one step further. It does not only define a new syntax, which can be compiled to JavaScript, but provides a complete Dart ecosystem. The Dart ecosystem is based on a Dart VM executing native Dart code. In addition Dart applications can be compiled to JavaScript files to be run in the browser or in standalone JavaScript environments. Dart thereby hides the complexity and still remaining incompatibilities between the different JavaScript engines from the developer. The Dart ecosystem further offers access to public and private package repositories, which are fully integrated into the build process. The provided overview shows that there are currently various programming languages available that base on JavaScript and hence provide cross-platform applicability. From the surveyed languages, Dart appears to be best evolved and most promising approach to meet requirements of CMA-based applications. Furthermore it eases the development of crossplatform applications by hiding peculiarities. At first sight Dart seems to be a very tailored and narrowed choice. Currently the landscape of available native Dart applications is very limited, but Dart features a high grade of compatibility with JavaScript in just a few lines of code. From a JavaScript enabled browser's perspective, there is no difference if an application uses HTML5 and JavaScript technology, or if it uses HTML5 and Dart technology. Furthermore, there is currently active development in progress to create a framework called Sky1 enabling direct mobile app development in the Dart programming language. Currently the project primarily targets the Android platform but may be extended to also support other platforms. Web-based applications are no longer limited to the web. In fact, web-based applications

are emerging even on classical and mobile end-user devices. For instance, Firefox OS [10] natively supports HTML5 with JavaScript applications. Furthermore, Windows Phone also natively support apps based on HTML5 and JavaScript. For all other platforms, crossplatform solutions like Adobe PhoneGap [11] or Appcelerator
Titanium [12] exist. They wrap web-based applications into native packages and display content by using web-view
technology. Of course, there are also disadvantages of using web-based technologies on mobile devices. Native applications get the look and feel of the respective platform out of the box, whereas web-based solutions require additional effort or libraries to mimic a platform-specific look and feel. Furthermore, applications utilizing web technologies still do not achieve the performance of native applications, even though the performance of web-based applications is continuously improved. The most significant advantage of web-based technologies is the fact that the hurdle of maintaining multiple code bases for different platforms is removed, which reduces maintenance effort. In the context of CMA, we see this as a great chance to build a system that is available on all major platforms, is not locked to a single system, and does not need major adaptations once new operating-system versions are released.

To fully employ the potentials of web-based technologies, we have chosen Dart as our fundamental programming language for our proposed CMA framework. In the following subsection we derive a technology-specific architecture that is tailored to the Dart ecosystem.

C. Technology-Specific Architecture

The resulting refined technology-specific architecture as shown in Figure 2 takes into account the decision to rely
on Dart and applies refinements on the general architecture. According to the architecture, it is assumed that the communication with proximity resources takes place using lowlatency Wi-Fi connections. High-latency 3G/4G connections are used, if no suitable proximity resources are available. Wi- Fi connections can also be used to connect to resources that are not in the proximity of the user. In this case, potentially untrusted resources are used as gateways.

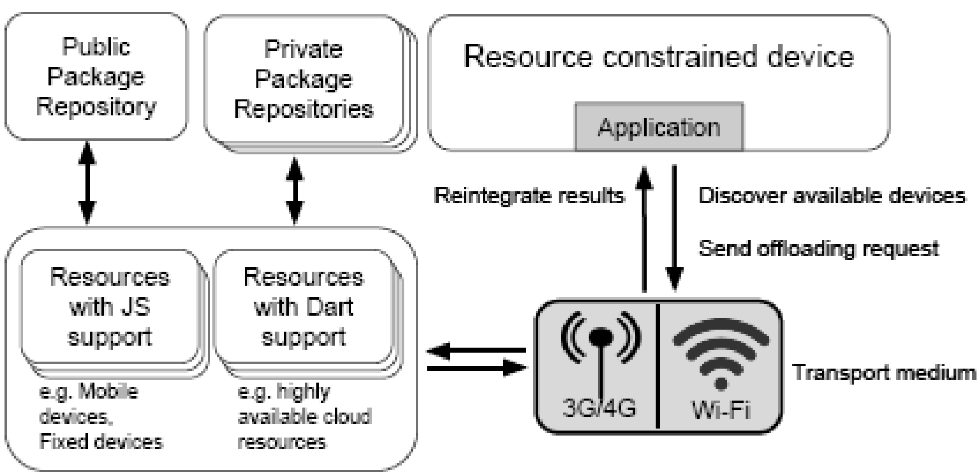

Figure 3: Technology Specific Architecture

VI. CONCLUSIONS AND FUTURE WORK

In this paper we presented POWER, a CMA framework based on a novel architecture. In contrast to existing CMA solutions, the proposed framework provides interoperability and cross-platform applicability by relying on web technologies. Development of the proposed framework's architecture has been based on a set of relevant requirements. A conducted evaluation against these requirements has revealed that the proposed framework meets most requirements already on architectural level. The feasibility of the proposed framework has been evaluated by means of a concrete implementation. The implementation already demonstrates the proposed framework's capabilities to boost the performance of resourceintensive processes on mobile end-user devices and even for web-based applications. We will focus our future work on further improving the framework in terms of resource discovery and end-to-end security. Furthermore, adherence to our defined requirements is our top priority.

VII. REFERENCES

[1] M. Satyanarayanan, "Pervasice Computing: Vision and Challenges," IEEE Personal Communications, pp. 10–17, 2001.
[2] A. Reiter and T. Zefferer, "Paving the Way for Security in Cloud-Based Mobile Augmentation Systems," in 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (IEEE Mobile Cloud 2015) (Note: to appear), 2015.
[3] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Stefan, R. Chandra, and B. Paramvir, "MAUI : Making Smartphones Last
Longer with Code Offload," in Proceedings of the 8th international conference on Mobile systems, applications, and services, vol. 17, 2010,
pp. 49–62.
[4] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: Elastic Execution Between Mobile Device and Cloud," in Proceedings
of the sixth conference on Computer systems, 2011, pp. 301–314.
[5] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "ThinkAir: Dynamic resource allocation and parallel execution in the cloud for
mobile code offloading," in 2012 Proceedings IEEE INFOCOM. Ieee, Mar. 2012, pp. 945–953.
[2] M. Satyanarayanan, "Mobile computing: where's the tofu?" *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 1,
no. 1, pp. 17–21, 1997.
[3] S. Abolfazli, Z. Sanaei, and A. Gani, "Mobile Cloud Computing: A Review on Smartphone Augmentation Approaches," in *Proc. WSEAS*
CISCO '12, Singapore, 2012.
[4] M. Othman and S. Hailes, "Power conservation strategy for mobile computers using load sharing," *ACM SIGMOBILE Mobile Computing*
and Communications Review, vol. 2, no. 1, pp. 44–51, Jan. 1998.
[5] A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning, "Saving portable computer battery power through remote process execution,"
ACM SIGMOBILE Mobile Computing and Communications Review, vol. 2, no. 1, pp. 19–26, 1998.
[6] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges," *IEEE Pers. Commun.*, vol. 8, no. August, pp. 10–17, 2001.
[7] J. Flinn, D. Narayanan, and M. Satyanarayanan, "Self-tuned remote execution for pervasive computing," in *Proc HotOS'01*, Kr¨un, Germany,
2001, pp. 61–66.
[8] J. Flinn, P. SoYoung, and M. Satyanarayanan, "Balancing performance, energy, and quality in pervasive computing," in *Proc. IEEE ICDCS '02*, Vienna, Austria, 2002, pp. 217–226.
[9] R. K. Balan, "Simplifying cyber foraging," Doctorate Thesis, Carnegie Mellon University, 2006.
[10] H.-I. Balan, R and Flinn, J and Satyanarayanan, M and S.Sinnamohideen and Yang, "The Case for Cyber Foraging," in *Proc.*
ACM SIGOPS EW10, Saint Malo, France, 2002, pp. 87–92.
[11] R. K. Balan, D. Gergle, M. Satyanarayanan, and J. Herbsleb, "Simplifying cyber foraging for mobile devices," in *Proc. ACM MobiSys'07*,
Puerto Rico, 2007, pp. 272–285.
[12] R. K. Balan, M. Satyanarayanan, S. Park, and T. Okoshi, "Tactics-based remote execution for mobile computing." in *Proc. ACM MobiSys'03*, San Francisco, California, USA, 2003, pp. 273–286.
[13] S. Goyal and J. Carter, "A lightweight secure cyber foraging infrastructure for resource-constrained devices," in *Proc. MCSA'04*, Lake
District National Park, UK, 2004, pp. 186–195.
[14] M. D. Kristensen, "Enabling cyber foraging for mobile devices," in *Proc. MiNEMA'7*, Magdeburg, Germany, 2007, pp. 32–36.
[15] M. D. Kristensen and N. O. Bouvin, "Developing cyber foraging applications for portable devices," in *Proc. 7th IEEE Int'l Conf.*
Polymers and Adhesives in Microelectronics and Photonics, and 2nd IEEE Int' Interdisciplinary Conf. PORTABLE-POLYTRONIC, 2008,
pp. 1–6.