Enabling Real-Time Mobile Cloud Computing through Emerging Technologies

Tolga Soyata
University of Rochester, USA

A volume in the Advances in Wireless Technologies and Telecommunication (AWTT) Book Series

Information Science REFERENCE
An Imprint of IGI Global

Managing Director: Lindsay Johnston
Managing Editor: Austin DeMarco
Director of Intellectual Property & Contracts: Jan Travers
Acquisitions Editor: Kayla Wolfe
Production Editor: Christina Henning
Development Editor: Brandon Carbaugh
Cover Design: Jason Mull

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA, USA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Enabling real-time mobile cloud computing through emerging technologies / Tolga Soyata, editor.
pages cm
Includes bibliographical references and index.
ISBN 978-1-4666-8662-5 (hc) -- ISBN 978-1-4666-8663-2 (eISBN) 1. Cloud computing. 2. Mobile computing. I. Soyata, Tolga, 1967-
QA76.585.E55 2015
004.67'82--dc23

2015015533

This book is published in the IGI Global book series Advances in Wireless Technologies and Telecommunication (AWTT) (ISSN: 2327-3305; eISSN: 2327-3313)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter 10
Operational Cost of Running Real-Time Mobile Cloud Applications

Ovunc Kocabas
University of Rochester, USA

Regina Gyampoh-Vidogah
Independent Researcher, UK

Tolga Soyata
University of Rochester, USA

ABSTRACT

This chapter describes the concepts and cost models used for determining the cost of providing cloud services to mobile applications using different pricing models. Two recently implemented mobile-cloud applications are studied in terms of both the cost of providing such services by the cloud operator, and the cost of operating them by the cloud user. Computing resource requirements of both applications are identified and worksheets are presented to demonstrate how businesses can estimate the operational cost of implementing such real-time mobile cloud applications at a large scale, as well as how much cloud operators can profit from providing resources for these applications. In addition, the nature of available service level agreements (SLA) and the importance of quality of service (QoS) specifications within these SLAs are emphasized and explained for mobile cloud application deployment.

INTRODUCTION

Cloud is the platform of multiple servers over a widely disbursed geographic area, connected by the Internet for the purpose of serving data or computation (Bansal, 2013). Mobile Cloud Computing (MCC) can be described as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (Shanklin, 2014) from mobile devices. MCC therefore refers to both the applications delivered as services over the Internet and the hardware and system software in data-

DOI: 10.4018/978-1-4666-8662-5.ch010

centers that provide those services. The services themselves have long been referred to as Software as a Service (SaaS). Some vendors use terms such as Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) and others to describe their products, but we abstain from these because accepted definitions for them still differ widely. The datacenter hardware and software is what we will call a *cloud*. When a cloud is made available in paying costs as they occur to the general public, it is called a *public cloud* and the service being sold is *utility computing. Private cloud* on the other hand refers to internal data centers of a business or other organization (Armbrust, et al., 2010).

The point at which these internal data centers are large enough to enable organizations to benefit from the advantages of cloud computing are the subject of much debate. (Kovachev, Cao, & Klamma, 2013) described cloud computing as the sum of SaaS and utility computing, but does not include small or medium-sized datacenters, though some of these rely on virtualization for management. People can be users or providers of SaaS, or utility computing. The focus here is on SaaS providers (cloud users) and cloud providers, who have received less attention than SaaS users. Mobile computing is the delivery of services, software and processing capacity over the Internet, reducing cost, increasing storage, automating systems, decoupling of service delivery from underlying technology, and providing flexibility and mobility of information. However, the actual realization of these benefits is far from being achieved for mobile applications (Kovachev, Cao, & Klamma, 2013).

MCC is introduced as an integration of cloud computing into the mobile environment. It is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (Mouftah & Kantarci, 2013). MCC is a model for transparent elastic augmentation of mobile devices via ubiquitous wireless access to cloud storage and computing resources, with context-awareness and dynamic adjusting of offloading in respect to change in operating conditions, while preserving available sensing and interactive capabilities of mobile devices (Fesehaye, Gao, Nahrstedt, & Wang, 2012).

However, there has been some confusion about the mobile cloud computing model about its capabilities and can sometimes be described in general terms that includes almost any kind of outsourcing of hosting and computing resources. In other words, mobile represents a relatively new and fast growing segment of the cloud-computing paradigm (Rimal & Choi, 2012)

In view of the inherent advantages of this technology, enterprises today are looking to cloud computing to help them better deliver existing as well as new, innovative services on demand across network, computing, and storage resources at reduced cost (Chappel, 2013). This is because cloud economics will play a vital role in shaping the mobile cloud industry of the future (IBM, 2013). In a recent (Microsoft, 2010) white paper titled "Economics of the Cloud", it stated that the mobile computing industry is moving towards the cloud driven by three important economies of scale because: 1) large data centers can deploy computational resources at significantly lower costs than smaller ones; 2) demand pooling improves utilization of resources; and 3) multi-tenancy lowers application maintenance and labor costs for large public clouds. The cloud also provides an opportunity for IT professionals to focus more on technological innovation rather than thinking of the budget of "waiting to force things to move." However, many organizations find it difficult to determine the total operating costs of using cloud services (Microsoft, 2010).

The recent survey conducted by (Prasad, Gyani, & Murti, 2012) supports this view and it was revealed that, user and potential users of mobile cloud services would reduce costs and time needed to deploy tools for quicker analysis and planning. 14% somewhat disagreed and in helping to improve planning and performance management in users' organizations, 42% believed they would be helped by use of mobile cloud computing for rapid deployment of specific applications. Respondents also favored cloud usage.

Those with the tendency to use the cloud-based finance applications, 36% had already deployed one or more applications and 23% had deployments in process. Another 28% are considering deployment of one or more cloud-based applications. 13% of all respondents have decided to forgo on-premise to considering cloud-based applications.

These findings suggests that, the key idea revealed concerning mobile cloud computing is to: (1) understand how cloud applications can provide data center operators with greater reach and more rapid deployment; (2) identify and understand how cloud computing can change and reduce current cost; and (3) making sure collaboration with application vendors and enterprises are key to developing applications that fit into each organization's strategic vision for more cost-effective options (Dinh, Lee, Niyato, & Wang, 2013). The chapter briefly describes some of the cost models that have been developed, how this has fed into pricing models of cloud service providers and illustrates how these models can be used to estimate the true operational cost of real-time mobile cloud computing services. The aim is to help decision makers make the business case for specific applications and what services matches the business case.

CLOUD OPERATOR PRICING

In this chapter, a comparison of pricing of major cloud service providers will be provided. We will look at Microsoft Azure, Google Cloud Platform and Amazon Web Services (AWS). For each provider the pricing will be based on the US East region. Operating system is assumed to be Linux.

Microsoft Azure Pricing

Virtual machine lists are based on their properties. Table 1 and Table 2 demonstrate the pricing for basic and standard tiers, which provide all possible machine (instance) configurations. Compute intensive Instances A8 and A9 contain the Intel Xeon E5-2670 CPU @2.6 GHz. Microsoft Azure offers pricing for two tiers: basic and standard. While both tiers are similar in terms of virtual machine configurations, standard tier offers additional capabilities such as load balancing and auto-scaling for performance improvement. These tiers are the data for virtual machines with SSD storage. Storage prices are for persistent storage and billed by GB per month. Table 3 tabulates the storage pricing when the data is stored with 99.9% read/write availability SLA (MS-SLA).

Data transfers are charged based on the direction to Azure data centers. Inbound data transfers, which are the data transfers into the data centers are free of charge. Outbound data transfers are charged based on the data amount shown in Table 4.

Google Cloud Platform Pricing

Google App Engine for PaaS services and Compute Engine for IaaS services are shown in Table 5. Google Compute Engine billing includes a minimum of 10 minutes of usage. After this 10-minute minimum interval, each instance is charged at 1-minute increments (rounded up to the nearest minute). Additionally, Google offers discounts based on sustained (continuous) usage. The f1-micro and g1-small instances are for non resource-intensive tasks that remain active for long periods of time.

Google charges network differently for networking inside the cloud infrastructure and outside communication via the Internet as shown in Table 6. Incoming data into the cloud services and outgoing

Table 1. Basic Tier Pricing for A and D series of virtual machines

Instance Type	Cores	RAM (GB)	Disk Size (GB)	Price (per Hr)
A0	1	0.75	20	$0.018
A1	1	1.75	40	$0.044
A2	2	3.5	60	$0.088
A3	4	7	120	$0.176
A4	8	14	240	$0.352
D1	1	3.5	10	$0.077
D2	2	7	40	$0.154
D3	4	14	100	$0.308
D4	8	28	200	$0.616
D11	2	14	40	$0.195
D12	4	28	100	$0.390
D13	8	56	200	$0.702

Table 2. Microsoft Azure Standard Tier Pricing for A and D series of virtual machines

Instance Type	Cores	RAM (GB)	Disk Size (GB)	Price (per Hr)
A0	1	0.75	20	$0.020
A1	1	1.75	70	$0.060
A2	2	3.5	135	$0.120
A3	4	7	285	$0.240
A4	8	14	605	$0.480
A5	2	14	135	$0.250
A6	4	28	285	$0.500
A7	8	56	605	$1.000
A8	8	56	382	$1.970
A9	16	112	382	$4.470
D1	1	3.5	50	$0.094
D2	2	7	100	$0.188
D3	4	14	250	$0.376
D4	8	28	500	$0.752
D11	2	14	100	$0.238
D12	4	28	200	$0.476
D13	8	56	400	$0.857
D14	16	112	800	$1.542

Table 3. Microsoft Azure Storage Pricing

Storage Capacity (TB per Month)	Price (per GB)
0 – 1	$0.024
1 – 50	$0.0236
50 – 500	$0.0232
500 – 1000	$0.0228
1000 – 5000	$0.0224

Table 4. Microsoft Azure Data Transfer Pricing

Outbound Data Transfers (per Month)	Pricing (per GB)
First 5 GB	Free
5 GB – 10 TB	$0.087
Next 40 TB	$0.083
Next 100 TB	$0.07
Next 350 TB	$0.05

Table 5. Google Compute Engine Pricing

Instance Type	Cores	Memory (GB)	Price	Discounted (25% - 50%)	Discounted (50% - 75%)	Discounted (75% - 100%)
n1-standard-1	1	3.75	$0.063	$0.050	$0.038	$0.025
n1-standard-2	2	7.5	$0.126	$0.101	$0.076	$0.050
n1-standard-4	4	15	$0.252	$0.202	$0.151	$0.101
n1-standard-8	8	30	$0.504	$0.403	$0.302	$0.202
n1-standard-16	16	60	$1.008	$0.806	$0.605	$0.403
n1-highmem-2	2	13	$0.148	$0.118	$0.089	$0.059
n1-highmem-4	4	26	$0.296	$0.237	$0.178	$0.118
n1-highmem-8	8	52	$0.592	$0.474	$0.355	$0.237
n1-highmem-16	16	104	$1.184	$0.947	$0.710	$0.474
n1-highcpu-2	2	1.8	$0.080	$0.064	$0.048	$0.032
n1-highcpu-4	4	3.6	$0.160	$0.128	$0.096	$0.064
n1-highcpu-8	8	7.2	$0.320	$0.256	$0.192	$0.128
n1-highcpu-16	16	14.4	$0.640	$0.512	$0.384	$0.256
f1-micro	1	0.6	$0.012	$0.010	$0.007	$0.005
g1-small	1	1.7	$0.032	$0.026	$0.019	$0.013

Table 6. Google Cloud Platform Network Pricing

Outbound Data Transfers (per GB)	Cost
0 – 1 TB	$0.12
1 – 10 TB	$0.11
> 10 TB	$0.08

data for same zone or to a different cloud service in the same region are free. Alternatively, outgoing data to a different Zone in the same Region or different Region within the US are charged at $0.010 per GB of data movement. Finally storage costs are $0.04 per GB per month for standard disks and $0.325 per GB per month for SSD disks.

Amazon Web Services (AWS) Pricing

The cost of running applications in AWS is determined by three factors: computation, storage and data transfer. We use AWS Elastic Compute Cloud (EC2) instances to calculate computation cost as shown in Table 7. AWS Simple Storage Service (S3) pricing shown in Table 8 is used to calculate the storage costs. Data transfer cost is included for only outgoing data transfers from EC2 and S3 to the Internet as shown in Table 9. Incoming data traffic is free of charge. EC2 instance types and their configurations are detailed in Table 7: EC2 instances are grouped into categories based on their capabilities. t2 and m3 instances are for general purpose applications. c3 instances are optimized for computation with high-performance processors. g2 instances contain GPUs for graphics and general purpose GPU applications. r3 instances are optimized for memory-intensive applications and provide lowest price per GB of RAM. i2 and hs1 instances are optimized for storage and provide lowest price per GB of storage on the instance. The pricing of EC2 instances depends on the usage as shown in Table 7. On demand instances are charged by the hour with no commitments. Reserved instances are charged a one-time upfront fee but in return provide a lower per-hour cost. Reserved instances require a commitment and differ in commitment duration (1 year to 3 years) and utilization (light, medium and heavy). Additionally, EC2 provides Spot Instances, which can be purchased by bidding on unused EC2 instances. Pricing of Spot Instances is set by EC2 and may change based on availability of Spot Instances.

CLOUD OPERATOR SERVICE LEVEL AGREEMENTS

In all instances, with the information provided by service providers, the nature and content of service level agreement can influence the choice of the provider. A Service Level Agreement (SLA) is used as a formal contract between the service provider and a consumer to ensure service quality (Wu, 2014). An SLA should specify the details of the service usually in quantifiable terms. The goal of an SLA is therefore to establish a scalable and automatic management framework that can adapt to dynamic and real time environmental changes using multiple qualities of service (QoS) parameters. The SLA for mobile cloud computing should have terms that include multiple domains with heterogeneous resources. In addition, consumers should be involved in the management process of SLA to a certain extent especially regarding reliability and trust/security. In particular, QoS parameters must be updated dynamically over time due to the continuing changes in the mobile application operating environments (Kwon, et al., 2014).

Table 7. AWS EC2 Instance Pricing

Instance Name	Cores	RAM (GB)	Storage (GB)	Processor Type	On Demand Rate	Reserved Upfront Fee (3 Years)	Hourly Rate (3 Years)
t2.micro	1	1	-	Xeon Family	$0.013	$109	$0.002
t2.small	1	2	-	Xeon Family	$0.026	$218	$0.004
t2.medium	2	4	-	Xeon Family	$0.052	$436	$0.008
m3.medium	1	3.75	1 x 4 SSD	Xeon E5-2670	$0.070	$337	$0.015
m3.large	2	7.5	1 x 32 SSD	Xeon E5-2670	$0.140	$673	$0.030
m3.xlarge	4	15	2 x 40 SSD	Xeon E5-2670	$0.280	$1345	$0.060
m3.2xlarge	8	30	2 x 80 SSD	Xeon E5-2670	$0.560	$2691	$0.120
c3.large	2	3.75	2 x 16 SSD	Xeon E5-2680	$0.105	$508	$0.022
c3.xlarge	4	7.5	2 x 40 SSD	Xeon E5-2680	$0.210	$1016	$0.045
c3.2xlarge	8	15	2 x 80 SSD	Xeon E5-2680	$0.420	$2031	$0.090
c3.4xlarge	16	30	2 x 160 SSD	Xeon E5-2680	$0.840	$4063	$0.180
c3.8xlarge	32	60	2 x 320 SSD	Xeon E5-2680	$1.680	$8126	$0.359
g2.2xlarge	8	15	60 SSD	Xeon E5-2670	$0.650	$6307	$0.060
r3.large	2	15.25	1 x 32 SSD	Xeon E5-2670	$0.175	$1033	$0.026
r3.xlarge	4	30.5	1 x 80 SSD	Xeon E5-2670	$0.350	$2066	$0.052
r3.2xlarge	8	61	1 x 160 SSD	Xeon E5-2670	$0.700	$4132	$0.104
r3.4xlarge	16	122	1 x 320 SSD	Xeon E5-2670	$1.400	$8264	$0.208
r3.8xlarge	32	244	2 x 320 SSD	Xeon E5-2670	$2.800	$16528	$0.416
i2.xlarge	4	30.5	1 x 800 SSD	Xeon E5-2670	$0.853	$2740	$0.121
i2.2xlarge	8	61	2 x 800 SSD	Xeon E5-2670	$1.705	$5480	$0.241
i2.4xlarge	16	122	4 x 800 SSD	Xeon E5-2670	$3.410	$10960	$0.482
i2.8xlarge	32	244	8 x 800 SSD	Xeon E5-2670	$6.820	$21920	$0.964
hs1.8xlarge	16	117	24 x 2048	Xeon Family	$4.600	$16924	$0.76

Table 8. AWS S3 Storage Pricing

Storage Size (TB)	Price (per GB)
0 – 1	$0.03
1 – 50	$0.0295
50 – 500	$0.029
500 – 1000	$0.0285
1000 – 5000	$0.0280
> 5000	$0.0275

Table 9. AWS Data Transfer Pricing

Data Transfer Out Size (TB)	Price (per GB)
0 – 1 (GB)	$0.00
1 TB – 10 TB	$0.12
10 TB – 50 TB	$0.09
50 TB – 150 TB	$0.07
150 TB – 500 TB	$0.05

Amazon and Microsoft Azure offer a tiered service credit plan that gives users credits based on the discrepancy between SLA specifications and the actual service levels delivered. These providers typically offer cloud storage SLA that articulates precise levels of service such as 99.9% uptime and recourse or compensation to the user should the provider fail to provide the service as described. Another normal cloud storage SLA detail is service availability, which specifies the maximum amount of time a read request can take, how many retries are allowed and so on.

Microsoft Azure SLA

Azure Active Directory Premium service is available in the following scenarios such as: Users are able to login to the service, login to access applications on the access panel and reset passwords. IT administrators are able to create, read, write and delete entries in the directory or provision or de-provision users to applications in the directory. Windows Azure has separate SLAs for compute and storage. No SLA is provided for free tier of Azure Active Directory. Availability is calculated over a monthly billing cycle (Azure, 2014).

Google SLA

Google has implemented industry standard systems and procedures for cloud SLA to ensure the security and confidentiality of an application and customer data, protect against anticipated threats or hazards to the security or integrity of an application and customer data, and protect against unauthorized access. Customers have the ability to access, monitor, and use or disclose their data submitted by end users through the service. Customer data and applications can only be used to provide the services to customer and its end users and to help secure and improve the services. For instance, this may include identifying and fixing problems in the services, enhancing the services to better protect against attacks and abuse, and making suggestions aimed at improving performance or reducing costs (Google, 2013).

Amazon EC2 SLA

Amazon AWS SLA policy governs the use of Amazon Elastic Compute Cloud. This SLA applies separately to each account using Amazon EC2 or Amazon EBS. AWS is committed to using commercially reasonable efforts to make Amazon EC2 and Amazon EBS each available with a monthly uptime percentage of at least 99.95%, in each case during any monthly billing cycle. The monthly uptime percentage is less than 99.95% but equal to or greater than 99.0% and service credit of 10% and less than 30% (Amazon EC2, 2013).

SECURITY OF CLOUD OUTSOURCING

Security issues should be considered when computational tasks are being outsourced for mobile cloud applications especially regarding transmission and data receipt (Soyata, Ba, Heinzelman, Kwon, & Shi, 2013). For this reason, placing critical data in the cloud in the hands of a third party to ensure the data remains secured is of paramount importance. This means the data need to be encrypted at all times with clearly defined roles of who manages encryption keys. The data in the cloud needs to be accessible only

by authorized users. That is making it restricted and monitoring of who accesses what data through the cloud. This is because protecting clients' data is essential in order to ensure data is not compromised, due to breach or disaster. In this instance the application developers should encrypt the data leaving the backup services to cloud service providers.

CASE STUDY A: CLOUD-BASED HEALTH MONITORING SYSTEM

Automating health monitoring has become significant because of the drive by the US government to modernize the US health system using cloud computing based medical applications. (Kocabas, et al., 2013). (Kocabas & Soyata, 2014) emphasized that, while one motivation for this is to reduce operational costs at the healthcare organization (HCO) by eradicating the datacenters managed by the HCO, an equally important motivation is to improve healthcare by providing the doctors and health professionals with long-term patient data as an auxiliary diagnosis tool (Page, Kocabas, Soyata, Aktas, & Couderc, 2014). This section illustrates how cloud-pricing models can be used to estimate the cost of real-time cloud computing applications conducted in (Kocabas, et al., 2013). To do this, background information on the applications included in the study of long-term patient ECG-data monitoring system are reviewed.

Mobile Cloud Application Description

Despite the undeniable transformation cloud computing made in the application world, medical applications lack the pace of adopting the trend. Due to the Health Insurance Portability and Accountability Act (HIPAA) (HIPAA, n.d.), Personal Health Information (PHI) privacy is treated as the most sensitive information and the penalties associated with its mistreatment are steep. Storage of encrypted PHI through a Business Associate Agreement (BAA) (US-HHS, n.d.) is currently available from cloud operators, such as CareCloud (CareCloud, n.d.). This service is for *storage-only* data and no *medical application* can be executed on this data, as this would require the data to be temporarily transformed into the *unencrypted domain*.

A novel method for running medical cloud applications in the *encrypted domain* has been proposed in (Kocabas, et al., 2013), (Kocabas & Soyata, 2014) that executes applications in a way where the underlying patient PHI is not visible to the cloud during execution, thereby completely eliminating PHI privacy concerns. The method uses Fully Homomorphic Encryption (FHE), which is a type of encryption that allows operations on encrypted data without observing the data itself. While using FHE solves the privacy issues regarding medical cloud computing, one problem prevents its wide adoption: the FHE is still extremely resource-intensive in terms of storage, bandwidth and computational requirements.

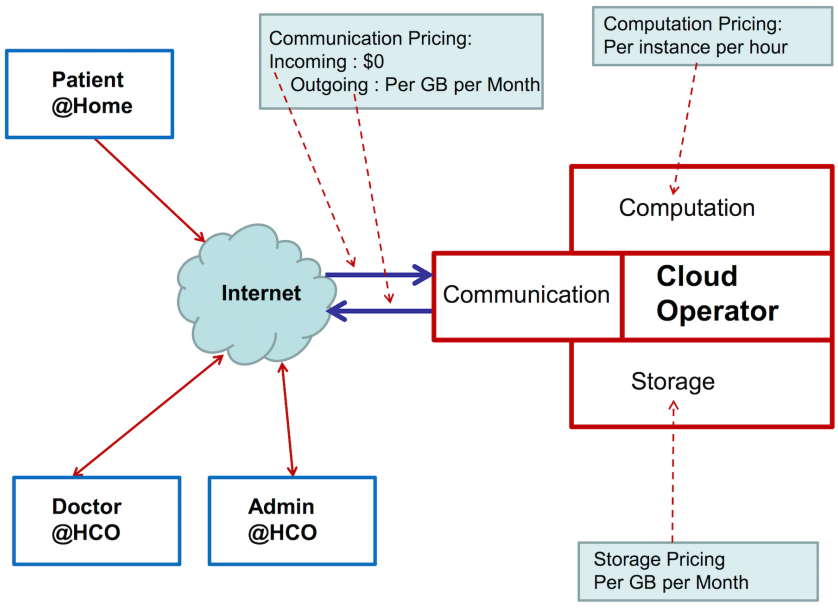
The medical application will be used for monitoring patients remotely at their home and providing patient statistical data to the HCO personnel. For the remote monitoring application, detecting Long-QT Syndrome (LQTS) will be used as the target application. Additionally, vital patient health statistics such as average heart rate (HR), minimum and maximum HR will be calculated.

In this section, background information is provided on the cloud application platform and the related outsourcing costs. Figure 1 depicts the conceptualized medical application platform, which is intended to allow patient health monitoring at home through medical data acquisition devices such as ECG patches, or multi-sensory acquisition devices. The acquired patient data is assumed to be encrypted at home (top of Figure 1) and transmitted to the cloud (middle of Figure 1). To ensure PHI privacy, FHE

is used, which requires PHI encryption at home. After this encryption, the data is unrecognizable to the cloud. Note that, this information transmission is one-way. On the bottom of Figure 1, the HCO personnel and HCO administrators access the application and transmit their preferences (i.e., two-way). The medical applications run in the cloud, which operate on the encrypted PHI without ever observing the actual medical data. Compared to conventional cloud applications, the significant difference of this application scenario is its resource intensity. A *resource* is defined as one of the three entities that a cloud operator charges for: a) storage, b) computation, and c) communication. Each of these resources will be elaborated on separately and a cost analysis of the medical cloud application will be analyzed based on these resources.

- **Storage:** This most widely utilized resource involves renting a shared storage space from the cloud operator based on the previously provided pricing. The pricing of this resource varies widely based on the allocated space as shown in, for example, Table 3 (Microsoft Azure) and Table 8 (AWS S3 Storage). The most meaningful product to purchase for an HCO is based on per-GB-per-month pricing. As we will describe later, the medical applications have substantial storage requirements to the host the FHE-encrypted PHI.
- **Computation:** The execution of the medical algorithms based on FHE put a substantial strain on cloud computational resources. While it is possible to cut costs by using shared computational instances, it makes a lot more sense to use dedicated *boxes* (i.e., computers) to perform the type of computations. The pricing for these instances is based on per-hour usage of that instance. Examples of computation pricing have been provided in Table 2 (Microsoft Azure), Table 5 (Google Compute Engine), and Table 7 (AWS EC2).

Figure 1. Conceptualized medical application environment and related cloud-pricing metrics. (HCO = Health Care Organization)



- **Communication:** This resource involves data traffic from and into the cloud operator. Usually cloud providers have no incoming-data costs but, a per-GB-per-month outgoing-data cost, as shown in Figure 1. Examples of pricing are provided previously in Table 4 (Microsoft Azure), Table 6 (Google Cloud Platform), Table 9 (Amazon AWS).

Cloud Outsourcing the Health Monitoring Application

While it is possible to encrypt private health information (PHI) through standard encryption techniques, such as AES (NIST-AES, 2001), such data cannot be used as an input into a medical application without decrypting it. However, decrypting PHI in the cloud even temporarily violates PHI privacy. Therefore, a medical application using a standard encryption such as AES cannot use the cloud for running a medical application. Currently, there exists no offering from any cloud operator that allows running a medical application without violating PHI privacy.

Fully Homomorphic Encryption (FHE) is a type of encryption that allows operations on encrypted data without observing the data itself. While using FHE sounds promising, it puts significant strain on all three types of resources described earlier. We will describe the reasons for this in this section and will evaluate the application in terms of its usage of all three resources:

- **Storage:** The target medical application acquires patient vital information at home for long term monitoring and diagnosis via monitoring devices such as Alivecor iPhone attachment (AliveCor, n.d.). For the FHE-based applications to work, the data has to be homomorphically-encrypted at the source (i.e., patient's home) via a computationally capable device, such as a cloudlet (Soyata, Muraleedharan, Funai, Kwon, & Heinzelman, 2012). Once this is done, the size of the data expands multiple orders of magnitude. To cope with this problem, the data is processed partially before encryption. For example, in the case of a cardiac monitoring, essential values, such as QT and RR intervals, can be extracted from ECG signals through simple algorithms (Couderc, et al., 2011). Note that we focus on the resource requirements of such algorithms, and the algorithms are beyond the scope of our study. Despite the reduction of the data after processing with the described algorithms (Couderc, et al., 2011), the size of the data still places an extreme pressure on the storage resources as will be quantified later.
- **Computation:** In addition to expanding the relevant data significantly, the FHE-based algorithms also require substantial amount of computational resources. One distinction between Storage and Computation, though, is that, while storage must always be occupied, computation resources can be shared within the healthcare organization's (HCO) dedicated computer, thereby allowing significant cost reduction. A patient health monitoring is only necessary during the time-of-monitoring, e.g., 2 weeks out of the entire year, lending itself to reduced costs due to this low duty cycle.
- **Communication:** While the amount of data being transmitted *into* the cloud is substantial (see Figure 1), this transfer is zero cost for cloud operators that do not charge for incoming bandwidth such as AWS as described previously. When the operations are complete, the amount of data being transmitted *out of the cloud* is a lot less than the incoming data, thereby making this resource the least expensive to rent for these types of applications. An insight can be gained into this through an example: Assume that, a patient's ECG data is being transmitted continuously at one beat per second (corresponding to a heart rate of 60). The result of an entire monitoring period (e.g., 5 minutes) is a Yes/No answer, which is, clearly, a lot less data than sum of all of the original ECG samples.

We evaluate the cost of cloud outsourcing of medical applications that use FHE by using an ECG database named THEW (Couderc, 2010). We choose long QT syndrome (LQTS) (Bazzett, 1997) detection as the primary medical application that is widely used for long-term patient monitoring. Additional applications that provide vital patient health information such as Average Heart Rate, Minimum and Maximum Heart Rate are also implemented. All of the medical applications will operate on the same THEW sample data and provide statistics for every 5-minute ECG monitoring period. We implement these applications with the open-source HElib library (Halevi & Shoup) and run our simulations on a cluster node with two Intel Xeon E5-2695 Processors and 63GB RAM. The parameters of HElib are selected based on the analysis provided in (Gentry, Halevi, & Smart, 2012).

Operational Cost Worksheet for Case A

Resource requirements of the aforementioned medical applications (i.e., LQTS detection, average heart rate computation, and min/max heart rate computation) are reported in **Table 10**. We estimate the cost of running these medical applications using Microsoft Azure, Google Compute Engine and Amazon Web Services in the following sections.

Google Compute Engine

We select a Google Compute Engine machine type based on the type of medical computation to be performed. LQTS and Average Heart Rate (HR) are less compute-intensive compared to Minimum and Maximum HR computation. We select an *n1-standard-2* machine type with 2 CPUs to compute LQTS and Average HR. For the Minimum and Maximum HR we select *n1-highcpu-2* machine with 2 CPUs. Based on the duty cycle information provided in Table 10, we use appropriate sustained discount price provided by Google Compute Engine. The total cost of running these computations are reported in Table 11. The sum of the application costs per patient is $81.12 per month for computing all four values (LQTS, average HR, min HR, max HR).

Table 10. Performance of Medical Applications on a Cluster Node over a period of one month.

Medical Application	Run-Time (Hours)	Duty Cycle (%)	Storage (GB)	Data Transfer Out (GB)
LQTS	51.3	7	75.3	322.6
Average HR	197.7	27	252.9	1166.4
Min. & Max. HR	483.2	66	266.5	1229.7

Table 11. Monthly Cost of Running Medical Applications with Google Compute Engine

Medical Application	GCE Instance	GCE Cost	Storage Cost	Transfer Cost	Total Cost
LQTS, Avg. HR	*n1-standard-2*	20.00	13.12	5.32	$38.44
Min. & Max. HR	*n1-highcpu-2*	23.23	10.66	8.79	$42.68

Amazon Web Services (AWS)

For the same computations, to select an EC2 instance type, we look at our application characteristics. LQTS and Average Heart Rate (HR) computation is less compute-intensive compared to Minimum and Maximum HR. A general-purpose instance of *m3.large* with 2 CPU's is enough to compute both LQTS and Average HR in parallel. On the other hand Minimum and Maximum HR are compute-intensive with 67% duty cycle. Therefore a *c3.large* instance optimized for compute-intensive applications could compute both Minimum and Maximum HR in parallel.

Table 12 presents the cost of outsourcing one patient's monitoring to AWS for a month. EC2 costs include both a fixed cost fee and run-time fee. Fixed cost fee is calculated by dividing one-time upfront fee into equal monthly payments. Run-time fee is calculated by multiplying the run-time of the applications with the hourly rate. Since two applications are executed in parallel, we select application run-time as the runtime of the application that takes longer to finish. S3 costs are the storage costs derived from the application storage requirements presented in Table 8. Minimum and Maximum HR computation use same data set, and therefore their storage amount is counted only once. The transfer cost is based on the data transferred out to a doctor at the Health Care Organization from AWS. The overall cost of running all the applications per patient with AWS services is $96 per month.

Microsoft Azure

Based on the application characteristics, we choose *D2* instance for LQTS and Average HR computation and *D11* instance for Minimum and Maximum HR computation. Since LQTS and Average HR is less compute-intensive *D2* instance with 2 cores and 7 GB RAM is enough to compute them in parallel. *D11* instance with 2 cores and 14 GB RAM is chosen for more compute-intensive Minimum and Maximum HR in parallel. Table 13 shows the cost of running the same applications on Microsoft Azure, with a total monthly cost of $148.50 based on the same parameters shown in Table 10.

Table 12. Monthly Cost of Running Medical Applications with AWS Services

Medical Application	EC2 Instance	EC2 Cost	S3 Cost	Transfer Cost	Total Cost
LQTS, Avg. HR	*m3.large*	24.63	9.84	4.20	$38.68
Min. & Max. HR	*c3.large*	43.15	7.86	6.31	$57.32

Table 13. Monthly Cost of Running Medical Applications with Microsoft Azure

Medical Application	Azure VM Type	VM Cost	Storage Cost	Transfer Cost	Total Cost
LQTS, Avg. HR	D2	$30.5	$7.9	$3.4	$41.8
Min. & Max. H	D11	$94.4	$6.4	$5.9	$106.7

CASE STUDY B: REAL-TIME FACE RECOGNITION USING MOCHA

Face recognition is the process of identifying a person by matching the person's facial features to a database that keeps the records of features of many people. This process involves mainly three steps: face detection, projection and search. During the face detection step, the face of a person is detected from an image. Then in the projection step, related features are extracted from the detected face. The related features vary with the methods used for projection. Finally the features are used for matching the face with an entry in the database during the search step. We will explain each step in detail and analyze the complexity and resource requirements in the following sections.

Computational Requirements of Face Recognition

- **Face Detection:** We will use the Viola-Jones method (Viola & Jones, 2001) for face detection, which is one of the fastest and most accurate algorithms for face detection. The Viola-Jones method is the first face detection algorithm that operates in real-time and widely used in mobile devices and cameras. The method detects faces by partitioning the image into rectangular sections and checking the change of intensity between these sections. Each rectangular section, also known as Haar-like feature, contains sum of the pixel intensities covered by the rectangular section. By using the pixel intensities, a face is detected by checking the changes of intensity between the rectangular sections. Complexity of the face detection is depends on the number of pixels in an image. In (Viola & Jones, 2001), it is shown that the number of faces in one image has negligible effect on the complexity of face detection due to nature of classifiers used in Viola-Jones method. The approximate time for face detection is approximated as follows (Alling, Powers, & Soyata, 2015):

$$T_{FD}=n_{pixels}\times 0.01347\times GFLOPS^{-0.622726}$$

where T_{FD} represents the detection time in milliseconds, n_{pixels} is the number of pixels in the image and GFLOPS is the maximum number of floating point operations that a processor can execute per second. Face detection can be performed with the surveillance device (e.g., mobile phone, tablet, surveillance camera) in real-time using Viola-Jones method. Even better performance results can be achieved using a cloudlet that has significant computation power with reasonable cost. For example in (Soyata, Muraleedharan, Funai, Kwon, & Heinzelman, 2012) a cloudlet with Nvidia GT520 GPU card can detect faces from an 800x480 image around 250 ms with a cost less than $100. Therefore we will assume that, face detection will be performed outside the cloud with surveillance devices that could be assisted with a cloudlet. At the end of this step, only face parts of the image are cropped and send to cloud for further processing.

- **Projection:** The projection step uses the image produced from face detection step and projects the image into eigenvalues that represents the weights to reconstruct the image from Eigenfaces stored in the database. The complexity of the projection step thus linearly increases with number of Eigenfaces on the database and image size in terms of pixels. The approximate time for projection step in ms is as follows (Alling, Powers, & Soyata, 2015):

$$T_P = \left(\frac{0.004001 \times p_{db} \times n_{db} \times r_{eigenfaces}}{GFLOPS_{device}} \right)$$

where p_{db} is the pixel count of an image in the database, n_{db} is the number of images in the database and $r_{eigenfaces}$ is the ratio of eigenfaces to images in the database.

- **Search:** This step uses the eigenvectors produced in the projection step and compares them with the eigenvectors in the database for a match. The face is recognized if the similarity of it is close to a database entry. The similarity is calculated by using Euclidean distance between eigenvector of the image and eigenvector for each database entry. Therefore complexity of this step depends on number of images in the database and number of dimensions in the eigenvector. Since the size of eigenvector depends on the number of images in the database, search time in ms can be approximated as follows:

$$T_S = 1000 \times \left(\frac{3.575 \times 10^{-7} \times n_{db}^2 \times 1.018 \times 10^{-4} \times n_{db} + 9.773 \times 10^{-3}}{GFLOPS_{device}} \right)$$

Operational Cost Worksheet for Case B

The computational requirements for face detection are analyzed in detail in (Alling, Powers, & Soyata, 2015) and will not be repeated here. We will use these requirements to calculate the cost of running face recognition in the cloud. First a database needs to be generated that contains Eigenfaces. Complexity of generating an Eigenface database increases exponentially with the number of faces contained in the database. Table 14 shows the computational complexity of building a database on an Nvidia GX760 GPU. Note that database must be loaded into memory during the search step of the face recognition.

We assume that the database has 5000 entries, which requires 5.6 GB storage space. GFLOPS of the CPUs in the cloud instances depend on the type of the processors. We will use the settings for the database with 5,000 images, which requires moderate time to generate a database and provides reason-

Table 14. Resource requirements for generating Eigenface database with an NVIDIA GX760 GPU.

Number of Images in Database	Computation Time	Database Size (GB)
1000	8.5 min	1.1
2000	57 min	2.2
3000	8.7 hr	3.4
4000	49 hr	4.5
5000	182 hr	5.6
10000	342 days	11.4
50000	1115 years	57.9

ably good coverage. Although it requires approximately one week to generate such a database, this step will be performed only once. Therefore we assume that the database is already generated and has a size of 5.6 GB. The cropped images from the face detection are assumed to be 180x180 pixels.

To compute the cost of running face recognition in the cloud, we look at the cost of recognizing a single face on a single virtual instance from cloud providers. This will be used as the unit price of face recognition and can be used for analyzing the cost for different case scenarios. We select virtual instances based on the amount of memory that can store the database, which is around 5.6 GB. Based on the memory requirements, we select the smallest instance that is capable of storing this database in the RAM. We select the high memory machine types from AWS, Microsoft Azure and Google Compute Engine. All high memory machine types have an Intel Xeon 2670 processor and we list the configurations selected from each cloud provider in Table 15.

We use the equations from the previous section to approximate the time for projection and search steps. The computation time of both steps will be based on the GFLOPS of the virtual instances. The virtual instances are assigned to one of the threads of an Intel Xeon processor. GLOPS of a virtual instance (thread) can be calculated as GLOPS = clock frequency (GHz) x (FLOP / cycle). Intel Xeon processors are usually capable of performing 8 floating-point operations per clock cycle, so we assume that GLOPS of a virtual instance is approximately 8 x clock frequency (GHz). Table 16 presents the time for computing each step of recognizing single face on the virtual instances.

The instance configurations have two virtual cores, therefore by using the instance types from Table 15, approximately 1,237K face recognition computations can be done. With these settings the cost of continuous face recognition over the period of a month will depend on two parameters: storage and computation, as shown in Table 17. The network bandwidth cost will be negligible since cloud providers do not charge for incoming data and outgoing data is a simple Yes/No response, which can be neglected. Google Compute Engine charges based on sustained usage discount which reduces the price based on usage. Specifically, for 0-25% usage regular price is charged and price is reduced by 20% with each 25% increase in the usage.

Table 15. Cloud Instance Configurations for Face Recognition.

Cloud Service Provider	Instance Type	Virtual Machines	Memory (GB)	Local Disk (GB)
AWS	r3.large	2	15.25	32
Microsoft Azure	D11	2	14	100
Google Compute	n1-highmem-2	2	13	?

Table 16. Projection and Search times for recognizing one face in the cloud.

CPU Type	GFLOPS	Projection (ms)	Search (ms)	Total (ms)
Intel Xeon 2670	20.8	3895	454	4349

Table 17. Cost of running continuous face recognition for a month in the cloud.

Cloud Service Provider	Compute Cost	Storage Cost	Total Cost
AWS	$48.4	$0.17	$48.57
Microsoft Azure	$161.8	$0.13	$161.93
Google Compute Engine	$75.0	$0.22	$75.22

Service Level Agreements for Cases A and B

SLA management is important for ensuring that QoS is maintained for running real time mobile cloud application for health monitoring and real time face recognition (case A and B). The SLA for health monitoring and real-time face recognition concept should by necessity involve stating the transaction times expected to process the data packets at least at the same rate it receives data management to make sure that patients' data within these systems are accurate and complete, up-to-date with adequate fail safe, secure, backup and archiving process in place that is encrypted. The SLA should ensure storage space of patients being monitored for a 24 hr period is approximately 252 GB per month. Bandwidth requirements are vital and accurate through ECG patches and transmitted via the Internet to be stored on the ECG database.

SURVEY OF CURRENT MOBILE CLOUD OPERATOR COST MODELS

The previous part of this chapter analyzed operational costs strictly in terms of the cloud user, which is generally an organization that outsources these aforementioned applications to one of the existing cloud operators. In the following section, the analysis will be repeated, albeit, from the standpoint of the cloud operators. Cloud operator costs and profit models will be introduced to perform such analysis.

Data centers have evolved dramatically in recent years, due to the advent of social networking services, e-commerce and cloud computing. However, the requirement of services for users can be conflicting in that, there is high availability levels demanded against low sustainability impact and cost values. This means, for cloud service providers, the cost models they have determines their profitability and service levels they can provide. Cost models help business owners and managers of service providers to figure out the cost for cloud services activities and processes. Through the use of financial computations or cost accounting allocation, companies can take basic information relating to resources, energy use, infrastructure development and direct labor and transform data into useful costs for setting the price of services. Companies can put together different cost models based on their needs, whether financial or operational (Markendahl, Makitalo, & Werding, 2008).

Generally, different companies use cost models in their daily operations because the goal is to maximize the economic value for owners and shareholders. Finding ways to lower costs is a crucial step in achieving this goal. Another purpose for cost models is to create a repeatable process that allows owners and managers to apply the model to multiple situations. Through this business process, the company can develop a metric that becomes the standard expected rate of return for projects. This safeguards the company from losing money when engaging in new business opportunities that look profitable but really are not (Slack, Brandon-Jones, & Johnston, 2013).

According to (Fetai & Schuldt, 2012), for many organizations, especially cloud service providers, it is relatively complicated to determine the exact total operational costs incurred by offering own services in the cloud as well as to compare them with the costs incurred by datacenters. For instance, the average cost per year according to (Alford & Morton, 2010), to operate a large datacenter is usually between $10 million to $25 million, while according to an organization with 1,000 file servers faces average costs in the cloud between $22.5 million and $31.1 million. In practice, models exist to help support organizations in analyzing and comparing costs (Alford & Morton, 2010). Some of these cost models identified from literature are explained below.

The methodology adopted for our survey involves reviewing research findings by (Van den Bossche, Vanmechelen, & Broeckhove, 2013) regarding mobile cloud computing topics. This review was undertaken to reveal where there are gaps in the research literature to help contextualize this research in line with best practice approach towards the execution of this real-time research project (Johansson, 2007). This is also to make sure that the research has not already been done. Thus existing literature on this topic was examined to help position this study within the context of existing evidence by (Rickinson & May, 2009) to identify the main components for mobile cloud application cost analysis. For the purpose of this chapter, our scope was to explore (i) current mobile cloud application cost models; (ii) explore the pricing models that have evolved based on these cost models; (iii) illustrate how the operational cost of a mobile application implementation can be calculated and (iv) present a framework for establishing costs. This was based on empirical study in published information of existing work from academia, industrial, marketing and business sites. Secondly, using the various search engines, reading of articles, journals and Government and corporate publications (Rickinson & May, 2009).

Performance and Cost Assessment of Cloud Operations

A performance and cost assessment model has been developed as a modeling technology for modeling the performance and scalability of service oriented applications designed for a variety of platforms. Using a suite of cloud testing applications, (Brebner & Liu, 2011) conducted empirical evaluations of a variety of real cloud infrastructures, including Google App Engine, Amazon EC2, and Microsoft Azure. The insights from these experimental evaluations, and other public/published data, were combined with the modeling technology to predict the resource requirements in terms of cost, application performance, and limitations of a realistic application for different deployment scenarios. Costs in terms of power consumption per year ranges from $7,800 to $13,000 (Australia dollars for Energy business plan). Naturally the real total cost of ownership is substantially more and includes cooling, carbon emission offsets or green power sources, software licensing, maintenance and administration, capital costs and depreciation.

System Utilization Charging Model

(Woitaszek & Tufo, 2010) developed a comprehensive system utilization charging model similar to that used by Amazon EC2 and applied the model to current resources and planned procurements. This is a model for charging computational time, data transfers and storage. The charging rate is between 3 and 5 cents per CPU hour, a rate not competitive with the performance of current commodity processors. System and container ($8M / 5 years) $1,600,000 combined with annual operational staffing of $ 600,000 and annual power and cooling $365,000 and power expenses, the system is estimated to cost $2,565,000 per year. Depending on the number of CPU cores selected, the per-CPUh break-even charge (assuming 100% utilization and including the cost of the attached storage) will be 0.89 cents/CPUh to 1.7 cents/CPUh.

Dynamics of Cost Development

(Kristekova Z. et al, 2012) proposed a simulation model for analyzing cost-benefits between cloud computing and own datacenter, as well as by analyzing different scenarios virtually before transferring them into the real world. This simulation model for cost-benefit analysis of cloud computing versus own datacenter has been proposed with the potential to fill the gap where a cost model that covers dynamic

issues of cloud computing is lacking. In their model, to estimate the costs for a server, the initial cost of server, operating system licenses and additional network equipment were taken into account. The costs for server are calculated as product of "number of required server" and "initial costs for server." The costs for operating system licenses consist of "number of required server" and "costs per operating system license." The costs for additional network equipment are calculated as product of "number of server" and "the expenditures of network equipment." The expenditures of network equipment usually consist of 10 to 30% of the costs of server. Additionally, the ongoing maintenance costs for server and network equipment needs to be calculated in order to estimate the costs for infrastructure as sum of "power usage server", "power usage network equipment" multiplied with the costs of the desired tier level. In this way this model includes the power usage of the infrastructure to determine the power usage effectiveness (PUE), which is given by PUE = Total Facility Power/ IT Equipment Power. The PUE value can range from 1.0 to infinity, where 1.0 indicates 100% efficiency. The realistic PUE values are in the 1.3 to 3.0 range.

To calculate the costs for the administration, an estimate of how many servers one administrator can maintain is vital. This in turn depends on the size of datacenter. As such the final administration cost is, the "number of servers" divided by "the number of estimated server maintenance per administrator" multiplied by "the costs for one administrator." For the data transfer costs, many companies rely on the flat rates. After estimating the costs for hardware, software, infrastructure, administration and data transfer, then sum all these costs to obtain the total costs for data center. This particular consideration has fed into pricing component based on for example CPU usage and transactions rate required for processing application requirements (Kristekova Z. et al, 2012).

Inter-Organizational Economic Models for Pricing Cloud Network Services

(Pal & Hui, 2013) developed inter-organizational economic models for pricing cloud network services when several cloud providers co-exist in a market servicing a single application type. The development involved analysis and comparison of models that cloud providers can adopt to provision resources in a manner such that there is *minimum* amount of resources wasted, and at the same time the user service-level/ QoS is maintained. On this basis, the ability to process the number of user requests processed per unit of time on the cloud determines the amount of resources to be provisioned to achieve a required capacity.

View Materialization Cost Model

(Nguyen, Bimonte, d'Orazio, & Darmont, 2012) proposed an approach for decreasing the cost of data management in the cloud, by using a classical database performance optimization technique, such as view materialization. This cost model complements existing materialized view cost models with a monetary cost component that is rudimentary in the cloud. This is the basis of the pricing models incorporated by Amazon to render more versatile service. This model also fits into the 'pay as you go paradigm' of cloud computing and allow providers achieve a multi-criteria optimization of the view materialization versus CPU power consumption problem, under budget constraints.

Nguyen et al, demonstrated the power of the model by introducing a fictitious example, which illustrated the complexity of selecting materialized views in the cloud. In this illustration, the storage cost is say $0.10 per GB per month, and the computing cost is $0.24 per hour and a 500 GB dataset is stored in the cloud for a month. If Q is the monthly query workload, processed in 50 hours then, the storage cost

is $50, computing cost is $12, for a total of $62. If some materialized views are used, it was assumed that workload processing time becomes 40 hours. Thus, computing cost becomes $9.6. However, materialized views use up additional storage space of, 50 GB. Thus, storage cost becomes $55, for a total cost of $64.6. This means that overall, performance has improved by 20%, but cost has also increased by 4%.

Cost-Efficient Scheduling of Hybrid Cloud

According to (Van den Bossche, Vanmechelen, & Broeckhove, 2013), MCC has found broad acceptance in both industry and research with public cloud offerings is now often used in conjunction with privately owned infrastructure. Technical aspects such as the impact of network latency, bandwidth constraints, data confidentiality and security, as well as economic aspects such as sunk costs and price uncertainty are key drivers towards the adoption of a hybrid cloud model. This model introduced a hybrid cloud to determine which workloads are to be outsourced and to what cloud provider. The choice of how this is done can minimize the cost of running a partition of the total workload on one or multiple public cloud providers while taking into account the application requirements such as deadline constraints and data requirements according to. This is because, the variety of cost factors, pricing models and cloud provider offerings makes the consideration of the automated scheduling approach in hybrid clouds model worthwhile.

In (Van den Bossche, Vanmechelen, & Broeckhove, 2013) model a set of algorithms were used to cost-efficiently, scheduling the deadline-constrained bag-of-tasks applications on both public cloud providers and private infrastructure was proposed. The algorithms took into account both computational and data transfer costs as well as network bandwidth constraints. Using this model, Bossche et al evaluated and assessed the performance in a realistic setting with respect to cost savings, deadlines met, computational efficiency, and the impact of errors in runtime estimates on these performance metrics. On the basis of this model, some service providers offer mixed public and private options in their price components to maximize the utilization of their computing resources through shared services.

Mercury Cost Model

The Mercury cost model introduced by (Callou, Ferreira, Maciel, Tutsch, & Souza, 2014), is a tool for dependability, performance and energy flow evaluation. The tool supports reliability block diagrams (RBD), stochastic Petri nets (SPNs), continuous-time Markov chains (CTMC) and energy flow models (EFM). The EFM verifies the energy flow on data center architectures, taking into account the energy efficiency and power capacity that each device can provide (assuming power systems) or extract (considering cooling components). The EFM also estimates the sustainability impact and cost issues of data center architectures. The approach of this model is to evaluate and optimize these requirements to support cloud infrastructure solution designers. This offers an integrated approach in order to estimate and optimize high conflicting requirements and the availability levels demanded against the low sustainability impact and cost values.

Cloudlets Model Extending the Utility of Mobile-Cloud Computing

(Soyata, Ba, Heinzelman, Kwon, & Shi, 2013) defined application cost as an objective task that quantifies the fees charged by Cloud operators, such as Amazon Web Services, during the execution of the application. In their example, they stated that, Amazon charges for compute-usage per hour per CPU

occurrence. This means increasing the application costs as the required amount of computation increases. According to Soyata et al, cloud operators' charges for Microsoft SQL server, for example, is based on the usage of database occurrences. Hence Soyata et al, proposed the Cloudlet model (Soyata, et al., 2012) of extending the utility of mobile cloud computing without increasing resource use. Soyata et al, illustration shows that, applications requiring higher computational and storage resources might cost more during operation on a Cloud platform such as AWS with variety of options when executing mobile-cloud applications (Wang, Liu, & Soyata, 2014) (Powers, Alling, Gyampoh-Vidogah, & Soyata, 2014) (Page, Kocabas, Ames, & Venkitasubramaniam, 2014).

Table 18 summarizes the cost models identified from literature, what they support and contribution of these models, authors and how they have been used in real life applications. The next section discusses the current application and price components offered by cloud service providers and how they relate to the aspects of the cost models.

On the basis of the cost models, different cloud charging and billing models allow choice of hosting options between and even within cloud platforms. This adds both flexibility and complexity to modeling (Brebner & Liu, 2011). For the purpose of operational costs of mobile computing pricing, operational costs for various resource types and loads such as the cost of CPU, network, data management, security

Table 18. Summary of the Cost Models

Cost Model	Application	Contribution	Author
Service Oriented performance cost model	Modeling the performance and scalability of Service Oriented applications architected for a variety of platforms.	Predict the resource requirements in terms of cost application performance.	(Brebner & Liu, 2011)
System utilisation cost model	Billing model of Computational time, data transfers & storage	Applied to current resources and planned procurements.	(Woitaszek & Tufo, 2010)
Dynamics of cost development	Intended to fill the gap where a cost model that covers dynamic issues for cloud is lacking.	Analysed cost benefits between cloud computing and datacenter.	(Kristekova Z. et al, 2012)
Inter-organizational economic models for pricing cloud network services	Analysis and comparison of models.	The ability to process the number of user requests processed per unit of time on the cloud.	(Pal & Hui, 2013)
View Materialization cost model	For decreasing the cost of data management in the cloud.	Cost model that complement existing materialised view cost model.	(Nguyen, Bimonte, d'Orazio, & Darmont, 2012)
Cost-efficient scheduling of Hybrid Cloud	Determine which workloads are to be outsourced, and to what cloud provide.	Service providers that mixed public and private options in their price components are to maximize the utilization of their computing resources through shared services.	(Van den Bossche, Vanmechelen, & Broeckhove, 2013)
Mercury Model	This model is to evaluate and optimize these requirements to support cloud infrastructure solutions designers.	Offers an integrated approach to estimate and optimize high conflicting requirements and the availability levels demanded against the low sustainability impact and cost values.	(Callou, Ferreira, Maciel, Tutsch, & Souza, 2014)
Cloudlets	For extending the utility of mobile-cloud computing by providing compute and storage resources accessible for end processing of applications	Compared different approaches that enhance application performance via cloud-based execution.	(Soyata, Ba, Heinzelman, Kwon, & Shi, 2013)

operations, transactions and connections are used by providers. These components essentially relate to activity based costing, an accounting concept through which organizations recoup costs incurred throughout normal business activity cycle (Cloudscape, 2013).

The same concept combined with trace analysis, has been applied to resources in the cloud based on the cost models. This emphasizes the fact that a cloud resource, whether it is a system, an application or a service, would have some form of metrics collected either to analyze performance or for the purpose of costing (Mihoob, Molina-Jimenez, & Shrivastava, 2011). Subsequently, providers on the basis of cost models discussed in the previous sections have derived several price components for users which are often not explained but providers have to build into their pricing mechanisms. These are: Transaction, data management, CPU time (resource use), traffic or network capacity and storage space components and the operational expenditures. For the purpose of this chapter, the main pricing components are explained as a basis of working out the operational costs to businesses for implementing real time mobile cloud based systems.

SUMMARY

This chapter provided background on mobile cloud computing in general and described in detail the concepts and cost models that have been used to determine the cost of providing cloud services and running real-time mobile-cloud applications. These cost analyses have been done from two different perspectives: the mobile-cloud user and the cloud operators. From the standpoint of the mobile-cloud user, the cost of running a real-time application involves renting the most suitable cloud resources (computation, storage, network bandwidth) from a cloud operator. On the other hand, the costs of a cloud operator primarily involve datacenter operating expenses such as electricity, equipment depreciation, and the ability to take advantage of economies of scale by sharing resources among multiple customers.

Two case studies are used as the basis of explaining the process of arriving at operational cost:s. First case study is a long-term health monitoring system described by (Kocabas, et al., 2013) that uses Homomorphic encryption to achieve privacy-preserving medical computation in the cloud. The second case study is a real-time mobile-cloud face recognition system described (Soyata, Ba, Heinzelman, Kwon, & Shi, 2013) which uses a cloudlet for acceleration to reduce the computational and network bandwidth burden on cloud instances. This second system is designed primarily as a means to extend the utility of mobile-cloud computing to provide computational and storage resources accessible at the edge of the network, both for end processing of applications and for managing the distribution of applications to other distributed compute resources.

Worksheets which demonstrate how businesses can estimate the operational cost of implementing real-time mobile cloud applications are provided. The starting point of this estimation is first a cost analysis for the mobile cloud operation constructed as the architecture of the system. Then the estimation of each cost component is provided as defined by cloud operators to estimate the actual total operational cost of running the real time mobile application. A simple step has been illustrated to show how the process used in these worksheets can be used to guide in estimating the true real-time mobile cloud application operational costs. The aim is to help decision makers make the business case for specific applications. Steps to establish the application process is to assess current models either to rent or buy. When the analysis of operational costs are established, then construction of real time operational costs should be establish that include the whole process and evaluation of applications.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation grant CNS-1239423 and a gift from Nvidia Corporation.

REFERENCES

Alford, T., & Morton, G. (2010). *The Economics of Cloud Computing*. Retrieved from http://www.boozallen.com/media/file/Economics-of-Cloud-Computing.pdf

AliveCor. (n.d.). Retrieved from http://www.alivecor.com/home

Alling, A., Powers, N., & Soyata, T. (2015). Face Recognition: A Tutorial on Computational Aspects. In *Emerging Research Surrounding Power Consumption and Performance Issues in Utility Computing*. Hershey, PA: IGI Global.

Amazon EC2. (2013). Retrieved from Retrieved from: http://aws.amazon.com/ec2/sla/

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., & Stoica, I. et al. (2010). A View of Cloud Computing. *Communications of the ACM*, *53*(4), 50–58. doi:10.1145/1721654.1721672

Microsoft Azure. (2014). Retrieved from Microsoft Azure: http://azure.microsoft.com/en-us/support/legal/sla/

Bansal, N. (2013). Cloud computing technology (with BPOS and Windows Azure). *International Journal of Cloud Computing*, *2*(1), 48–60. doi:10.1504/IJCC.2013.050955

Bazzett, H. (1997). An analysis of the time-relations of electrocardiograms. *Annals of Noninvasive Electrocardiology*, *2*(2), 177–194. doi:10.1111/j.1542-474X.1997.tb00325.x

Brebner, P., & Liu, A. (2011). Performance and Cost Assessment of Cloud Services. *Computer Science*, *6568*, 39–50.

Callou, G., Ferreira, J., Maciel, P., Tutsch, D., & Souza, R. (2014). *Open Access Energies.* Retrieved from An Integrated Modeling Approach to Evaluate and Optimize Data Center Sustainability, Dependability and Cost: http://www.mdpi.com/1996-1073/7/1/238

CareCloud. (n.d.). Retrieved from http://www.carecloud.com/

Chappel, C. (2013, June 22). *Heavy Reading: Unlocking Network value: service Innovation in the Era of SDN, White paper.* Retrieved from http://www.cisco.com/web/solutions/trends/open_network_environment/docs/hr_service_innovation.pdf

Cloudscape. (2013, June 22). *451 Research.* Retrieved from The Cloud Pricing Codex: https://451research.com /report-long?icid=2770

Couderc, J.-P. (2010). The Telemetric and Holter ECG Warehouse initiative (THEW): A data repository for the design, implementation and validation of ECG related technologies. *IEEE Engineering in Medicine and Biology Society (EMBC)*, 6252-6255.

Couderc, J.-P., Garnett, C., Li, M., Handzel, R., McNitt, S., Xia, X., & Zareba, W. et al. (2011). Highly automated QT measurement techniques in 7 thorough QT studies implemented under ICH E14 guidelines. *Annals of Noninvasive Electrocardiology*, *16*(1), 13–24. doi:10.1111/j.1542-474X.2010.00402.x PMID:21251129

Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2013). Wireless Communications and Mobile Computing. *Wireless communications and mobile computing*, 1587-1611. doi:10.1002/wcm.1203

Fesehaye, D., Gao, Y., Nahrstedt, K., & Wang, G. (2012). Impact of cloudlets on interactive mobile cloud applications. *In Proceedings of Enterprise Distributed Object Computing Conference (EDOC)* (pp. 123-132). Washington DC: IEEE Computer Society, USA. doi:10.1109/EDOC.2012.23

Fetai, I., & Schuldt, H. (2012). Cost-Based Data Consistency in a Data-as-a-Service Cloud Environment. *Cloud 12 Proceedings of the IEEE Fifth International Conference on Cloud Computing* (pp. 526-533). Washington, DC: IEEE.

Gentry, C., Halevi, S., & Smart, N. (2012). *Homomorphic evaluation of the AES circuit* (pp. 850–867). CRYPTO.

Google. (2013, September 1). *Cloud Standards Customer Council.* Retrieved from What to Expect and What to Negotiate: Public Cloud Service Agreements.

Halevi, S., & Shoup, V. (n.d.). *HELib.* Retrieved from HELib: https://github.com.shaih/HElib

HIPAA. (n.d.). *Health Insurance Portability and Accountability Act*. Retrieved from http://www.hhs.gov/ocr/privacy/

IBM. (2013). *Shaping the future of the oil and gas industry with smarter cloud computing. Thought Leadership White Paper.* IBM Corporation. Retrieved from http://www-935.ibm.com/services/multimedia/Shaping_the_future_of_the_oil_and_gas_industry_with_smarter_cloud_computing.pdf

Johansson, K. (2007, June 22). *Cost Effective Deployment Strategies for Heterogeneous Wireless Networks.* Royal Institute of Technology.

Kocabas, O., & Soyata, T. (2014). Medical Data Analytics in the cloud using Homomorphic Encryption. In *Handbook of Research on Cloud Infrastructures for Big Data Analytics* (pp. 471–488). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-5864-6.ch019

Kocabas, O., Soyata, T., Couderc, J.-P., Aktas, M., Xia, J., & Huang, M. (2013). Assessment of Cloud-based Health Monitoring using Homomorphic Encryption. *Proceedings of the 31th IEEE Conference on Computer Design*, (pp. 443-446). Asheville, NC, USA. doi:10.1109/ICCD.2013.6657078

Kovachev, D., Cao, Y., & Klamma, R. (2013). Retrieved from Mobile Cloud Computing: A Comparison of Application Models.: http://arxiv.org/abs/1107.4940v1

Kristekova, Z., ... B. J. (2012, June 15). *Simulation Model for Cost-Benefit Analysis of Cloud Computing versus In-House Datacenters.* Retrieved from http://digisrv-1.biblio.etc.tu-bs.de: 8080/ docportal/servlets/MCRFileNodeServlet

Kwon, M., Dou, Z., Heinzelman, W., Soyata, T., Ba, H., & Shi, J. (2014). Use of Network Latency Profiling and Redundancy for Cloud Server Selection. *Proceedings of the 7th IEEE International Conference on Cloud Computing*, (pp. 826-832). Alaska, USA. doi:10.1109/CLOUD.2014.114

Markendahl, J., Makitalo, O., & Werding, J. (2008). Analysis of Cost Structure and Business Model options for Wireless Access Provisioning using Femtocell solutions. *19th European Regional ITS Conference*.

Microsoft. (2010, May 15). *The Economics of Cloud.* Retrieved from http://www.microsoft.com/en-us/news/presskits/cloud/docs/the-economics-of-the-cloud.pdf

Mihoob, A., Molina-Jimenez, C., & Shrivastava, S. (2011). *Consumer side resource accounting in the cloud*. Berlin, Germany: Springer. doi:10.1007/978-3-642-27260-8_5

Mouftah, H. T., & Kantarci, B. (2013). *Communication Infrastructures for Cloud Computing*. Hershey, PA: IGI Global.

MS-SLA. (n.d.). Retrieved from Microsoft SLA Storage - Introduction: https://azure.microsoft.com/en-us/documentation/articles/storage-introduction/

Nguyen, T.-V.-A., Bimonte, S., d'Orazio, L., & Darmont, J. (2012). Cost models for view materialization in the cloud. *EDBT-ICDT '12 Proceedings of the Joint EDBT/ICDT Workshops*, (pp. 47-54). New York.

NIST-AES. (2001, Nov). *FIPS-197.* Retrieved from Advanced Encryption Standard (AES).

Page, A., Kocabas, O., Ames, S., & Venkitasubramaniam, M. (2014). Cloud-based Secure Health Monitoring: Optimizing Fully-Homomorphic Encryption for Streaming Algorithms. *IEEE Globecom 2014 Workshop on Cloud Computing Systems, Networks, and Applications.*

Page, A., Kocabas, O., Soyata, T., Aktas, M., & Couderc, J.-P. (2014). Cloud-Based Privacy-Preserving Remote ECG Monitoring and Surveillance. *Annals of Noninvasive Electrocardiology*, n/a. doi:10.1111/anec.12204 PMID:25510621

Pal, R., & Hui, P. (2013). Economic Models for Cloud Service Markets: Pricing and Capacity Planning. *Journal of Distributed Computing and Networking*, *496*, 113–124.

Powers, N., Alling, A., Gyampoh-Vidogah, R., & Soyata, T. (2014). AXaaS: Case for Acceleration as a Service. *IEEE Globecom 2014 Workshop on Cloud Computing Systems, Networks, and Applications.*

Prasad, R. M., Gyani, J., & Murti, P. (2012). Mobile Cloud Computing: Implications and Challenges. *Journal of Information Engineering and Applications*, *2*(7), 1–15.

Rickinson, M., & May, H. (2009). *A comparative study of methodological approaches to reviewing literature.* The Higher Education Academy. Retrieved June 15, 2014, from Http://www.heacademy.ac.uk/assets/ documents/ resources/comparativestudy.pdf

Rimal, B., & Choi, E. (2012). A service-oriented taxonomical spectrum, cloudy challenges and opportunities of cloud computing. *International Journal of Communication Systems. Special Issue*, *25*(6), 796–819. doi:10.1002/dac.1279

Shanklin, W. (2014, March 26). *Revisiting Cloud Computing: how has it changed - and changed us?* Retrieved from Gizmag: http://www.gizmag.com/revisiting-cloud-computing/26768/

Slack, N., Brandon-Jones, A., & Johnston, R. (2013). *Operations Management*. Lombarda, Italy: Pearson.

Soyata, T., Ba, H., Heinzelman, W., Kwon, M., & Shi, J. (2013). Accelerating mobile cloud computing: A survey. In *Communication Infrastructures for Cloud Computing* (pp. 175–197). IGI Global.

Soyata, T., Muraleedharan, R., Funai, C., Kwon, M., & Heinzelman, W. (2012). Cloud-Vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture. *Computers and Communications (ISCC), 2012 IEEE Symposium on*, 59-66.

Soyata, T., Muraleedharan, R., Langdon, J., Funai, C., Kwon, M., & Heinzelman, W. (2012). COMBAT: mobile-Cloud-based cOmpute/coMmunications infrastructure for BATtlefield applications. *Proceedings of the Society for Photo-Instrumentation Engineers*, 8403.

US-HHS. (n.d.). *Business Associate Agreement*. Retrieved from http://www.hhs.gov/ocr/privacy/hipaa/understanding/coveredentities/contractprov.html

Van den Bossche, R., Vanmechelen, K., & Broeckhove, J. (2013). Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds. *Computer Systems*, *29*(4), 973–985. doi:10.1016/j.future.2012.12.012

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (pp. 511-518). doi:10.1109/CVPR.2001.990517

Wang, H., Liu, W., & Soyata, T. (2014). Accessing Big Data in the Cloud Using Mobile Devices. In *Handbook of Research on Cloud Infrastructures for Big Data Analytics* (pp. 444–470). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-5864-6.ch018

Woitaszek, M., & Tufo, H. M. (2010). Developing a Cloud Computing Charging Model for High-Performance Computing. *10th IEEE International Conference on Computer and Information Technology* (pp. 210-217). Bradford: IEEE. doi:10.1109/CIT.2010.72

Wu, L. (2014). *SLA-based Resource Provisioning for Management of Cloud-based Software-as-a-Service Applications* (Doctoral dissertation). Retrieved from University of Melbourne Cloud Laboratory: http://cloudbus.org/students/LinlinPhDThesis2014.pdf

KEY TERMS AND DEFINITIONS

Advanced Encryption Standard (AES): A symmetric-key encryption system used for encrypting/decrypting digital data with the same private key.

Amazon Web Services (AWS): Cloud computing services provided by Amazon.com.

Business Associate Agreement (BAA): An agreement that a vendor of an HCO signs confirming that, they will treat PHI according to HIPAA laws to prevent the HCO itself from violating it.

Cloud Computing: Outsourcing computation and storage to a company that rents these entities at a much lower unit price than is possible to produce by any user. Cloud operators take advantage of economies of scale by aggregating requests from multiple customers to lower the cost of these commodity "utilities."

Cloud Operator: A company that rents computation and storage in a way similar to utility companies (e.g., gas and electric companies) do by adhering to monthly or other types of billing. Most notable cloud operators are Microsoft (Azure), Amazon, and Google.

Datacenter: A facility to store and maintain computer servers, storage devices and networking systems. The high level operating software that runs in the "cloud" allows the cloud operator to channel the computation/storage requests to resources based on cost and availability criteria.

ECG Patch: An external sensor that is attached to a patient's body, typically in multiple body points, that records the ECG signals in a way similar to the ones obtained at the HCO. The advantage of using an ECG patch is the ability to obtain a patient's ECG outside the HCO, for long term health monitoring. This could provide significantly more information to a doctor in terms of the cardiac functionality of a patient over the long term, and potentially allow the doctor to identify cardiac issues that cannot be indentified in short term ECG recordings obtained at the HCO.

Electrocardiogram (ECG): A diagram, typically printed on specialized ECG paper, that contains information about the cardiac functionality of a patient within a short period of time (e.g., less than a minute). The information that is plotted is the voltage levels on each electrode (i.e., "lead"). By looking at such information, multiple cardiac conditions can be readily identified by a healthcare professional.

Face Recognition: A process of identifying a person by comparing facial features of the person with a database of known faces. In this operation, multiple pre-processing steps (such as conversion from faces to Eigenfaces) help substantially reduce the processing necessary to reach the same computational results.

Fully Homomorphic Encryption (FHE): A public key encryption algorithm that can perform addition or multiplication operations on encrypted data. Using FHE, it is possible to do cloud computing, where the cloud cannot observe the data that it is operating on. This allows the introduction of secure medical cloud applications where the medical data privacy concerns are eliminated, since the cloud can never observe the patient data.

Google App Engine: Platform as a Service (PaaS) provided by Google Cloud Platform.

Google Cloud Platform: PaaS and SaaS services provided by Google for deploying and managing applications in the cloud.

Graphics Processing Unit (GPU): A deice that is capable of processing significantly higher amounts of data (typically one or two orders-of-magnitude) as compared to a CPU. Computation using a GPU does not necessarily benefit every application, but, rather, ones that have such massive parallelism. Almost every Image Processing application can benefit from the parallelism of a GPU due to the inherent parallelism in the structure of an image, composed of millions of pixels.

Health Insurance Portability and Accountability Act (HIPAA): A set of rules and regulations to protect the privacy of an individual's medical information, i.e., PHI. An example such rule is covering a computer monitor with a privacy screen to avoid a third party from seeing the individual's information.

Healthcare Organization (HCO): Any organization in the chain of organizations that provides a phase of the healthcare to an individual. These organizations are subject to HIPAA laws.

Hybrid Cloud: A cloud computing deployment model where cloud infrastructure is a combination of private and public cloud resources.

IaaS (Infrastructure as a Service): A cloud computing service model where cloud providers deliver computing infrastructure. These services may include servers, storage, and operating systems. Clients provision resource needed to run their applications and outsource the required resources form cloud providers. Examples of IaaS are Google Compute Engine, Amazon Web Services and Microsoft Azure.

Long QT Syndrome (LQTS): A cardiac condition where the QTc interval is prolonged. The variation among a large population should be small when QTc (corrected) value of the QT interval is used. Safe values of QTc are below 440 ms, though, values under 500 ms are acceptable. Patients with QTc $>$ 500 ms are under risk of serious cardiac hazards.

Microsoft Azure: PaaS and SaaS services provided by Microsoft for deploying and managing applications in the cloud.

Mobile-Cloud Computing: A computing model that combines mobile devices and cloud computing to remedy lack of resources of the mobile devices such as computation power, storage and battery. Mobile devices use cloud resources for data processing and storage.

PaaS (Platform as a Service): A cloud computing service model where cloud providers deliver computing platforms and environments to allow clients to deploy and manage their applications. These services may include programming languages, libraries, tools and services. Clients control deploying applications and cloud providers manage the underlying cloud infrastructure based on demand. Examples of PaaS are Google App Engine, Microsoft Azure, Heroku.com.

Pretty Good Privacy (PGP): A public-key cryptography system to encrypt and sign digital data.

Private Cloud: A cloud computing deployment model where the cloud services is owned and managed by a single organization.

Protected Health Information (PHI): Any information that can be associated with an individual, such has his/her health status. This information must be protected to avoid a privacy violation of the individual.

Public Cloud: A cloud computing deployment model where the cloud services is open to use for general public. The cloud infrastructure may be owned, managed by third party and services are provided to general public.

QT Interval: QT interval is another very common internal obtained from the ECG. This internal on each heartbeat delineates the ventricular recovery phase of the heart.

QTc Value: Since the QT interval varies based on the heart rate, a corrected version of the QT interval is much more meaningful to use in identifying proper cardiac operation. QTc adjusts QT for heart rate, and is, therefore, a lot more steady over a general population. Nearly 100 years ago, Bazzett suggested a correction formula of $QTc=QT/\sqrt{RR}$, which is is known as Bazett's formula, or QTcB. An alternative suggestion followed from Fridericia which proved to be more accurate for a wider range of heart rates, which replaces the square root of RR with the cube root. Bazzett and Frediricia formulas can be written as: $QTcB = QT/(RR/sec)^{1/2}$ $QTcF = QT/(RR/sec)^{1/3}$. The divisions of RR by 1 second (i.e., /sec) are in place to preserve units between QT and QTc.

RR Interval: The most commonly used metric that is obtained from the ECG. Each beat has a similar patterns with readily identifiable Q, R, S, T, and U points. The temporal distance between two R points is called the RR interval, or the heart rate.

SaaS (Software as a Service): A cloud computing service model where cloud providers deliver services as software applications hosted in the cloud. Cloud providers manage the cloud infrastructure that is needed to run application software. Clients access these services from a client interface (e.g, web browser). Examples of SaaS are Gmail, Microsoft Office 365, Saleforce.com.

Service Level Agreement (SLA): An agreement between service providers and clients that establishes the scope, quality and responsibilities of service provider to the client.

Tiered Pricing: A pricing model that sets the price of per unit item based on a range (i.e., tier).

Utility Computing: A service-provisioning model that provides computing resources and infrastructure to clients based on demand. Computing resources might include servers, storage and services. Service providers charge clients based on the usage instead of a flat rate.