

TOWARDS ESTABLISHING A CATALOGUE OF PATTERNS FOR ARCHITECTING MOBILE CLOUD SOFTWARE

Aakash Ahmad, Ahmed B. Altamimi, Abdulrahman Alreshidi

College of Computer Science and Engineering,
University of Ha'il, Ha'il, Saudi Arabia

ABSTRACT

Mobile computing empowers its users to exploit portable computation and context-aware communication, however; a mobile device lacks energy and performance to execute computation and memory intensive tasks. On the contrary, cloud computing exploits the 'pay-per-use' software and hardware services to provide virtually unlimited processing and storage resources. The unification of mobile and cloud computing as Mobile Cloud Computing (MCC) enables mobility and context awareness with computation and storage services to provide systems that are portable, yet resource sufficient. In an architectural context for MCC systems that require context-awareness, mobility and scalability, etc., there is a need to capitalise on reusable solutions – utilising patterns and best practices – to architect and develop mobile cloud software. This research aims to build and exploit a catalogue of patterns that support reusable design knowledge for architecture-based development of the MCC systems. We have discovered some patterns as generic and reusable solutions and demonstrate their usage in the context of mobile cloud systems. The proposed research aims to establish the catalogue as patterns repository – facilitating a continuous discovery and documentation of new patterns overtime that support reusable knowledge and practices to develop MCC systems.

KEYWORDS

Patterns and Frameworks, Software Architecture, Software Reuse, Cloud Computing, Mobile Cloud Systems.

1. INTRODUCTION

Mobile computing has emerged as a pervasive technology by exploiting anywhere, anytime – portable, context-sensitive and connected – mobile devices [1]. Such devices with embedded sensors (hardware) and freely available mobile apps (software) have empowered users to perform a variety of tasks ranging from mobile commerce to health and fitness monitoring [2]. However, a mobile device is considered as a resource-constrained computer that lacks the energy, efficiency and quality of service for computation and memory-intensive tasks [3, 4]. Cloud computing model have proven to be a success for provisioning/de-provisioning of the 'pay-per-use', on-demand and virtually unlimited hardware and software resources [5]. The unification of mobile and cloud as Mobile Cloud Computing (MCC) represents the state-of-the-art mobile computing technology that aims to minimize the resource poverty of mobile devices by exploiting the

resource sufficient cloud-based software and hardware services [6, 7]. Despite the benefits of MCC technology, a number of challenges must be addressed while architecting and developing MCC systems [1, 2, 6]. Moreover, a rapid demand for developing mobile cloud based software requires a number of highly knowledgeable and experienced architects who may not be widely available as MCC has recently emerged as an innovative technology [2]. Architectural styles and patterns have been used for providing packaged knowledge about well-known design solutions to both experienced and novice architects [8, 9]. Specifically, patterns embody proven solutions to recurring problems by capturing concentrated wisdom of many practitioners and consolidated design knowledge from multiple systems [10]. In recent years, pattern-based approaches resulted in (i) promoting reuse while (iii) decreasing the efforts required during architectural design and evolution processes [8, 11]. In addition, pattern-oriented solutions [10] enhance quality by applying the best practices and knowledge to resolve recurring problems of architectural design [4].

In this research, we aim to support pattern-based architecting for MCC software that can accelerate the process of gaining knowledge and experience in successfully modeling and evolving the system's structure and behavior at higher abstractions [3, 7]. Specifically, we focus on building and exploiting the catalogue as a collection of architectural patterns that promote reuse of design knowledge for architecting MCC systems that is currently lacking in existing research [3, 4, 12]. We model and utilise an MCC architectural pattern as a generic and repeatable solution to recurring problems of architectural design. While architecting MCC systems, one exploits dynamically composed services to develop systems that are portable, context-sensitive and efficient [4, 5, 7]. In comparison to the more traditional (object-oriented, component-based and service-driven) systems [8, 9], patterns for mobile MCC architectures are characterized by specific requirements such as mobility, context-sensitivity for (front-end) mobile computing with service composition, and scalability of (back-end) cloud services.

One of the key challenges in providing pattern-based architectural knowledge is systematic discovery and detailed documentation of patterns as reusable packages of known solutions to recurring problems [8, 10]. In this paper, we report (i) our effort towards establishing a catalogue of architectural patterns for MCC applications; and (ii) demonstrate how the discovered architectural patterns can be applied to architect a mobile cloud system. Pattern discovery is a continuous process that requires frequent mining of pattern sources to discover and document new patterns that emerge over-time [10, 18]. Our approach consisted of three simple steps including: pattern discovery, pattern documentation, and pattern application. With regards to the existing research in [4, 5, 6, 7], our research aims to contribute:

- Systematic discovery of patterns that address architectural aspects of mobile cloud software. The discovered patterns provide a foundation to establish a pattern catalogue as a repository of reusable solution and best practices to design and architect the MCC systems.
- Exploit the pattern catalogue with discovered patterns as elements of reuse knowledge that guides a step-wise process of pattern-driven and reuse-oriented architecting of MCC systems.

This research reports our preliminary efforts to discover and document patterns. The futuristic research aims to support a (semi-) automated discovery of patterns with user's decision support in

the pattern discovery process. In addition to the discovery of new patterns, case study based evaluation are planned to evaluate the applicability of pattern-based architecting.

The rest of this paper is organized as follows. Section 2 presents the related research. Section 3 discusses the research methodology and proposed solution. Section 4 highlights the reference architecture for mobile-cloud systems. Section 5 presents the discovered patterns and their documentation. Section 6 discusses some threats to validity and futuristic dimensions of the research. Section 7 concludes the paper.

2. RELATED RESEARCH

In this section, first we discuss the existing research on (i) patterns for mobile and cloud computing architectures, and then highlight some (ii) reference architecture and patterns for mobile cloud based systems. The discussion of the related research here highlights the research state-of-the-art and justifies the needs and scope of the proposed work.

2.1. Patterns¹ for Mobile Cloud Computing Architectures

Cloud Computing – One of the thorough work on cloud architecture patterns [13] reports best practices for scalability, big data, fault handling and distributed services on Windows Azure (Platform as a Service: PaaS). Patterns in [13] provide guidelines and practical solutions to address the scalability and elasticity in cloud-native applications for Windows Azure platform. Also, the work reported in [10] provides a collection of patterns for development of cloud service models (SaaS, PaaS, IaaS), and their deployment using cloud deployment models (private, public, hybrid, community). The patterns are organized in a framework to guide developers to systematically select and apply these patterns. The work in [15, 16] reports a community-driven development of patterns. Specifically, the *Cloud Computing Design Patterns* in [15] represent a collection of 39 patterns to address the scalability, reliability, security and monitoring issues of cloud applications. Also, the *Cloud Design Patterns* [16] present a collection of patterns created by various (cloud) architects based on the type of problems, and their generic design patterns as repeatable solutions.

Mobile Computing – Compared to the research on cloud computing patterns, there is less work on pattern-based designing and architecting of mobile computing solutions. In [4], three architectural patterns are presented for mobile computing in the context of tactical-edge resource-constrained environments that support first responders and military personnel operating in edge environments. The proposed architecture patterns namely *data source integration*, *group-context awareness*, and *cyber foraging* are driven by flexibility, resource efficiency, and usability, which are key quality attributes for systems at the tactical edge. The proposed patterns enable system architects to instantiate them using a variety of technologies that can meet functional and quality requirements while developing and operating mobile systems. In a similar work [17], a collection of architectural solutions namely mobility patterns are presented. The proposed mobility patterns

¹The terms patterns and styles are often used interchangeably [10]. We focus on patterns that ‘provide generic, reusable solution to recurring problems of architectural design’ [8], while ‘styles provide a constrained composition of elements for architectural organization and restructuring’ [9].

have been derived from successful mobile applications and allow a designer to reuse design elements as building blocks for mobile applications.

In contrast to the existing research [10, 13, 4, 17], the proposed patterns in our catalogue are focused on architecture-centric development, execution, and management aspects of MCC systems. Moreover, our work aims to establish a pattern catalogue - continuously evolving pattern repository - based on an incremental discovery and specification of new architectural patterns guided by [8, 14, 18].

2.2. Reference Architectures and Patterns for MCC Systems

A number of reference architecture [3, 19] and a pattern-based solution for mobile service oriented architecture (SOA) [12] exist. The technical distinction between a reference architecture and architectural patterns are detailed in [20]. In [3], the authors have described a reference architecture that enables off-loading of (computation and memory-intensive) mobile code to cloud-based servers in the context of hostile environments. This paper also presents viable implementations scenarios along with architectural tradeoffs. A similar work [19] presents a highly-extensible reference architecture that enables group-context-aware mobile applications that integrates contextual information from individuals with that of nearby team members to enable context-driven team coordination. In the context of mobile SOA, a taxonomy of six architectural patterns namely; *standalone*, *full offloading*, *partial offloading*, *SaaS-based*, *CaaS-based*, and *offloaded CaaS-based* is presented in [12]. For each of the proposed architectural pattern, the authors have organized architectural patterns with key components and their interactions at runtime. Different architectural patterns exhibit various levels of qualities including the performance, efficiency and energy consumption for mobile applications.

In the sections above, we have presented an overview of the most relevant research to our proposed work. Specifically, the work [3, 12, 19] on reference architecture and patterns [20] is most relevant to our proposed solution. Based on an overview of the existing research and the current challenges for mobile computing [1, 2, 6], we claim that the proposed solution is a preliminary attempt to establish a catalogue that aims to support architecture-centric (reusable) development of MCC systems.

3. RESEARCH METHODOLOGY AND PROPOSED SOLUTION

In this section, first we discuss the methodology for discovering and documenting the patterns (Section 3.1). We then provide an overview of the solution for pattern-based process for architectural design (Section 3.2).

3.1. Methodology for Pattern Discovery and Documentation

Pattern discovery is based on design review method [21] by reviewing recurring design solutions to frequent problems of architecting MCC systems [6, 7]. The design review team was comprised of 3 members with experience of (a) conducting the systematic review [1, 15], (b) pattern mining [18], and (c) development of mobile and cloud systems. Our approach to discover and document the patterns consisted of the following steps.

Step I – Conducting Systematic Review on Architectural Solutions for MCC Systems: The review was conducted to investigate the recurring challenges, design problems and existing solutions to develop mobile cloud architectures [24]. A systematic review [22] is expected to minimise the potential bias in the review and has a protocol that guides the process. Based on the research questions (RQs) below and the review protocol, we selected 85 studies (problem-solution map) as primary sources of pattern discovery.

RQ1 – What methods/techniques/frameworks/solutions are provided in existing (research and practices) to model/develop/evolve MCC system architectures?

RQ2 – What are the patterns/styles/frameworks to support reusable design knowledge for architecting MCC systems?

Step II – Identification of Pattern Data Sets: Once the studies were identified, we extracted the data sets in Table 1 from selected studies. Datasets refers to mapping the existing architectural design problems and their solutions. For an objective evaluation, we derived 7 items in Table 1 (I1 to I7 – item collection is referred to as datasets) by following the guidelines in [21] and based on our experience with architecture pattern mining [18] and classification of reusable architectural solutions and patterns [23]. Items in Table 1 guided the pattern mining team to objectively review the problem (P) and solution (S) mapping, the attributes (A) that affect the solution and the occurrence frequency (T) of the repeatable solution by analyzing the pattern datasets. Once a decision (D) is reached, the results are documented as pattern elements (E) for a peer-review before finalization.

Step III - Thematic Analysis to Investigate Pattern Datasets: After identification of the datasets, thematic analysis as the final step helps to ‘identify, analyse and report’ patterns from datasets by following three steps. A theme is a possible solution, or method or a mechanism to resolve problems.

(1) **Data Analysis** process comprises of (a) analysing datasets, (b) to extract design attributes from problem-solution mapping (I5 in Table 1).

(2) **Pattern Discovery** process involves (a) searching of the recurring themes based on data analysis, and (b) reviewing the identified themes. To discover patterns, we reviewed studies and aimed at discovering design problems (I2) and their relate solutions (I3). We consider a recurring theme as a discovered pattern (I6).

(3) **Pattern Documentation** is the last process that includes (a) classification of related themes based on design attributes (I5) and documented them in a template (I7).

3.2. Solution Overview for Pattern-based Architecting

We propose pattern-driven architecting as a 3-step process with underlying activities and repositories in Figure 1. Pattern discovery involves *pattern discovery* and *pattern documentation* (in Section 5). Pattern documentation involves pattern classification (in Section 6). Pattern application involves *selection* and *instantiation* (in Section 7). If a designer finds suitable patterns from the catalogue then the first two steps are skipped

Table 1. Dataset Items for Pattern Discovery Process

ID	Items	Description
I1	Design Space (C)	All the available architectural design (C = 85 studies).
I2	Recurring Problem (P)	Repeatable problems existing in the design space ($P \in C$).
I3	Frequent Solution (S)	Solutions to repeatable problems in the design space ($S \in C$).
I4	Frequency Threshold (T)	Threshold for occurrence of S to be discovered as a pattern
I5	Design Attributes (A)	Attributes affecting S (A = 08) <i>Context Awareness, Mobility, Computational Efficiency, Energy Efficiency, Service Reliability, Service Availability, Data Storage, Data Processing</i>
I6	Discovered Pattern (N)	2 = Yes, 1 = Not sure (consensus required), 0 = No
I7	Pattern Elements (E)	Elements of Pattern Description (E = 9) <i>Name, Intent, Problem (P), Solution (S), Impact, Origin, Uses, Reference Diagram, Architecture Elements, Constraints</i>

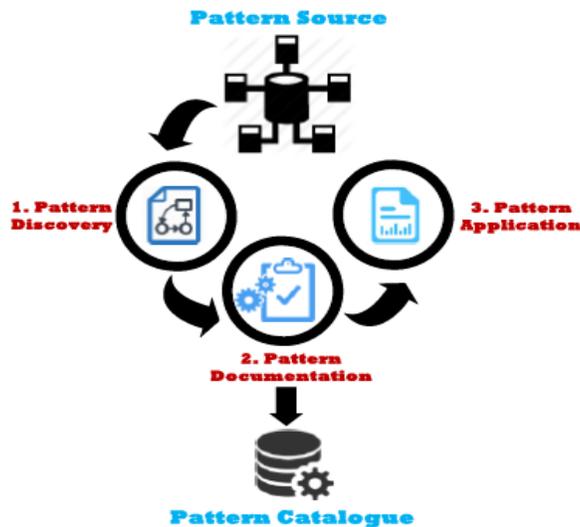


Figure 1. A Three Step Processes for Pattern-driven Architecting

4. REFERENCE ARCHITECTURE AND CHARACTERISTICS FOR MCC SYSTEMS

We now present the reference architecture for the MCC systems (Section 4.1) and also discuss the characteristic of the MCC architectures (Section 4.2)

4.1 A Reference Architecture for MCC Systems

In Figure 2 a) we illustrate the reference architecture for the mobile cloud systems that consists two distinct layers of architectural component, (i) Cloud Computing Layer (resource sufficient - back-end), and (ii) Mobile Computing Layer (portable and context-aware - front-end). The

discussion and presentation of the reference architecture is vital before presenting the architectural patterns and their presentation [20]. Specifically, the reference architecture in Figure 2 a) acts as a blueprint or simply a reference to derive advanced architectural solutions. Further details about the reference architectures, architectural solutions and architectural patterns are provided in [3, 4, 20].

(1) *Portable and Context-Aware and User Interface Layer* acts as a front-end that allows a mobile user to manipulate the data by exploiting the context and location information of the mobile device. However, the front-end mobile device lacks computation and storage-intensive resources.

(2) *Resource Sufficient and Elastic Cloud Layer* acts as a back-end to the off-loaded data by mobile device that allows scalable and virtually unlimited storage and processing resources [2, 7]. As in Figure 2, the mobile-cloud computing can empower its users by unifying the features such as context-sensitivity, location-awareness and mobility at the mobile computing layer, with virtually unlimited computation and storage resources of cloud computing. However, such system integration and operation requires a continuous network connectivity and involves latency along with security and privacy of data that communicates between mobile and cloud.

The intra-layer and inter-layer communication among the components is enabled by means of architectural connectors [10, 23]. For example, Figure 2 b) that represents a simplified (component and connector-based) architectural view of a system that invokes location-based services. In the component and connector based architectures the components acts as computational elements or data stores that are connected to each other using the connectors. Specifically, the *getLocationService* component (on a mobile device) requests for a location based service that is computed and provided by *sendLocationService* component on the (cloud server). Both these components are interconnected using *LocSrv* connector.

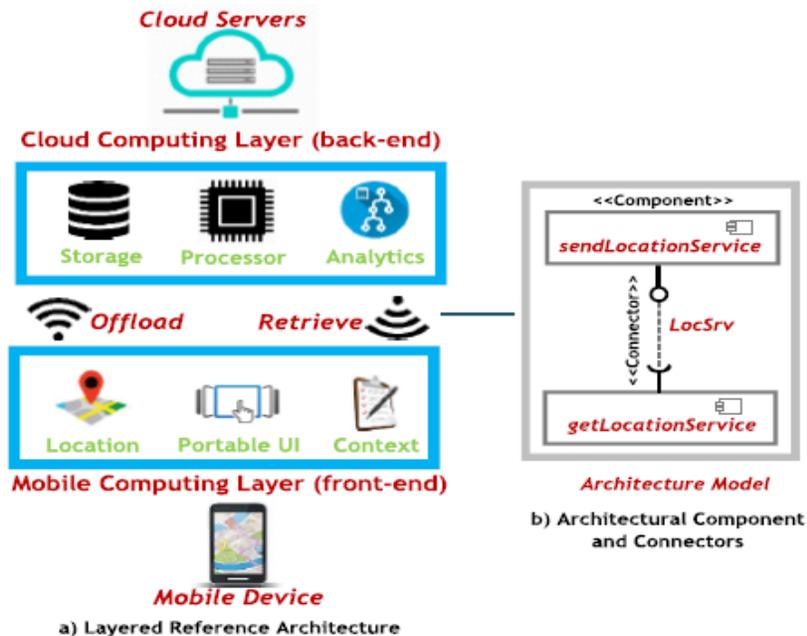


Figure 2. Reference Architecture for Mobile Cloud Systems.

4.2. Characteristics of Mobile Cloud Architectures

In the mobile cloud architectures, in addition to supporting the core functionality the characteristics of software quality are also vital for both the mobile computing and cloud computing. Specifically, for the mobile computing aspects of the MCC systems the main characteristics of the quality refer to the *context-awareness, mobility, efficiency and security of the mobile data* [1, 2]. In contrast to mobile, the cloud based systems exploit the principle of service-orientation that enables service composition as a foundation to develop cloud-based applications [15, 22]. In addition, central to cloud architectures are the quality of service (QoS) requirements that ensures composable services must satisfy the desired quality characteristics. These characteristics include but not limited to scalability, elasticity, multi-tenancy and virtualization of software services that distinguish the mobile cloud architectures [4, 7] from the traditional software architectures [9, 10]. Therefore, the patterns for traditional software development and architecture [9] cannot easily be applied to mobile cloud systems unless they support the above-mentioned characteristics specific to mobile cloud architectures. A single pattern may not ensure all these characteristics, however the pattern collection must try to address them all. For example, unlike the traditional architectures, mobile cloud based architectures are expected to serve context-aware multiple-tenants with each tenant having its own specific context and QoS requirement that can vary from performance and reliability to security aspects. Context-aware multi-tenant capabilities of MCC systems need to be considered not only at service but also at the platform and infrastructure [17, 3] level not addressed in existing patterns [15, 16].

5. CATALOGUE BASED DOCUMENTATION OF PATTERNS

In this section, we presents the pattern catalogue in terms of the individual patterns and their representation. A catalogue essentially documents and maintains the patterns as a repository of reusable architectural solution [10, 14, 15]. In the context of a catalogue (as a repository), the pattern template (structured representation) provides the necessary elements to capture and represent the individual patterns [10]. Due to space constraints, we only present the necessary elements of pattern template as per the guidelines from [8, 10] to document software patterns in a template. The pattern template provides a structured mechanism to capture individual elements of the patterns for storage, analysis and retrieval of the patterns effectively and efficiently. Our pattern template include:

- **Pattern Name** providing unique name for pattern,
- **Intent** describes the motivation or the known uses,
- **Design Problem and Solution** provides a mapping of the problem-solution view,
- **Architecture Elements** as component and connectors that are part of the pattern,
- **Reuse Design Knowledge** that is supported by the pattern,
- **Quality Characteristics** are the attributes of the quality that are affected by the pattern.
- **Reference Diagram** provides overview of the pattern also known as pattern thumbnail.

We now discuss the three discovered patterns namely Adaptive Mobile-Cloud Offloading, Mobile Cloudlets, Mobile Sensing and Cloud Analytics that are presented in a structured template in Table 2 – Table 4 respectively. We distinguish between pattern abstraction and pattern instantiation and explain the (i) applicability and uses of the pattern, and (ii) template-based documentation of patterns.

5.1. Pattern Abstraction and Pattern Instantiation

We use an example of one of the discovered patterns named Adaptive Mobile-Cloud Offloading to distinguish abstraction and instantiation of the pattern as in Figure 3.

(1). Abstraction for Pattern Modeling

Abstraction is vital to promote a pattern as a generic solution by abstracting the complex implementation specific details as in Figure 3 (a). As illustrated in Figure 3 (a), the abstraction of Adaptive Mobile-Cloud Offloading pattern is vital to help a pattern user (designer/architect) to analyze the high-level solution for off-loading computational and storage intensive data to the cloud-based servers. Pattern abstraction helps to analyze its impacts on architecture model before pattern application (preconditions) and architectural view when pattern is applied (post-conditions) to also promote pattern as an incremental design process.

(2). Instantiation for Pattern Application

Instantiation provides the necessary details in terms of concrete architectural elements to instantiate a pattern. This is also referred to as a pattern application by means of adding refinements – extending the abstract box and arrows with architectural components and connectors from Figure 3 (a) – to pattern abstraction in Figure 3 (b) [8, 10]. In Figure 3 (b), the instance of Adaptive Mobile-Cloud Offloading pattern utilizes the Data Access Bus to bind services (in Service Pool) to data collection (in Data Source) component.

Pattern I - Adaptive Mobile-Cloud Offloading

A) Pattern Intent: To enable a mobile device to dynamically determine what, how and where to off-load its data to enhance efficiency and of mobile computing

B) Design Problem: How to enable a (resource-constrained) mobile device to delegate its memory and computational-intensive data and tasks to (resource-sufficient) computers?

C) Solution: Integrate the off-loading logic between Mobile Computing and Cloud Computing Layers. Such an integrated logic enables a mobile device to exploit dynamic parameters – such as energy efficiency, computational overhead and storage requirements – to determine and off-load data and tasks to cloud computing servers.

D) Architecture Elements: Mobile Computing Layer with (resource constrained) mobile devices, Cloud Computing Layer (resource Sufficient) server are integrated with off-loading knowledge.

E) Reuse Design Knowledge: Integration of Off-loading logic/knowledge to delegate mobile computing data and tasks to cloud-based servers.

F) Quality Characteristics:

- Elasticity of cloud services (acquiring and releasing resources) based on dynamically determined off-loading.

- QoS-driven Offloading to ensure that dynamic parameters such as energy, storage and computational efficiency.

G) Reference Diagram: Figure 3 b) represents a concrete instance of the abstract pattern representation in 3 a). Specifically, Figure 3 b) illustrates a scenario of pattern application where mobile devices are used as portable computers to capture contextual images that need analytics and processing to gather information. The image processing must be delegated to the cloud servers that have image databases to match and process the image. The offloading logic is integrated between the mobile device and cloud server to enable a mobile device to selectively and dynamically off-load the images to the appropriate cloud-based server based on energy, computational or storage efficiency.

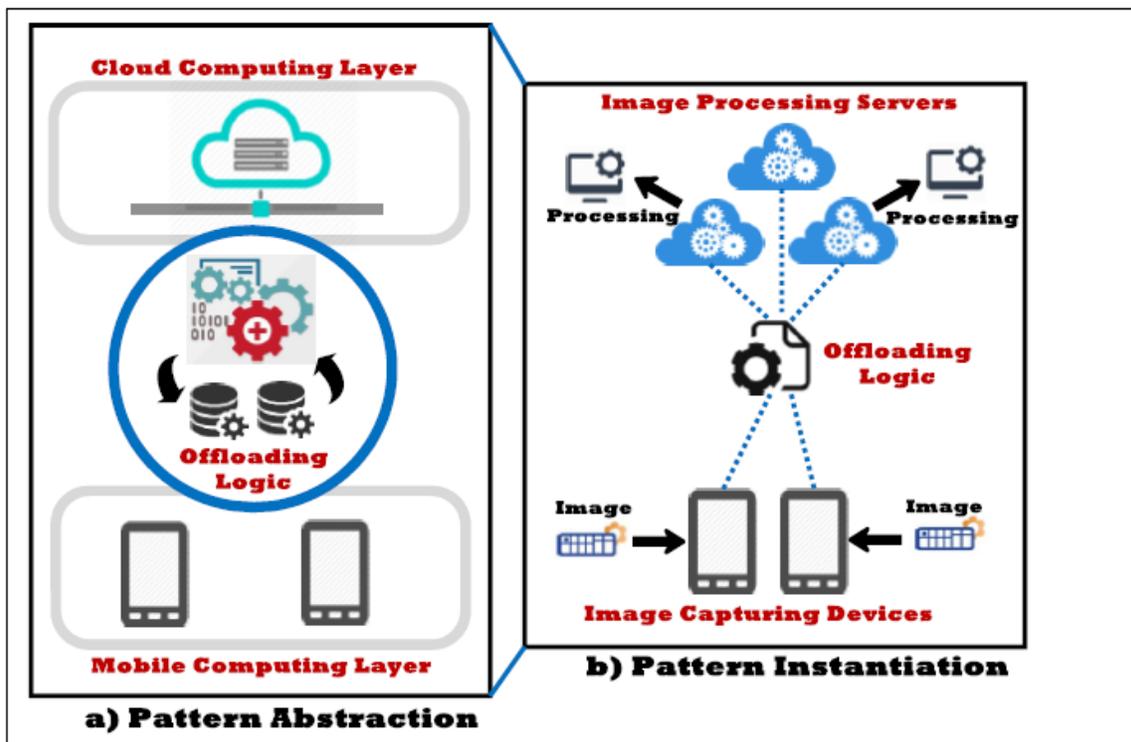


Figure 3. Adaptive Mobile Cloud Offloading Pattern
(Pattern Abstraction and Pattern Instantiation)

Pattern II – Mobile Cloudlets

A) Pattern Intent: To enable a mobile device in a hostile environment to frequently off-load data to servers that are in close proximity of mobile devices that they serve.

B) Design Problem: How to minimise the off-loading latency (on remote server) to a single-hop network while maximising the performance and QoS for mobile computing tasks?

C) Solution: The proposed architecture integrates an intermediate layer (based on localised cloud known as cloudlets) between the enterprise cloud and the mobile device. The solution assumes

that connectivity to the main cloud (enterprise) cloud is either not reliable or commonly unavailable.

D) Reference Diagram: This architecture inserts an intermediate layer between the central core (i.e., enterprise cloud) and the mobile devices. At the heart of this architecture is a large centralized core that could be implemented as one of Amazon's data centers or a private enterprise cloud. At the edges of this architecture are offload elements for mobile devices. These elements, or cloudlets, are dispersed and located close to the mobile devices they serve [23]. This architecture decreases latency by using a single-hop network and potentially lowers battery consumption by using WiFi or short-range radio instead of broadband wireless which typically consumes more energy [24] [25].

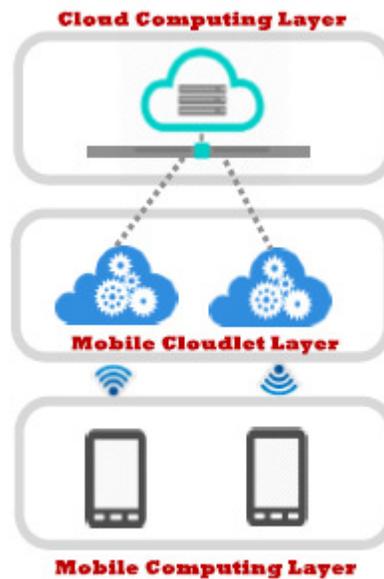


Figure 4. Mobile Cloudlets Pattern

Pattern III – Mobile Sensing and Cloud Analytics

A) Pattern Intent: To exploit the context-sensitive mobile device to capture contextual data that is sent to the cloud-server for data processing and analytics.

B) Design Problem: How to gather the contextual information that can be processed and analysed to perform decisions?

C) Solution: The proposed architecture presents a two layered architecture namely the mobile sensing and cloud analytics layer. Specifically, the mobile layers enables the capturing of the contextual information (e.g.; environmental condition, traffic congestion) and send it to the cloud based server. The cloud server runs the algorithms to analyse data and provide decision support based on processed data.

D) Reference Diagram: This architecture inserts an intermediate layer between the central core (i.e., enterprise cloud) and the mobile devices. At the heart of this architecture is a large centralized core that could be implemented as one of Amazon's data centers or a private

enterprise cloud. At the edges of this architecture are offload elements for mobile devices. These elements, or cloudlets, are dispersed and located close to the mobile devices they serve [23]. This architecture decreases latency by using a single-hop network and potentially lowers battery consumption by using WiFi or short-range radio instead of broadband wireless which typically consumes more energy [24] [25].

E) Reference Diagram

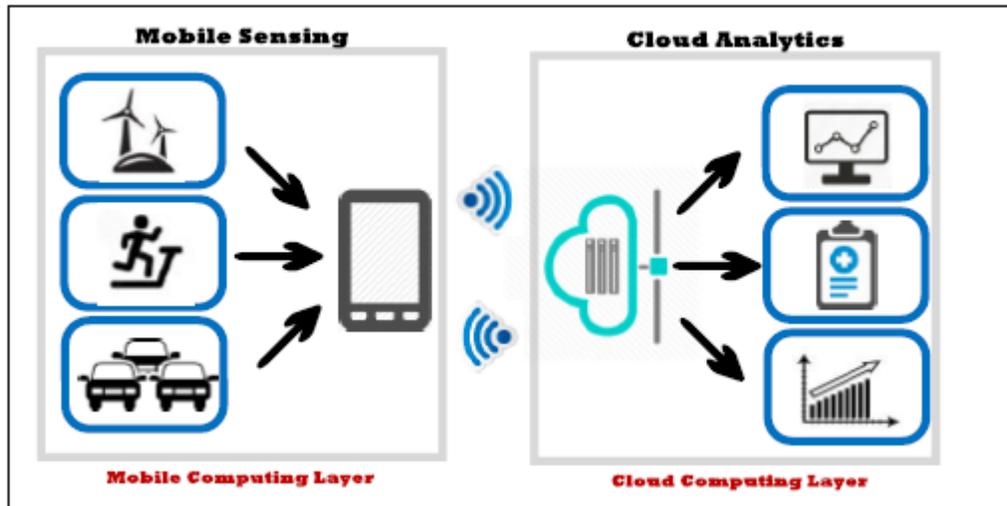


Figure 5. Mobile Sensing and Cloud Analytics

6. THREATS TO VALIDITY AND FUTURE RESEARCH

After presenting the patterns, we now discuss some validity threats and indicate some possible dimensions of the future research.

6.1. Threats to Validity

This paper reports our preliminary work on the discovery and documentation of the patterns to architect mobile cloud software systems. The number of patterns presented here is small and only aims to address and alleviate the resource poverty of the mobile devices by offloading storage and processing intensive data to the cloud-based servers.

(1) Threats to the Applicability of Patterns - We have only demonstrated the patterns and hypothetical scenarios for their application during the architectural design process. However, these patterns must be validated in a real context (case studies and scenarios) for the mobile cloud systems. Case study based validations also correspond to our possible future work. Specifically, we aim to evaluate the increased reusability and decreased efforts with pattern-based architecting. Moreover, research on design and architectural patterns have highlighted pattern-based software design as a continuous and intellectual process [9, 23]. Pattern-based architecting often requires human intervention, and collective decision before applying a specific pattern. Therefore, in addition to other types of validation, the feedback of the practitioner's is also vital to objectively evaluate the applicability and usefulness of the patterns.

(2) Threats to Continuous Discovery of Patterns - Design and architectural patterns are regarded as reusable solutions and best practices that have emerged overtime. For example, the concept of applying reuse-knowledge or best practices to design and develop software emerged from the Gang-Of-Four (GOF) design patterns over more than two decades ago [9]. With now more than two decades of research and practices, design patterns have matured to establish various catalogues and languages as an integral part of modern-day design-to-develop processes [8, 9, 10, 12]. Patterns and best practices cannot be invented, however; they can be (empirically) discovered, overtime with systematic investigation of the pattern sources.

Currently one of the main threats is the limited number of patterns as reported in this paper. However, this threat can be overcome by establish a process and identifying the sources of patterns that can be mined to continuously discover new pattern. Future research demands for a continuous and in an ideal scenario an automated discovery of patterns. Only a longterm approach to discover new patterns and extend the catalogue can prove effective. In comparison to the traditional software systems, mobile cloud systems are new and therefore need innovative patterns to-support their reusability [13, 17].

6.2. Dimensions of Future Research

In the future, our main work is to establish a catalogue of patterns that continuously grows by accommodating new patterns. However, a continuous discovery for new patterns poses two challenges. The first challenge is to identify the sources of patterns that can be empirically investigated to discover reusable knowledge and patterns from them. The second challenge requires the replacement the manual mechanism of pattern discover with a (semi-automated) discovery of the patterns.

Algorithms and Tool Support for Pattern Discovery - In future, we aim to focus on developing algorithms and tools that automate the process of pattern discovery. These algorithms and tools can support automation as well as appropriate user intervention to guided and execute the pattern discovery process. We plan to identify the logs that maintain a history of the steps/activates executed to architect the mobile cloud systems as guided by [11, 18]. The logs can provide the historical data to perform the post-mortem analysis of the architectural design. The algorithms needs to be designed to operate on these logs and to discover new patterns as best and repeatable practices [18].

7. CONCLUSIONS

Mobile cloud architectures rely on mobility and dynamically available software services that needs a recurring need for context-awareness, elasticity and scalability etc. that can be best supported by applying reusable practices and solutions. We proposed a 3-step; pattern-driven architecting process that exploits empirically discovered patterns to guide architectural design for MCC systems. A collection of patterns enhance reusability by abstracting (design primitives). Pattern discovery is a continuous process and provides a systemic approach to investigate emerging design problems and their recurring solutions.

We have highlighted some validity threats that include an objective assessment of the applicability, reusability and efficiency of the patterns based on more complex case studies. In

addition, to support the vision of pattern-based architecting for MCC, new patterns must be continuously discovered and also validated by the pattern users/designers.

REFERENCES

- [1] M. Satyanarayanan. Mobile Computing: the Next Decade. *ACM SIGMOBILE Mobile Computing and Communications Review*. vol 15, no 2, pp: 2-10, 2011.
- [2] G. P. Picco, C. Julien, A. L. Murphy, M. Musolesi, G.-C. Roman. Software Engineering for Mobility: Reflecting on the Past, Peering into the Future. In *Proceedings of the on Future of Software Engineering (FOSE)*, pp: 13–28. ACM, 2014.
- [3] S. Simanta, K. Ha, G. Lewis, E. Morris, and M. Satyanarayanan. A Reference Architecture for Mobile Code Offload in Hostile Environments. In *Fourth International Conference on Mobile Computing, Applications and Services (MobiCASE)*,), pp: 274–293, 2013.
- [4] G. Lewis, S. Simanta, M. Novakouski, G. Cahill, J. Boleng, E. J. Morris, J. Root. Architecture Patterns for Mobile Systems in Resource-Constrained Environments. In *Military Communications Conference (MILCOM)*, pp: 680–685, 2013.
- [5] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia. A View of Cloud Computing. In *Communications of the ACM (CACM)*, vol 53, no 4, pp: 50 – 58, 2010.
- [6] IBM Developers Work. Mobile Cloud Computing Devices, Trends, Issues, and the Enabling Technologies. [Online:]<http://www.ibm.com/developerworks/cloud/library/cl-mobilecloudcomputing/>, Accessed 02/12/2015.
- [7] H.T. Dinh, C. Lee, D. Niyato, P. Wang. A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches. In *Wireless Communications and Mobile Computing*, vol. 13, no. pp: 1587 – 1611, 2013.
- [8] F. Buschmann, K. Henney, D. C. Schmidt. *Pattern Oriented Software Architecture vol 5: On Patterns and Pattern Languages*. Wiley and Sons, ISBN-13: 978-0471486480, 2007.
- [9] C. Pahl, S. Giesecke, W. Hasselbring. Ontology-based Modelling of Architectural Styles. In *Information and Software Technology*. vol. 51, no. 12, pp: 1739–1749, 2009.
- [10] N. B. Harrison, P. Avgeriou, U. Zdun. Using Patterns to Capture Architectural Decisions. In *IEEE Software*, vol. 24, no. 4, pp: 38-45, 2007.
- [11] J. Cámara, P. Correia, R. Lemos, D. Garlan, P. Gomes, B. Schmerl, R. Ventura. Evolving an Adaptive Industrial Software System to Use Architecture-based Self Adaptation. In *International Symposium on Software Engineering for Adaptive and Self Managing Systems (SEAMS)*, pp: 13 – 22, 2013.
- [12] J. Kim. Architectural Patterns for Service-based Mobile Applications. In *IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*, pp: 1-4, 2010.
- [13] B. Wilder. *Cloud Architecture Patterns*. O’Reilly Media, Inc, ISBN 10: 1-4493-1977-7, 2012.
- [14] C. Fehling, F. Leymann, R. Retter, D. Schumm, W. Schupeck. An Architectural Pattern Language of Cloud-based Applications. In *Proceedings of the 18th Conference on Pattern Languages of Programs (PLoP)*, 2011.

- [15] Cloud Computing Design Patterns: [online:] accessed on January 5, 2016. <http://www.cloudpatterns.org/>
- [16] Cloud Design Patterns. [online:] accessed on January 6, 2016. http://en.clouddesignpattern.org/index.php/Main_Page
- [17] J. Roth. Patterns of Mobile Interaction. In *Personal and Ubiquitous Computing*, vol 6, no 4, pp: 282 - 289, 2002.
- [18] A. Ahmad, A., P. Jamshidi, C. Pahl. Graph-based Pattern Identification from Architecture Change Logs. In *10th Workshop on Systems/Software Architecture*, pp. 200-213. Springer. 2012.
- [19] G. Lewis, M. Novakouski, and E. Snchez. A Reference Architecture for Group-Context-Aware Mobile Applications. In *Mobile Computing, Applications, and Services*, volume 110 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp: 44–63. Springer, 2013.
- [20] Architectural Patterns, Reference Models, and Reference Architectures, [online:] accessed on January 6, 2016. <http://www.ece.ubc.ca/~matei/EECE417/BASS/ch02lev1sec3.html>
- [21] K. Z. Chen. Integration of Design Method Software for Concurrent Engineering using Axiomatic Design. In *International Journal of Manufacturing Technology Management*, vol 9, no 4, pp. 242–252, 1998
- [22] P. Jamshidi, A. Ahmad, C. Pahl. Cloud Migration Research: A Systematic Review. In *IEEE Transactions on Cloud Computing (IEEE TCC)*, vol 1, no. 2, pp: 142 – 157, 2013.
- [23] A. Ahmad, P. Jamshidi, C. Pahl. Classification and Comparison of Architecture Evolution Reuse Knowledge – A Systematic Review. In *Journal of Software Evolution and Process*, vol 26, no 7, pp: 654-691, 2014. [m.org/10.1145/161468.16147](http://dx.doi.org/10.1145/161468.16147).
- [24] A. Ahmad, A. B. Tamimi, N. Saeed, M. Hamayun, M. Fraz. Research Protocol of Software Architecture for Mobile Cloud Systems: A Mapping Study. Technical Report, College of Computer Science and Engineering, University of Ha'il, 2017. [Online] <http://media>.