Rapid mHealth – a mobile healthcare application development framework

Senthil Kumaresan, Satya Sai Srinivas, Anutosh Maitra* and Nataraj Kuntagod

Accenture Technology Labs, Accenture,
Divyasree Technology Park, Kundallahalli,
560037, Bangalore, India
Email: senthil.kumaresan@accenture.com
Email: satya.sai.srinivas@accenture.com
Email: anutosh.maitra@accenture.com
Email: nataraj.s.kuntagod@accenture.com
*Corresponding author

Abstract: A new philosophy and architecture of rapid mobile healthcare application development is presented here. Many monolith applications developed on the Lambda architecture had been successful earlier, but of late, this architecture is found inadequate when deployed for different stakeholders. Today's healthcare applications handle large volume of streaming data obtained from dispersed devices. The architecture is to support multiple use cases as the business models are evolving and the products have to be malleable enough to support different functions. This paper describes a working architecture on which mobile healthcare applications can be rapidly built through ingestion and processing of voluminous streaming data. The use of micro-services keeps this architecture flexible. On the client side, application platform as a service (APaaS) frameworks are employed to assist quick development and validation of solutions. Findings from some successfully deployed mobile healthcare applications based on this architecture are also discussed.

Keywords: mobile healthcare; prototype; microservices; application platform as a service; APaaS; internet of things; IoT.

Reference to this paper should be made as follows: Kumaresan, S., Srinivas, S.S., Maitra, A. and Kuntagod, N. (2017) 'Rapid mHealth – a mobile healthcare application development framework', *Int. J. Telemedicine and Clinical Practices*, Vol. 2, No. 1, pp.42–62.

Biographical notes: Senthil Kumaresan is a R&D Technologist at Accenture Technology Labs. He has nine years of experience in enabling novel IT solutions (SMAC, IoT and machine learning based systems) and architected multiple award winning solutions in the process. His current research focus is to build technology solutions that could work under constrained environments prevalent in emerging economies for larger societal impact. He has also published multiple technical papers and is holding multiple patents in the mobile application areas. He has a Bachelor's degree in Engineering from Amrita Institute of Technology, Coimbatore, India.

Satya Sai Srinivas is a Solutions Architect at Accenture Technology Labs and working on SMAC and IoT-based offerings. He has 16 years of IT experience across various domains including service oriented architecture,

social computing and distributed computing. He is actively involved in evangelisation of various new technology frameworks. He has done his Master's degree in Engineering from Indian Institute of Science, Bangalore, India.

Anutosh Maitra is a Senior Researcher and Research Program Manager in Accenture Technology Labs. He did his PhD in Electrical Engineering from Indian Institute of Technology and has over 20 years of R&D experience in industry and academia. A specialist in data and text mining and analytics, he holds a multiple patents and has a large number of publications internationally. He is also an invited faculty member in various business and technology institutions globally.

Nataraj Kuntagod serves as VP, Open Innovation at Accenture Technology Labs. He has introduced several transformative solutions and products for the 'next six billion' in the areas of agriculture, healthcare, M-gov, energy and digital citizens. He has 23 years of broad industry experience in various technology domains. He also held several senior engineering and management positions at large product manufacturers and as an entrepreneur, working in areas like cellular infrastructure, enterprise mobility, SATCOM, connected vehicles, mHealth and IoT. He holds multiple patents in the area of wireless/IoT and has a Master's degree in Electronics Design from Indian Institute of Science, Bangalore, India.

1 Introduction

Rapid advancement in sensor and mobile internet technology, ease of use and the falling cost of devices make mobile an appropriate and adaptable tool to bridge the digital divide. The massive growth of mobile internet in the emerging markets has spurred many governments and other organisations to start delivering digital citizen centric services. Although providers and patients in both developed and developing countries face similar issues, there is a significant difference in terms of needs and priorities. In much of the developed world efficient, proactive treatment by tracking and monitoring and lifestyle management is the demand, whereas in low and middle-income countries, coverage and accessibility of healthcare services to vast population have been the prime need. Also, the advancements in internet of things (IoT) technology and related trends of the mega investments in the area bring into focus the massive cluster of sensors and smart devices, an estimated 50 billion of them by the year 2020, that have the ability to sense their environment and communicate (Carter, 2015). Slowly, but surely, IoT industry is moving towards smart services evolving from smart devices and connected smart products. User centric smart healthcare services are getting widely popular. However, such smart services need an in-depth understanding of the user's preferences and requirements dynamically over place and time and as such for these 'healthcare' sensors, the balance is fast shifting towards accessing 'continuous physiological data' rather than analysis of 'on-spot' measurements. Traditional information systems housed in healthcare hubs often fail to handle large volume of continuously streaming data; but it is difficult to do away with these existing expensive legacy systems. So there is a growing need for an integration framework that ties the conventional static data oriented systems with the new mobile data oriented information systems. Streaming data oriented healthcare analytics a major demand of modern day mobile healthcare applications.

1.1 Background of healthcare services in developing countries

In many developing countries, the government run healthcare facilities have inadequate infrastructure and relatively poor quality of service. This makes more and more people rely upon private healthcare providers who are facing their own challenges to develop a sustainable business model. In addition, the market has fairly small number of hospitals offering tertiary care compared to the existing and projected demand. Cost of quality healthcare can become a heavy burden for the low and medium income families, especially for aging populations or people suffering from chronic ailments. Tertiary care has seen, and projected to see, heavy market expansion in the coming decade (Dinodia Capital Advisors, 2014). The tertiary segment allows for higher revenue realisation per patient, presents a significant cost barrier for new entrants, and allows the players to differentiate their product offerings. Globally, on an average, there are about 2.9 hospital beds per 1,000 population (NationMaster.com World Development Indicator Database, 2014) and this presents a significant window of opportunity for private healthcare providers. Nonetheless, the average operating margin of the tertiary care hospitals have been declining, partly due to the high capital investments required per bed. Operating margins are also shrinking because of increasing expenses of advanced medical equipment and the cost of employment of skilled professionals. Further, beds are not optimally used due to the lack of proper information management and as the standards and regulations are often in a nascent stage, even in many developed countries, the service charges have to be restrained to stay ahead of the competition.

It is in this backdrop, the Hub and Spoke model (Health Planning and Infrastructure Division, Government of Australia, 2010) of healthcare business evolved to optimise the utilisation of resources by differentiating between the type of care needed and the facility to be utilised. While the tertiary care with expensive medical infrastructure was to be provided by the hub hospital, the spokes used to cater to the non-tertiary needs of the patients. However, the increase of lifestyle diseases, aging population with chronic illness and the post-operative monitoring is putting a strain on this model. Spot measurements of vital signs of patients in these segments do not provide good enough insights into their health. Continuous monitoring of vital physiological parameters over a period of time is required to provide deeper insights into their health and comprehensive treatment in a proactive manner. However, continuous monitoring of vital signs requires expensive medical equipment at the spoke hospitals too. Skilled personnel are needed to operate these sophisticated equipments, increasing the cost of operations even more and squeezing the operating margin, and negating the advantages of the hub and spoke model.

Remote patient monitoring systems that use wireless and information technologies to collect physiological data of patients and transfer them to a backend server for analysis and remote management of patients have shown great promise to reduce the high cost of healthcare. The availability of low cost physiological sensors and longer battery life due to the advent of low power body area wireless networks is giving rise to new ways of measuring vital signs; from discrete to continuous data measurement. Such continuous physiological monitoring driven services enable hospitals to provide round the clock monitoring at all beds and also to remotely monitor patients suffering from chronic ailments. This increases tertiary patient handling capacity of the hospitals without significant increase in expenses. Since this service can also be extended to spoke hospitals and homes for managing secondary and primary care, the hospital infrastructure can now exclusively cater to tertiary patients without having to deal with sub-optimal utilisation of beds. The positive effect on the operating revenue for the providers would also get amplified by enhancing existing healthcare quality and providing a differentiated service from competitors.

Other than the government, patients and private healthcare providers, the interests of many other parties like the network operator, device manufacturer and software system developers and integrators are also involved in this situation. Business models that would maximise the interests of all stakeholders are still evolving and may be highly complex in their implementation. One of the major steps to develop such an integrated and sustainable business model is to visualise and implement a framework that would enable rapid application development utilising continuously streaming data obtained from multiple dispersed sensors and other mobile devices. Moreover, the architecture has to be flexible to support multiple business use cases as the interests of the multiple stakeholders involved in the eco-system are often different. One cannot afford any rigidity in the system as it is expected that the business models would be further evolving and the architecture would have to sustain incorporation of multiple new services.

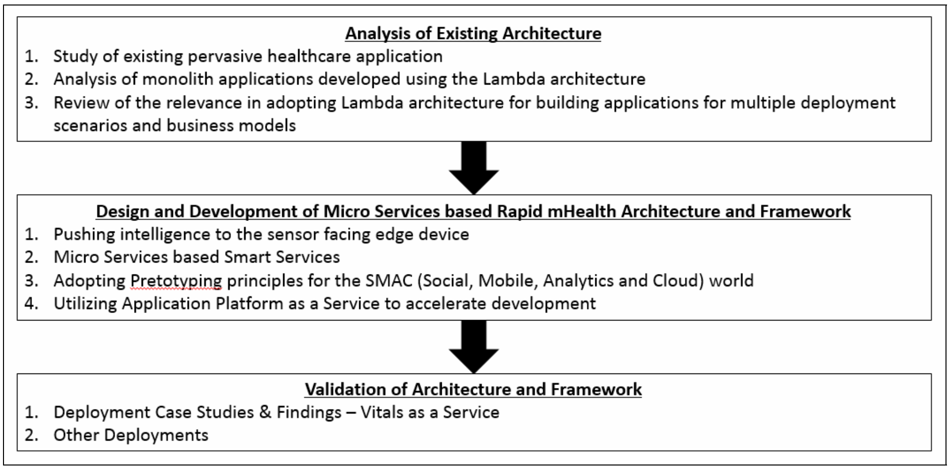
1.2 Scope and objective of the present work

The main objective of this literature is to propose the framework, technical components and guidelines to develop mobile health applications rapidly. The applications would have to cater to multiple service oriented deployment scenarios and business models. Results from multiple case studies developed using this architectural framework is examined and presented. The aim is to utilise this framework to rapidly build digital applications, which can also ingest huge amounts of data from multiple sensor networks, across multiple domains. Such a framework also needs to take into account the IoT protocols, and the medical compliance guidance related to the sensors and devices, however the present paper only discusses the software side of the framework to build healthcare applications, with an assumption that a separate data aggregation framework exists to ingest data streams from a sensor facing edge device.

2 Methodology

The research and development methodology is divided into three sections – an analysis of existing architecture widely used by the application developers to build mHealth applications, design and development of a new rapid mHealth framework and validation of the proposed framework. Multiple activities involved in these stages are schematically shown in Figure 1 and the succeeding sections discuss these steps in brief.

Figure 1 Summary of research methodology



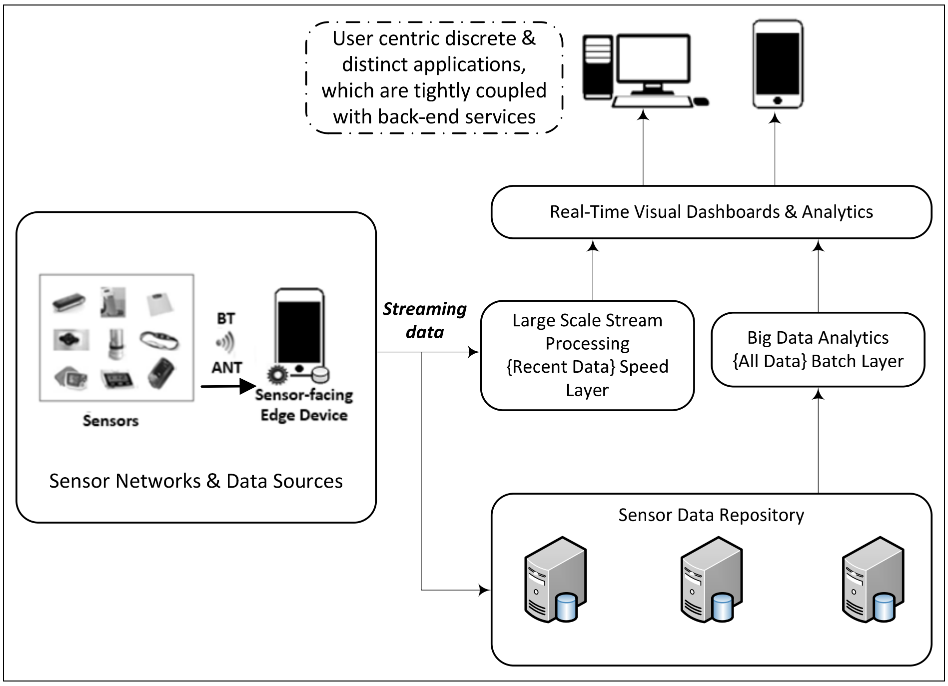
2.1 Analysis of present day architecture

A number of pervasive healthcare applications including pervasive health monitoring, intelligent emergency management system, pervasive healthcare data access, and ubiquitous mobile telemedicine have been developed over the last decade (Varshney, 2007). One major application in pervasive healthcare using data collected over smart phones, termed comprehensive health monitoring was presented in significant details (Toninelli et al., 2009) using wireless networking solutions for wireless LANs, adhoc wireless network and cellular/GSM/3G infrastructure oriented networks. A medical implant communication service band (MICS) wireless body sensor network presents (Yuce et al., 2007) a body sensor network design and implementation of patient physiological data collection for health monitoring purposes. The MICS band offers the advantage of miniaturised electronic devices that can either be used as an implanted node or as an external node. A prototype sensor network was implemented by incorporating temperature and pulse rate sensors on nodes. Each developed sensor node has the capability of physiological data acquisition and local processing. The sensor node can also transmit data over the air to a remote central control unit (CCU) for further processing and storage. The wireless body area network emerged as a dominant technology (Li et al., 2010) for e-healthcare to allow the data of a patient's vital body parameters and movements to be collected by small wearable or implantable sensors and communicated using short-range wireless communication techniques.

New sensors are continually being approved by the medical fraternity and are adopted for use with multiple healthcare information systems. This can fuel vast amount of data from millions of endpoints streaming at high velocity in the cloud. Several data models to process big data are already developed and a number of such models are still emerging. The growing role of cloud in big data ecosystem is well investigated in recent times (Sharma, 2015, 2016). Hadoop was the de-facto standard for storing and analysing large volumes of data till a few years back (Lee et al., 2013) Hadoop stores and analyses data through batch processing. In order to analyse the stream of data in real time, Nathan Martz defined the famously known Lambda architecture, through which one can combine processed data in Hadoop with processed data from a stream processor, and analyse all information generated simultaneously at any instant.

Figure 2 represents a logical block of a widely adopted Lambda architecture today (Marz and Warren, 2013). Lambda architecture is a data-processing architecture designed to handle massive quantities of data by taking advantage of both batch and stream processing methods. The software stack consists of a cloud data repository at the bottom layer to store the data that is generated from the server. In the middle layer, a large-scale stream processor module handles the streaming data from the sensors. The middle layer also hosts the big data analytics module that can process the data from the sensor data repository to provide meaningful insights. The top layer consists of real-time visual interfaces, developed for the digital world to assist decision making and real-time support for the end users.

Figure 2 Lambda architecture (see online version for colours)



In today's world, business or users can no longer wait to visualise the streaming data from the sensors. The existing monolith applications developed from the Lambda architecture are considered inadequate in the situations where the applications have to be deployed for different stakeholders with varied profiles and multiple business interests.

2.2 The proposed rapid mHealth architecture and framework

The proposed architecture and framework to build rapid mHealth applications for today's streaming and data rich digital world addresses three specific needs:

a ability to handle voluminous streaming data from dispersed devices and sensors

b ability to provide a loosely coupled service oriented architecture

c ability to provide intelligence on the edge.

An advisory from Gartner published in 2012 propagated the idea of adopting a pace-layered strategy to accelerate innovation (Gartner IT Glossary, 2012). The advisory defined three distinct application categories:

a systems of record – established packaged applications or legacy homegrown systems

b systems of differentiation – applications to enable processes unique to the enterprise

c system of innovation – new applications to be built to address new business opportunities or requirements.

IoT-based mHealth information systems typically fall into the categories of systems of differentiation and innovation. Attributing innovation to large development cycles and release timelines are no longer fancied by the organisations. The shelf life of IT products itself is decreasing rapidly. Mobile first services have hardly 12 months to make their mark before becoming obsolete due to technology innovation or competitors taking over and offering a better service. Products getting obsolete in three to five years a decade ago seems a lifetime in the current fast-paced product development marketplace. Organisations are looking for affordable and quality processes to test the solution even before the applications are built. Once the applications are validated by the business users, the organisations would like to rapidly develop the application using no or low code platforms which can help to compose applications using REST-based services. The proposed architecture is designed on the key principle of developing validated applications at the pace of business by composing smart services. The key ingredients of the framework are:

a intelligence at edge

b micro services-based smart services

c pretotyping principles for the social, mobile, analytics, cloud (SMAC) world

d application platform as a service (APaaS).

Figure 3 presents a logical architecture of the framework. Table 1 summarises the characteristics changes brought by the proposed framework to the state of the art. Key ingredients of such changes are presented in the succeeding sections.

Figure 3 Logical architecture – proposed framework (see online version for colours)

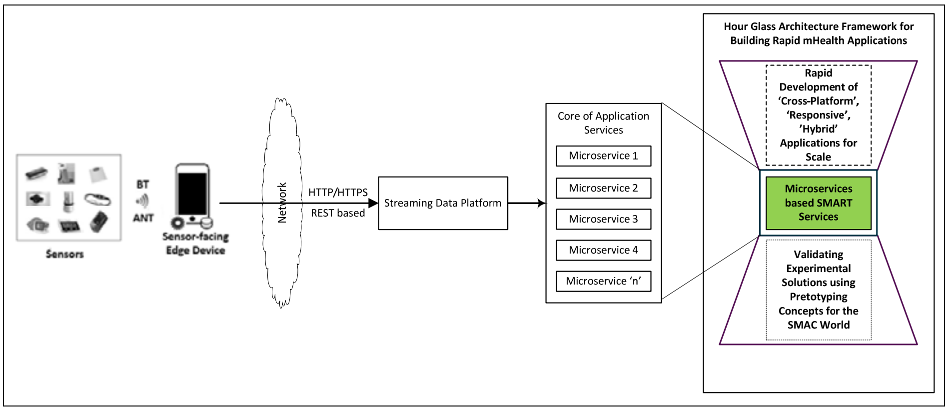


Table 1 Comparative study of the existing and the proposed architecture

Characteristic	*Existing present day architecture*	*Proposed rapid mHealth architecture and framework*
Rigidity	Developed applications are rigid as the client and server applications are tightly coupled, any change in the client or in the server side can affect the entire application.	Developed applications are not rigid as the client and server applications are loosely coupled, any change in the client or in the server side will not affect the entire application.
Flexibility	The architecture is not flexible to cater to multiple business models. The developers have to create a different application from the scratch for a new business model as the architecture can support only a restricted amount of reuse	The architecture is flexible to cater to multiple business models. The same application can be easily customised to address a new business model, since most of the components can be individually assembled
Effect of single component failure	Any single component failure can potentially bring down or affect the entire application since all functions are packaged inside a single enterprise archive	Single component failure will not bring down or affect the entire application since all functions are separately packaged and each component can be individually inspected
Scaling	It has been found that applications developed on this architecture do not scale up efficiently. This scaling is possible only by deploying the same enterprise archive on multiple servers and duplication of services leads to inefficient design.	Vertical and horizontal scaling is possible depending upon the business needs and operational analytics.
API economy	The architecture, by virtue, is unable to realise the value of API economy. Without an API management gateway, organisations cannot fuel internal innovation, reach new customers, extend products and create a vibrant partner ecosystem.	Built on top of the API management gateway, it helps the organisations to realise the API economy by extending their core API's to their internal and external stakeholders to build a vibrant partner ecosystem

2.2.1 Pushing intelligence to the sensor facing edge device

Much of the developing world has a concern over continuous network accessibility of faraway places. The philosophy of rapid healthcare application is based on the availability and reliability of communication network. The success of this new information management technique can only be addressed if immediate advisories to patients can be made independent of the communication network. Increased storage and processing capacity on mobile devices led to the concept of intelligence on the edge (Islam and Gregoire, 2010) that allows certain analysis and decision making to be handled by the client devices for the rapid intervention on the patient's condition. Obviously, any integrated platform to this end has to provide the flexibility of processing on the server as well as on the client side depending on the services the user is looking for.

The proposed framework consists of a decision support system built on top of the open Android mobile operating system. C Language Integrated Production System (CLIPS) rule engine was ported to Android and a dynamic context management framework is built around it. The guidelines specific to the mHealth application can be encoded as CLIPS rules, which can use the input from sensors, patient and caregiver. The CLIPS rule engine was ported and made as a library using the Google native development kit (NDK). Context manager developed in Java takes the raw context from multiple sources and based on the aggregate context, chooses the further course of action.

2.2.2 Micro services-based smart services

Micro services-based smart services (Strang, 2003) form the stem of the proposed architecture. Micro services is a software architecture style in which complex applications are composed of small, independent processes communicating with each other using language agnostic API's. These services are usually small, highly decoupled and focus on performing a small specific task. These smart services are instantiated from the streaming data platform and form the core of the application services. Developers, partner ecosystems and different stakeholders related to the business can utilise the published micro-services to accelerate the innovation or application development process.

2.2.3 Adopting 'pretotyping' principles for the SMAC world

'Pretotyping' is widely regarded as the method for quick and inexpensive validation of IoT based mHealth systems applications and ideas (Sironi, 2011). Pretotyping has become the key bridge between idea conceptualisation and the real product development in recent years. The method enables organisations to do a quick and cheap demand and feature validation with the target market at the highest fidelity. Traditional software development did not stress much on user experience. However, as the world is moving to digital, the software development life cycle is fast changing. The user experience is becoming the key to a product's success and one cannot afford to wait for months before experiencing how the system will look and behave. This translates to a need to have a mechanism of very quickly experiencing and validating the final system, ideally in a few days. In addition, given the fact that there would be multiple devices with different form factors and capabilities, such experience and validation become more important as the system interactions could be quite complex when deployed to practice. Multiple off the shelf pretotyping tools are available today. Tools like Justinmind, ProtoShare, DAIO, Stand In, Concept.ly can create interactive wireframes catering to a wide variety of devices and interactions.

2.2.4 APaaS to accelerate development

Once an idea or a new application is validated by pretotyping, organisations can prioritise and plan to build the applications at scale. Cloud is disrupting the way most software is consumed and deployed. Many organisations have successfully adopted cloud model because of the benefits in terms of the elasticity, agility and operational cost savings. The simplicity of a browser-based application compared to that of a desktop software made the cloud software a preferable one for the end users as well. But, the software development tools have not moved to the cloud yet. Even today, most of the developers download gigabytes of software libraries install it on their machines. XCode, as an example, is a heavy download at a couple of gigabytes, which developers have to re-download with every update if the developers want to stay current, and then configure and maintain it. Developers have recently started to adopt cloud-based development environment since the modern browsers are fast, HTML5 is more mature and networks are faster and ubiquitous from the development environment scenario. The benefits offered by a cloud-based development platform overweigh the inconvenience of an occasional drop in connectivity. The benefits of a cloud-based development platform are observed in:

1 operational efficiency since there is no need to download, configure and maintain tools and SDK's

2 ability to develop from anywhere, however remote the place is; all that a developer needs is a working network connection

3 real time collaboration as the project is shared in a centralised location, updates made by each developer are almost visible in real-time

4 rapid jumpstart and improved productivity result in shortened delivery cycles.

This trend has paved way for the APaaS, where the developers would develop and deploy the application using cloud-based development environments through a drag and drop-based visualisation with no or less code (Lathar et al., 2014). Quite a few no/low code tools like Appery.io, Ionic.io, Progress, Appzillon and Telerik are fast gaining popularity amongst mobile software developers.

2.3 Validation of the proposed architecture and framework

Figure 4 presents the technical architecture of the overall framework. In short, the architecture consists of the following important sub-systems:

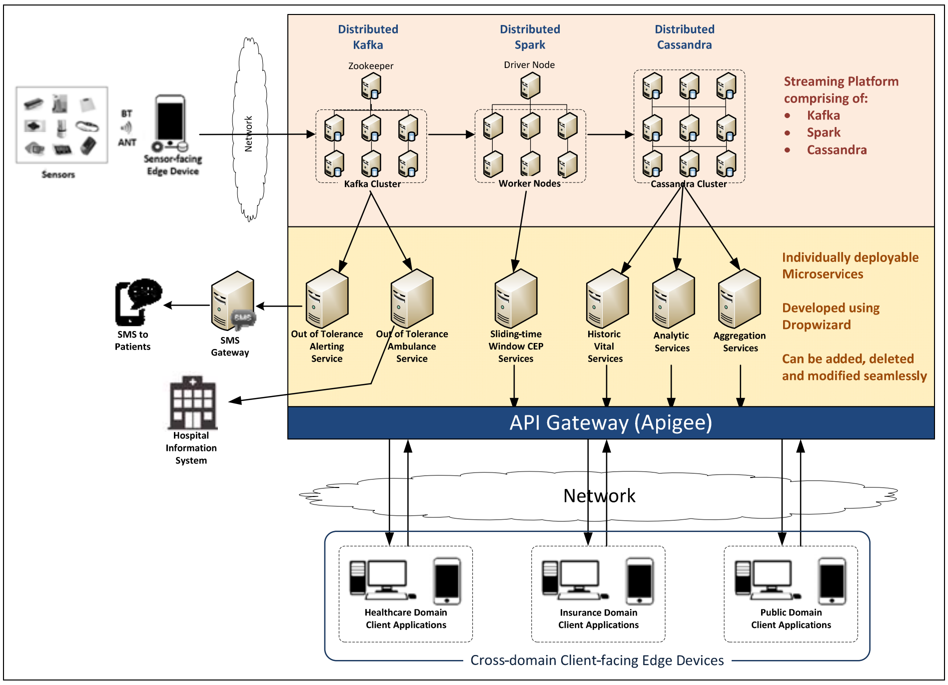
- Sensors: High precision sensors are now available that can continuously monitor the vitals of the patient in real time. These sensors are low powered devices and are only meant for capturing data and sending it across to a paired sensor-facing edge device via short-range wireless technologies such as Bluetooth, ZigBee, Z-wave and ANT.

- Sensor-facing edge devices: The primary function of these devices is to collate the data of all the sensors that are within its range and send it across to the server via Http, REST or proprietary network protocols. Apart from its primary role of delegating the sensory data to the server, a certain amount of analytical intelligence can also be built into these devices for computing thresholds and generating alerts. Additionally, in case of non-availability of network, these devices can store the sensory data temporarily and once the network becomes available it sends the time-stamped data to the server in batch mode.
- A highly-efficient streaming platform: For a system that monitors a large sized patient community, the expected throughput is of the order of 100 K messages per second and the expected volume is of the order of terabytes. The traditional systems comprising of conventional application servers and RDBMS databases would fail under such high-performance and scalability requirements. The proposed architecture brings in a highly-efficient distributed streaming platform comprising of:
  - a Apache Kafka, a highly efficient, distributed, publish subscribe kind of messaging system.
  - b Cassandra, a decentralised, distributed, linearly-scalable NoSQL database. Recent reviews (Sharma et al., 2014) indicated the efficacy of Cassandra in the leading big data models with its ability to handle huge streaming datasets with massive scalability. Apache Spark Streaming, a very fast, streaming framework is available for moving the messages across from Kafka to Cassandra.
- A flexible micro-services-based offering: Once the large volume of sensory data is captured, a flexible system is needed to extract meaningful insights out of this data. Each small chunk of insights makes one micro-service. Identifying appropriate data-source for exposing each service becomes the key differentiating factor in the application development stages. For example, out-of-tolerance-based services have to tap the data from Apache Kafka, while the analytic, historic and aggregator services can be exposed by tapping the data from Cassandra. These services are independent of each other, so implementing them as a set of micro-services is natural. Following are some of the advantages that micro-services philosophy can offer:
  - a since the services are independently deployed the new ones can be added and the unused ones can be removed independently
  - b depending on the load on each of the services, they can be individually scaled through seamless operations
  - c any requirement changes in these services can be individually accommodated, without affecting others.

These services are interfaced through an API gateway such as Apigee so as monitor the usage of the services and then monetise accordingly.

- Cross-domain applications for client facing edge devices: Existing off the shelf products that focus on APaaS can be used to rapidly compose applications using the no/low code platforms.

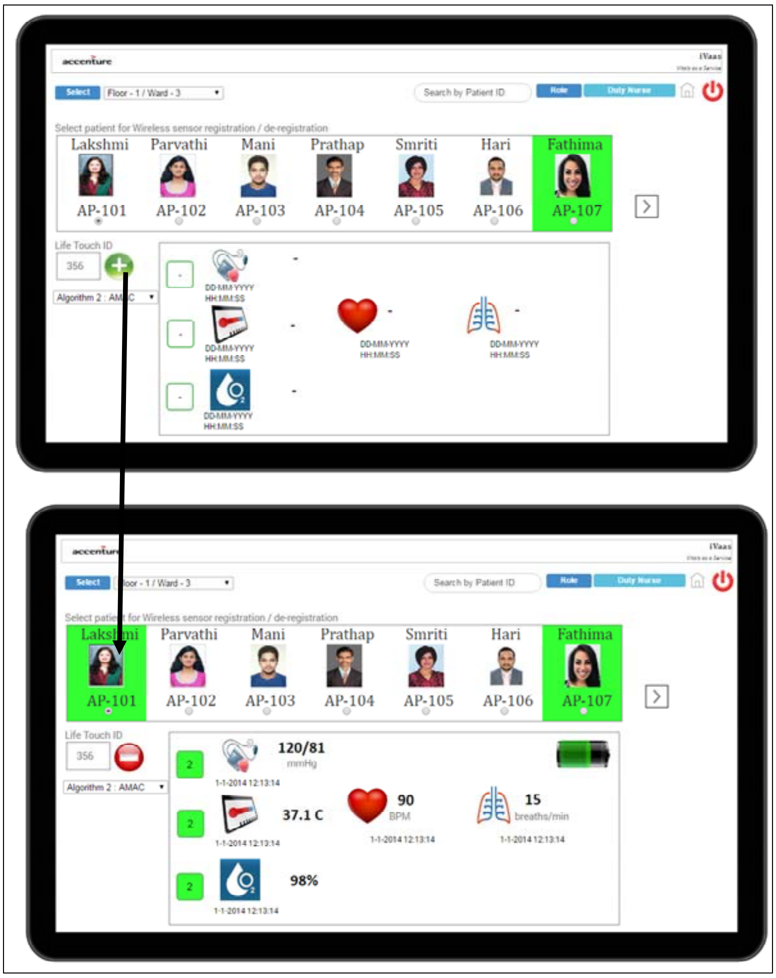
Figure 4 Technical architecture – proposed framework (see online version for colours)



2.3.1 Deployment case studies and findings – vitals as a service

One major deployment of the proposed architecture is in one of the products called 'vitals as a service' (VaaS), developed jointly by Accenture and its partner (iSansys Report, 2016). The low cost sensor-based platform allows continuous monitoring to patients across all beds in the hospitals, and makes it possible for the creation of 'virtual' beds remotely at home for secondary and primary care. The platform can provide an end-to-end ecosystem for continuous vital signs measurement and data capture which can be overlaid in the existing hospital IT infrastructure. VaaS can be deployed in a hospital, at a primary care centre, direct to the patient, to underwrite insurance policies and other applications too. Multiple business models were also supported by the VaaS platform. The core unit of measurement for monetisation is the 'patient day of monitoring' and the number of continuous monitoring sessions per day where each session could last from 30 min to 8 hours. Pretotyping was used to build a high fidelity, touch-based interface and deployed it on an actual device to show it to the end users. Figure 5 highlights a use case of how a 'duty nurse' could pair a sensor-based device to a patient for continuous monitoring using expert defined business rules.

Figure 5 Vaas user interaction screen (see online version for colours)



VaaS used Kafka, Storm and Cassandra as the technology blocks for the streaming platform. Multiple micro services like historic vitals services, sliding window CEP services, aggregation services and others were instantiated from the above blocks. These multiple micro-services were coupled with Apigee to deliver an intelligent API platform to accelerate the pace of digital business. With the use of an API management gateway, it was possible to understand the user behaviour across all digital channels, predict the behaviour and profile data to advise the next-best action and turn predictions into actions with real-time API's. The API management gateway also helped fine tune the product with the use of API analytics like identification of widely used services or poorly used services by different user groups. Based upon these analytical insights, organisations could take deployment decisions like phasing out a poorly used feature of the end user application rather than spending time and effort to maintain it. As mentioned before, the micro-services formed the stem of the hourglass. The partner ecosystem or internal and external innovators could then utilise the no/low code platform to rapidly build applications by composing these smart services.

2.3.2 Other deployments

The same framework had been deployed in other mHealth applications as well. One of these solutions tries to address the challenge of reducing the maternal and infant mortality (Kuntagod and Maitra, 2013). A polyclinic in Southern India was the site of first field deployment of the mHealth solution. The polyclinic handled 400 maternity outpatient visits and 35 deliveries every day, severely straining the polyclinic infrastructure. The polyclinic conducted the field trial in order to understand the implication of large-scale outreach deployment as 90% of the maternal patients came from rural areas. Using the mHealth solution, the hospital was able to see a 3 times increase in the patient handling capacity for each day.

Another deployment, named GPower, was in eastern India, where the healthcare outreach program monitored 1200 adolescent girls to flag off vulnerabilities detected in health, education, nutrition, and protection and to initiate real-time action to mitigate potential problems (Ghosh et al., 2015). Another successful mHealth solution in vision care was built to eradicate 80% of preventable blindness by helping the field workers, caregivers and doctors provide timely health assistance and support at the point of care (Kuntagod et al., 2015). The pilot deployment was carried out from Coimbatore and its surrounding rural areas with a major eye care provider in India, having a pan India presence. The outcomes were a significant increase in the number of patients surveyed by the field workers, increase in the conversion of referrals to surgery cases, reduction in absenteeism of field workers and reduced monitoring and publicity costs.

3 Results and discussions

In general, 'Pretotyping' helped in validating business needs and honing the requirements iteratively and enabled a meaningful dialogue with the end users as well as the healthcare business owners. The technique helped capture the requirements of the core service offerings in a couple of meetings. A complete new digital service offerings, comprising 16 different active screens could be pretotyped in a week, and saved six months at a conservative estimate for the application development. Use of this pretotype with minor alterations catering to the specific needs of other hospital chains resulted in extremely fast validation of the business service variations. By utilising the micro-services concept and no/low code platforms, the application developers located in different geographies were able to build multiple variations of the applications rapidly. In a few months, the separate needs for multiple deployment scenarios and business models could be addressed, simply by orchestrating the different services using a visual environment.

3.1 Efficient management of requirements volatility with pretotyping

Results presented above highlight the effect of pretotyping technique in requirement management on healthcare applications, Figure 6 shows that for the vision care management solution, a medium complexity application, the requirements volatility became very low in less than four weeks through the adoption of pretotyping principles. Similarly, Figure 7 shows that by adopting these principles, for a high complexity application as Gpower, a number of additional requirements could be captured early in the project cycles and the requirements volatility also started to reduce from the 6th week of the project initiation.

Figure 6 Requirements volatility – vision care management solution (see online version for colours)

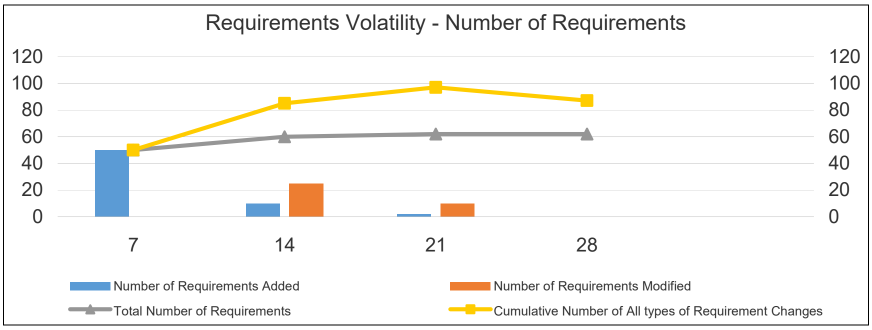
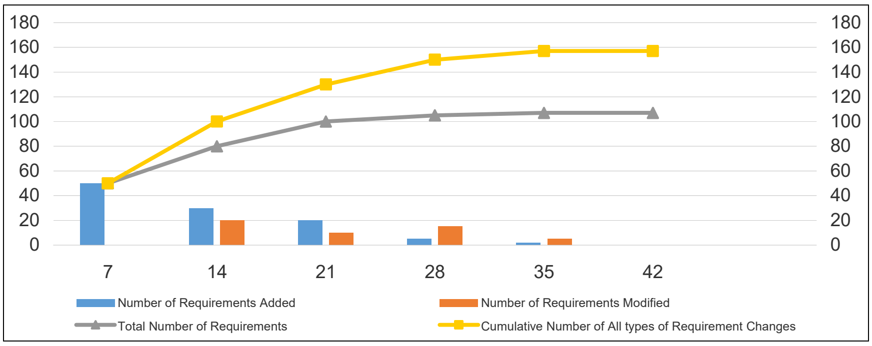


Figure 7 GPower – capturing new requirements (see online version for colours)



3.2 Observations on API usage – discrete data vs. streaming data APIs

Figure 8 shows the response size of a typical streaming-data API is around 10,000 times higher than that of a discrete-data API. Moreover, in a typical healthcare application, the usage of streaming-data APIs is comparatively much higher than that of discrete-data APIs usage. For instance, in VaaS application, it was observed that the response size of discrete-data APIs like that of APIs for heart rate, respiratory rate and temperature are of the order of 0.3 KB to 0.5 KB, whereas the response size of the streaming-data APIs like that of APIs for ECG, poincare and historical respiratory rate and historical heart rate is usually of the order of 2 MB to 3 MB. We also observed from the API gateway analytics, that the use of streaming-data APIs roughly contributes to 70% of the total APIs used, as depicted in Figure 9. Such observations stress the need of handling voluminous streaming data and thus any monolithic architecture would be virtually untenable for rapid healthcare applications.

Figure 8 Response size of discrete vs streaming APIs

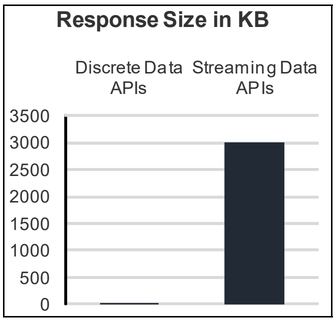
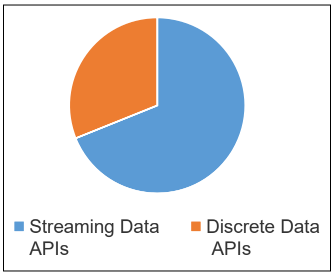


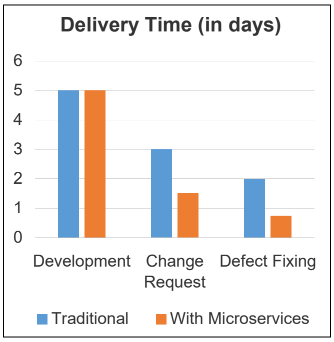
Figure 9 Usage % – discrete vs streaming APIs (see online version for colours)



3.3 Reduction in CR's and defect-fixing time with micro-services

Figure 10 presents that there is a significant amount of reduction in delivery time required for a change request or a defect fix when the micro-services-based approach is used. This reduction of time to address change requests or defects is achieved by restricting the rework only on the impacted services without invoking or investigating the effects of such changes on other services. As the services are individually deployable, unlike a monolithic application, isolation and adaptation of individual services without affecting all consumers are far easier.

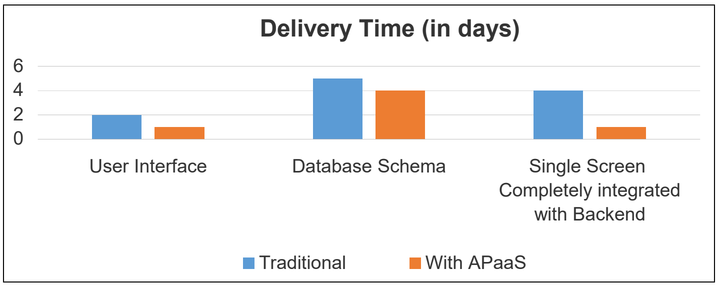
Figure 10 Implementation time – for development, CR and defect-fixing in traditional vs micro-services approaches (see online version for colours)



3.4 Accelerated development using APaaS

We also observed that there is a significant amount of reduction in delivery time required for designing user interfaces within such APaaS infrastructure. The platform itself is exposed as a separate service. Hence, integration of the interfaces to server side applications while satisfying the entire requirement set can be addressed to the minute details within a short span of time. Figure 11 presents some of the on field observations on development time cycle for traditional vis-à-vis APaaS-based approach.

Figure 11 Implementation time – for development of various modules compared against traditional and APaaS approach (see online version for colours)

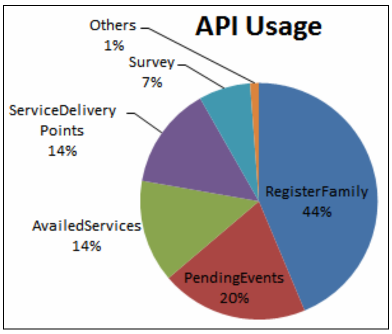


4 Business outcomes

4.1 Discovering operational insights by feature evaluation from API gateway

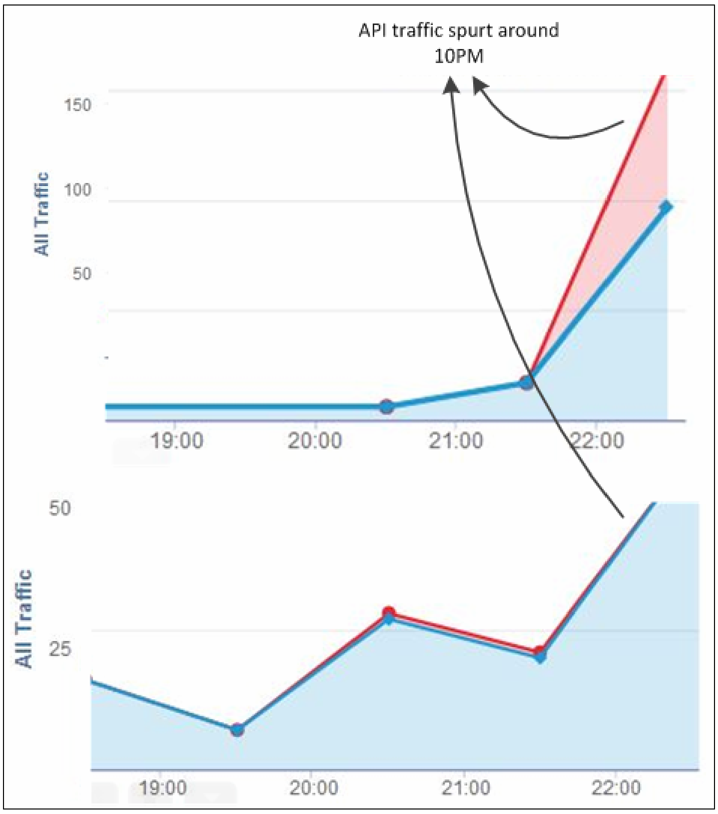
We gleaned several peculiar operational patterns through the usage statistics collected by the API gateway for the vision care management solution. For example, it was observed that during the initial phases of the trial, register family API was the most active one with about 44% of the traffic dedicated to this micro-service for registering a family into the mHealth system. Figure 12 shows a distribution of the traffic across the micro-services offered. Within the single register family API, 45% of the calls were for the family details record creation and 54% calls were for further updating the record.

Figure 12 API usage statistics (see online version for colours)



Another important pattern, as shown in Figure 13 is the spurt in activity after normal working hours observed repeatedly for multiple community health workers. This is an abnormal pattern, root cause for which is the lack of mobile handheld devices with some healthcare workers. These healthcare workers collected data using pen and paper, and only used to upload the data at a later point of time when a device was made available to them.

Figure 13 Abnormal traffic detected using non-working hours (see online version for colours)



Discovery of such operational patterns is crucial since it enables a straightforward codification of the patterns as a series of complex event processing rules and generate alerts as and when necessary without having to go through the API usage patterns every time a healthcare volunteer is using one of the exposed services.

4.2 Ascertain new business models

The rapid mHealth framework, as described above, enables the business leads and the product managers of the applications like VaaS to quickly able to spot the new deployment scenarios and the new business models. The micro-services-based architecture was able to protect the interests of all stakeholders within a single solution framework. Various business models are possible with the VaaS platform.

A set of simplified business models is enumerated in Table 2 and these models can serve as a starting points for deriving specific business cases based on specific hospital characteristics.

Table 2 Simplified business models

Model	*Location*	*Period of service*	*Data analysis frequency*	*Segment*	*Price level*	*Patient volume L = 1–50 M = 50–100 H = >100*
Membership	At kiosk	Annual	Weekly	Wellness	Low	High
Subscription	At home	Q1, Q2, Q3, or Q4	Fortnightly	Chronic diseases	High	Medium
SOS	At home	Week wise	Continuous	Post hospital	High	Medium
Premium	At clinics	Session	Continuous	Out patients	Low	Low
Patient care	At hospital	Daily	Continuous	In patients	Low	High
Research	At drug trails	Daily	Continuous	Clinical research	High	Medium
Health camps	Field	Annual	Weekly	Corporate social responsibility	Low	Low
Insurance	Field	Annual	Weekly	Risk cover	Low	High
Pharma	All	Annual	Weekly	Pharma	High	Medium

5 Limitations of the current work

The work presented in this paper provides an architectural framework to rapidly build mHealth applications through ingestion and processing of voluminous streaming data. The architecture has been successfully used in some of the practically deployed healthcare services in India. Still, further testing is essential when such a framework is used to build a large number of micro services integrated into a single application. In such a scenario, additional modules have to be built to handle micro services integrity, security and orchestration issues and a separate production monitoring mechanism is essential. The team communication overhead, while using non-uniform application technology stack is not discussed or covered in this paper.

5.1 Future scope of work

An important extension of the framework is to leverage the cloud computing infrastructure and capability. In general, application containers in the cloud make portability simple and easy. Containers help deliver better resource utilisation, easier configuration, faster deployment and more flexible development process. But, a single container may not be useful always. A container orchestration platform is to be created to deploy and manage multiple containers in a live system. A large healthcare application using micro service can consist of even dozens of containers that are spread across a number of physical nodes and independent containerised service. Without a container orchestration platform, scalability and bandwidth issues might become critical. Software orchestration implementation can be complex in itself, since it should cover other aspects like load balancing, performance monitoring, service discovery and micro-service segmentation. Our research is now moving from rapid development more towards autonomous, software-driven orchestration for efficiently managing application containers during the deployment phase. Findings and outcomes of this research may be presented in a succeeding paper.

6 Conclusions

Reigning in the costs of healthcare, while continuing to improve outcomes, challenges hospitals and providers everywhere – more so in growing economies. Even monitoring vital signs, the most basic indicators of patient status, counts for costly equipment that is often in short supply or needs repeated visits by overstretched healthcare professional taking spot measurements. Yet recognising and responding to early signs of patient deterioration have been shown to reduce overall morbidity and mortality, and continuous monitoring have been shown to reduce a patient's length of stay in the intensive care unit and total number of days in the hospital. Task shifting has been one of the key strategies followed by healthcare providers to tackle the health worker shortage, with technology playing a major enabling role. However, implementation of task shifting calls for an integrated framework that would enable rapid development and deployment of mobile healthcare solutions. Success of healthcare applications will be determined by the driving need, the usability and value of the solution and accessibility. Success should also be determined by the longevity of use and spread of the applications amongst many stakeholders. The research work presented here addresses the creation of a development framework specifically for healthcare needs. The trial deployments evaluated the proposed technology features. Technology interventions today have given healthcare agencies many tools and techniques to optimise the healthcare delivery in culturally diverse and low resource environments, with the added ability to evolve quickly while ensuring quality care for the patients. Results presented in this paper demonstrate that the proposed architecture is not only flexible, but can also help the applications to be deployed at scale while satisfying the interests of different stakeholders.

References

Carter, J. (2015) *Forget the Internet of Things, the Internet of Data is where its Really at* [online] http://www.in.techradar.com/news/world-of-tech/Forget-the-Internet-of-Things-ndash-the-Internet-of-Data-is-where-its-really-at/articleshow/48906388.cms (accessed 10 February 2016).

Dinodia Capital Advisors (2014) *Report: 'Indian Healthcare IT Sector – At the Cusp of Evolution*'.

Gartner IT Glossary (2012) [online] http://www.gartner.com/it-glossary/pace-layered-applicationstrategy.Ghosh (accessed 12 February 2016).

Ghosh, C., Maitra, A., Kuntagod, N. and Paul, S. (2015) 'Mitigating vulnerability of adolescent girls via innovative usage of digital technologies: insights from a field trial', *Proceedings of the 2nd International Conference on Science and Social Research*.

Health Planning and Infrastructure Division, Government of Australia (2010) *Paper: A Definition of a Rural Model for Health Service Delivery: A Hub and Spoke (Service Partner) Model*, Australia.

iSansys Report (2016) [online] http://www.isansys.com/en/services/vitals-as-a-service (accessed 10 February 2016).

Islam, S and Gregoire, J. (2010) 'Network edge intelligence for the emerging next-generation internet', *Future Internet*, Vol. 2, No. 4, pp.603–623

Kuntagod, N. and Maitra, A. (2013) 'A novel mobile application to assist maternal health workers in rural India', *Proceedings of the Software Engineering in Health Care (SEHC)*.

Kuntagod, N., Kumaresan, S. and Paul, S. (2015) 'Connected vision care for eradication of preventable blindness', *IEEE Medical Instruments and Measurements Conference*.

Lathar, P. Singh, Y. and Sharma, G. (2014) "Role of ANEKA cloud application platform in growing market', *Journal of Global Research in Computer Science*, Vol. 5, No. 6, pp.33–39.

Lee, S. Grover, K, and Lim, A. (2013) 'Enabling actionable analytics for mobile devices: performance issues of distributed analytics on Hadoop mobile clusters', *Journal of Cloud Computing: Advances, Systems and Applications*, Vol. 2, No. 1, pp.2–15.

Li, M., Lou, W. and Ren, K. (2010) 'Data security and privacy in wireless body area networks', *IEEE Wireless Communication*, Vol. 17, No. 1, pp.51–58.

Marz, N. and Warren, J. (2013) *Big Data: Principles and Best Practices of Scalable Real-time Data Systems*, April 2015, Manning Publications, ISBN 9781617290343.

NationMaster.com World Development Indicator Database (2014) *International Statistics* [online] http://www.nationmaster.com/country-info/stats/Health/Hospital-beds/Per-1,000-people (accessed 10 February 2016).

Sharma, S. (2015) *Evolution of as-a-Service Era in Cloud*, arXiv preprint arXiv:1507.00939, Center for Survey Statistics and Methodology, Iowa State University, Ames, Iowa, USA.

Sharma, S. (2016) 'Expanded cloud plumes hiding big data ecosystem', *Future Generation Computer Systems*, June 2016, Vol. 59, pp.63–92.

Sharma, S. et al. (2014) 'A brief review on leading big data models', *Data Science Journal*, Vol. 13, pp.138–157, DOI: http://doi.org/10.2481/dsj.14-041.

Sironi, G (2011) *DZone Article: Pretotyping: A Complete Example* [online] https://dzone.com/articles/pretotyping-complete-example (accessed 10 February 2016).

Strang, T. (2003) *Towards Autonomous Context Aware Services for Smart Mobile Devices*, pp.279–292, LNCS 2574, Springer, Berlin, Heidelberg.

Toninelli, A. Montanari, R. and Corradi, A. (2009) 'Enabling secure service discovery in mobile healthcare enterprise networks', *IEEE Wireless Communication*, Vol. 16, No. 3, pp.24–32.

Varshney, U. (2007) 'Pervasive healthcare and wireless health monitoring', *Springer, Mobile Network Application*, Vol. 12, Nos. 1/3, pp.113–127.

Yuce, M., Ng, S., Myo, N., Khan, J. and Liu, W. (2007) 'Wireless body sensor network using medical implant band', *Journal of Medical Systems*, Vol. 31, No. 6, pp.467–474.