Systems Engineering at MITRE

CLOUD COMPUTING SERIES

Platform as a Service: A 2010 Marketplace Analysis

Geoffrey Raines
Lawrence Pizette

MITRE

October 2010

Table of Contents

THE BIG PICTURE: Contemporary cloud computing Platform as a Service (PaaS) vendors make up a healthy marketplace with a wide range of service offerings covering key functions desired by an architect for enterprise solutions, and as the number of service providers grows, the implementation options for the IT decision maker are expanding.

Platform as a Service: A 2010 Marketplace Analysis

Geoffrey Raines
Lawrence Pizette

1.0 Introduction to Platform as a Service

Platform as a Service (PaaS) is a term that covers a distinct but wide range of the cloud computing marketplace. Platform offerings build upon cloud infrastructure, such as large-scale storage and raw processing power, to provide value-added services, ranging from development environments for the creation and operation of Web-based software applications, to programmable computational platforms and massive data stores. PaaS examples include capabilities for application design, application development, testing, deployment and hosting as well as application services such as team collaboration, security, testing and monitoring, application versioning, and application instrumentation, tying together physically separated developer teams working through their browsers to leverage the virtual cloud platform. Platforms are also expanding to real-world integration platform services including business process integration, and assured system-to-system messaging.

> *"The term "PaaS" describes many different approaches, each of which has a particular "sweet spot" among application scenarios. […] With the worldwide recession squeezing budgets, it is prudent for application development managers to create a position on PaaS and perhaps to even start experimenting with one of the products."[1]*
>
> *– John Rymer, Forrester*

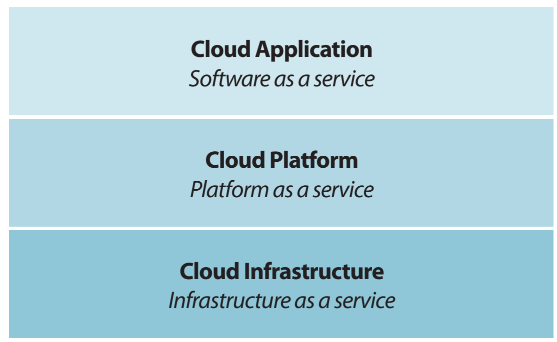

Figure 1.0-1. Platform as a Service and the NIST Cloud Computing Stack

As shown in Figure 1.0-1 PaaS fits in a notional stack of cloud computing offerings defined by the National Institute of Standards and Technology (NIST). In the NIST definition, PaaS generally relies on Infrastructure as a Service (IaaS), leveraging IaaS's underlying processing and storage capabilities. Finally, PaaS can often be used to create value-added Web systems for end-users, commonly referred to as Software as a Service (SaaS).

This paper, intended to support technical staff who architect enterprise-wide solutions, captures a summary of marketplace research taking place in the summer of 2010, and examines many of the commercial cloud PaaS providers in the marketplace, organizing them into general service provider categories such as development and operation platforms, database platforms, middleware and integration platforms, computational platforms, testing platforms, and billing platforms. The intent is to show the diversity and the current state of the PaaS marketplace. Some of the offerings discussed in the paper have strong brand recognition, while many may be new to the reader. Of course, the marketplace for new computing technologies is fluid and one can expect that the offerings in this paper will continue to evolve, new firms will be stood up to offer new services, and merger and acquisition (M&A) activities will have a large impact on the market.

For expediency, when a market segment has a group of similar cloud service providers, we felt it was best to describe exemplars of service providers in each broad platform category, since the market is growing quite large, with hundreds of new firms. While the market may have a dozen providers in a particular category, we looked to find service providers that would effectively represent the concepts of that genre. In select cases, only one service provider offered a service of particular interest, without a set of related competitors, and we note that in our service description. Our selection of one vendor over another for this exemplar discussion is not an endorsement of that vendor or their products.

One purpose of this document is to demonstrate the breadth of the PaaS market to those who may only be casually familiar with the most popular services, such as those by Amazon or Google. Also, the document hopes to put a magnifying glass on some of the valuable component services within the larger often-mentioned portfolios. Beyond the headlines there is a wealth of information on individual service functional descriptions, service security mechanisms, and service configuration parameters. The footnotes throughout the following sections often point to places to read further from an application programmer's point of view.

2.0 Development and Operations Platforms

One of the key uses of PaaS is to develop and field applications for the public Web, and one of the predominant added values is the ability to tie into and leverage a fully scalable underlying cloud computing infrastructure. In this context PaaS is leveraged to provide Software as a Service (SaaS) for a potentially world-wide audience of end users. This section explores example platform development environments from a number of marketplace service providers. Since there are a growing number of similar providers, an example of each type of PaaS is represented.

Differentiators among service providers in this segment of the marketplace include the following:

- **Target audience:** Some development platforms are meant for highly technical source code developers, while some support point-and-click browser operations for business users. The developer-oriented service providers will tend to offer items such as source code editors, test execution tools, and compilation and versioning tools, while the business focused platforms will rely on simplified interaction through a common client such as a Web browser.
- **Lock-in:** A key risk of many of the development platforms outlined here is the completely propriety nature of their library interfaces, their database access, and even their development languages. Federal customers investing in development in these environments, should recognize that the code and systems developed here can be locked-in to one vendor's platform and may not operate in any other environment.
- **Installed base:** While some PaaS providers require an entirely new application development effort, others support source code that has already previously been developed in other environments. This can be a significant factor in the reuse of prior investments.
- **Complexity:** PaaS providers vary considerably in the environment's ability to support the development of complex applications. For example, the complexity of an interactive data-rich mapping application is different than a data entry form.

2.1 Salesforce.com™—Example of a Code Development Environment

Application development environments are a popular category of PaaS with vendors including Engine Yard™ [3], NetSuite® [4], and Etelos™ [5]. One of the most well-known PaaS environments for application development is Salesforce.com, with internal infrastructure provided by the *Force.com*™ platform. Salesforce.com has traditionally focused on the capabilities for Customer Relationship Management (CRM) and offers a full suite of tools to develop scalable Web applications for planning, managing, and tracking interaction with large groups of customers.

> *"It's time to stop developing 'here' and running 'there'. Today, most applications are coded in one environment (usually custom-built for that project by a developer), then tested in another, and redeployed to yet another for production... A PaaS should let developers spend their time creating great software, rather than building environments and wrestling with configurations just to make their applications run."*
>
> *– Dave Mitchell, CTO, Bungee Labs [2]*

Force.com provides the underlying infrastructure for Salesforce.com as a run-time platform including a software development environment, a security architecture, and a data store.

Force.com provides an application development environment that uses a unique source code language called Apex, with development support through the site *developer.force.com.* Apex is intended to be used by application developers and is executed on an Apex runtime interpreter as shown in Figure 2.1-1. Apex has a number of pre-built libraries with a mature set of functionality that includes capabilities such as multi-tenant database access, login and security controls, business workflows and approvals, and application and component distribution. Though the libraries and functions can support a range of applications, a strong CRM flavor remains throughout the Force.com site, due to its development history directly supporting Salesforce.com.

Force.com has been actively leveraging component reuse in its environment among its customers, and has set up a forum known as AppExchange to allow capabilities to be reused as shown in Figure 2.1-2. Applications in the AppExchange are integrated with the underlying Salesforce.com cloud infrastructure. Writing on *AppExchange* for O'Reilly, Tony Stubblebine explains, "You can build apps and keep them within your organization—if, for example, you work in a corporate IT department. But if you want to share your application with the world, for profit or otherwise, the AppExchange directory is the answer. It's integrated directly into all Salesforce accounts. If you build an application inside your own account, you can package it directly to the AppExchange directory." [7] This reuse model has been an important factor in creating a large collection of pre-built applications and components in the Force.com environment and some have pointed to this feature as a major contributor to cost savings in this environment.

Database access is provided through the Salesforce.com Object Query Language (SOQL) which is an Structured Query Language (SQL)-like language for structuring queries. SOQL will look familiar to SQL programmers with its SELECT, FROM, and WHERE keywords. However, a number of SQL features such as joins and wildcards in field lists are not supported, and consequently traditional database programmers will have to revisit their approach to building tables and queries.[8] The Saleforce.com database is a multi-tenant object database, and does not follow the traditional table structures common in relational databases. Each object in the database has a number of strongly typed fields, including a pseudo primary database ID key for identifying the

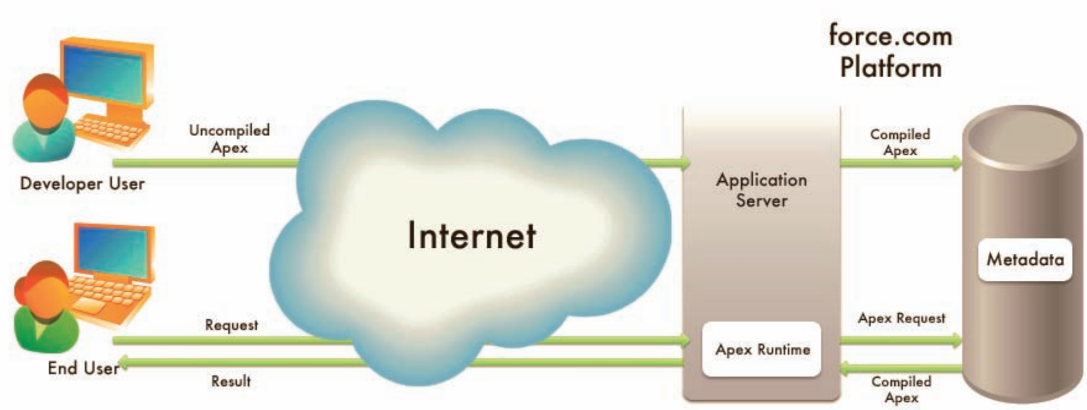

Figure 2.1-1. Apex Code Running on a Force.com Platform Server[6]

object in the entire multi-tenant database collection. Relationships are then used to create one-to-many and many-to-many relationships between objects. Support for database triggers, which provide event-driven database actions, are available in Apex code.

As discussed further in Section 3.0, many cloud providers have decided not to provide the traditional relational and Structured Query Language (SQL)-complaint structures, and instead offer a capability that embraces fast multi-tenant access.

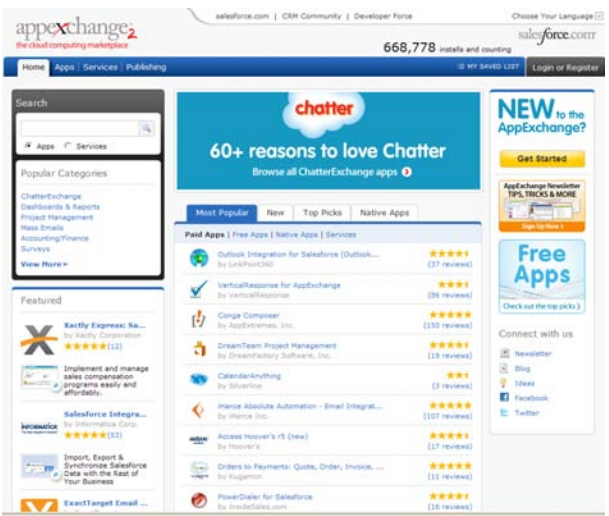

Figure 2.1-2. Using AppExchange to Find Reusable Applications and Components

In summary, Saleforce.com provides a development environment for technical developers to build large scale applications. The source code language and the database language are custom creations, and code that is developed in this environment reflects an investment that can only be currently leveraged within this platform. The development environment is full featured and mature with source code editors, test execution tools, with support for unit testing and automatic compilation. A substantial amount of educational information is available for the PaaS online in many media, and the learning curve is not insignificant.

2.2 LongJump™—Example of a Browser-based Web Application Builder[9]

Another class of PaaS is represented by LongJump, which allows the user to create SaaS offerings by using a simplified set of browser screens in an application building environment. This category of application builder supports the creation of data relationships, forms, and business processes, without requiring the user to be a software code developer. Vendors in this area include BungeeConnect™ [10], IS Tools™ [11], Magic Software™ [12], and Caspio™ [13]. As shown in Figure 2.2-1, part of the value proposition for these types of PaaS is that by making much of the application design intuitive and screen-based, LongJump intends to make the application building process accessible for a much broader set of non-coding consumers. In effect, this can reduce the expertise and projected labor costs to build a Web application, assuming that LongJump's provided application building interface is sufficient for the user's requirements. In addition, the Web application that results is then immediately ready to operate in a scalable cloud infrastructure. (In practice, users have suggested that the non-technical user can do the majority of the application, with occasional help from Java code developers, so the developers may still have a role in this environment, though the developer role is more limited compared to a custom development environment.)

Figure 2.2-1. Browser-based Process, Data, and Form Building[14]

Like SalesForce.com, LongJump provides value to users by offering reusable pre-built components. Writing for ZDNet, Dana Gardner explains, "The catalog offers out-of-the-box but customizable apps for things like relational data management and analysis, form-based applications, resource allocation and management, project fulfillment, approvals and workflow. LongJump provides building blocks of common processes and functions that developers can use to build custom apps to meet the specific needs of their vertical market, without having to reinvent the wheel." [15] The Federal program manager should note though that the interfaces for these components are unique to LongJump, making the systems developed here dependent on the underlying environment. Standards do not yet exist in this PaaS area.

LongJump currently obtains its cloud infrastructure and its *Statement on Auditing Standards* (SAS) 70 II certification from *RackSpace.com*. The SAS 70 II certification documents the effectiveness of the service providers internal information technology controls and data security safeguards. The LongJump architecture is multi-tenant with N+1 redundancy, suggesting redundant components for failover and resiliency, promising that all data and applications are backed up for continuity of operations (COOP).

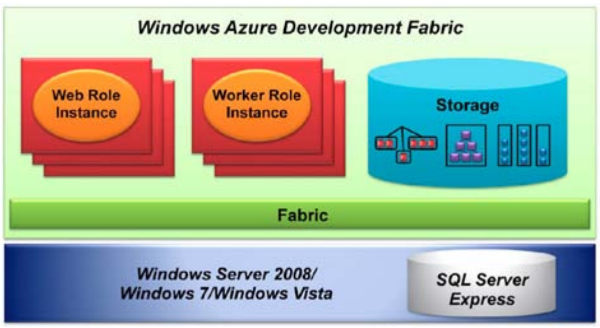

Figure 2.2-2. Coding with LongJump[16]

Though the code developer's interface is normally hidden from the casual application builder, LongJump has provided interface points for Java, Java Server Pages (JSP), and MySQL programmers. Figure 2.2-2 shows the underlying programming architecture for application coders. A Java application programmer's interface (API) is provided for record handling, email and document management, event and task management, transaction management, and other utilities. LongJump supports an extendable and transactional relational database model, with the user's application building screens mapping objects to tables and columns in the underlying system. These data tables can be accessed directly through the Java API. REST inbound interfaces are supported to allow for the submission of XML from external systems, and a SOAP Connect capability is provided for SOAP service integration.

In summary, LongJump offers the non technical business user the ability to build and field straightforward Web applications. Martin Butler Research states, "Applications of moderate complexity can be developed using the menu-driven tools which essentially offer data design, form design and business rules definition capability. More complex applications, or more complex parts of applications which are developed using the forms-based tools can be developed using the raw technology stack – MySQL, Apache, Java, JSR, HTML and XTML."[17] In summary, unlike some of the other development platforms in this section, LongJump has an offering targeted for a non-technical audience.

2.3 Google App Engine™ [18]—An Important Early Innovator in Python and Java

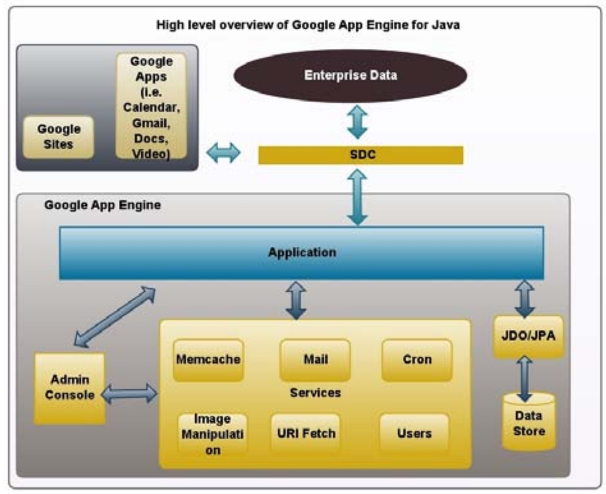

Figure 2.3-1. Google App Engine[19]

Another much discussed development environment and a unique early innovator in the SaaS development area is Google's PaaS know as the *Google App Engine* (GAE). The GAE provides a discrete set of library functions within a sandboxed execution area for developers. Access to the library functions in the App Engine is available in two languages, Java and Python, with Java being a more recent addition to the environment. As shown in Figure 2.3-1, a developer's application in the GAE is supported by some basic library functions. They include the following:

- **Memcache:** This is an in-memory cache used in place of long-term data storage for items such as session data, user preference data, or other data that does not need to be persistent beyond the session. Memcache content is accessed through a key and Google is planning for the interface to be compliant with the draft standard

Java Specification Request (JSR) 107, the Java Temporary Caching API specification.

- **Mail:** This interface supports the sending and receipt of email with optional multipurpose Internet mail extension (MIME) encoded attachments by an application.
- **Cron:** This library supports the scheduling of recurring tasks, such as daily reporting.
- **Image Manipulation:** This library has some basic image manipulation functions including resizing, rotating, flipping, and cropping images. The library can also provide histogram and sizing information on images.
- **URI Fetch:** This capability supports the use of HTTP and HTTPS to external applications. The common HTTP functions of GET, PUT, POST, HEAD, and DELETE are supported, enabling external Web application integration.
- **Users/Google Accounts:** This interface supports the authentication of users with Google accounts.

While Google uses the term sandbox in describing the functions and access that an application has, it is not exactly the same as the traditional Java sandbox. Many features of Java that a developer might be used to executing are not available in the Google sandbox. Google's developer site explains, "Applications run in a secure environment that provides limited access to the underlying operating system. These limitations allow Google's App Engine to distribute web requests for the application across multiple servers, and start and stop servers to meet traffic demands. The sandbox isolates your application in its own secure, reliable environment that is independent of the hardware, operating system and physical location of the web server."[20] For the software developer, the inability to read files from the file system or spawn long-lived subprocesses could be an important architectural consideration in designing effective applications for GAE.

As GAE for Business was announced in May 2010, how the marketplace will react to the GAE remains to be seen. The GAE provides a limited set of supportive functionality, and like other service providers has potential for vendor lock-in if the provided APIs are interwoven in a developer's application. The platform has very large scaling potential for SaaS application developers and has traditionally had a competitive price point.

2.4 Microsoft Azure Services Platform™ [21]—Potential Legacy Compatibility to a Broad Installed Market

Microsoft's Azure is an important and unique cloud offering to watch, because it will potentially support a very large installed base of contemporary IT systems and their developers. Mary Jo Foley explains, "With Azure, Microsoft is attempting to recreate its Windows ecosystem in the form of a utility. Today, developers and customers can develop and deploy on the Windows Azure operating system and make use of the SQL Azure hosted database."[22] As shown in Figure 2.4-1, Azure provides a convenient transition path for many developers, because it provides familiar tools both for application development and operation, including Visual Studio extensions, and the use of SQL-based relational databases.

The Microsoft architecture is built upon virtualization and supports two primary environments or virtual machine roles when running in Azure—the *Web* role and *Worker* role. The Web role is intended for operational environments where the software application will need to provide Web services such as HTTP communications to the outside world, and therefore a Microsoft Internet Information Services (IIS) Web server and HTTP load balancers are available in the binary image in this run-time environment. In the Web role all load balancing is stateless and the application developer must adjust their programming strategy to accommodate this architectural approach, using persistent storage as needed.

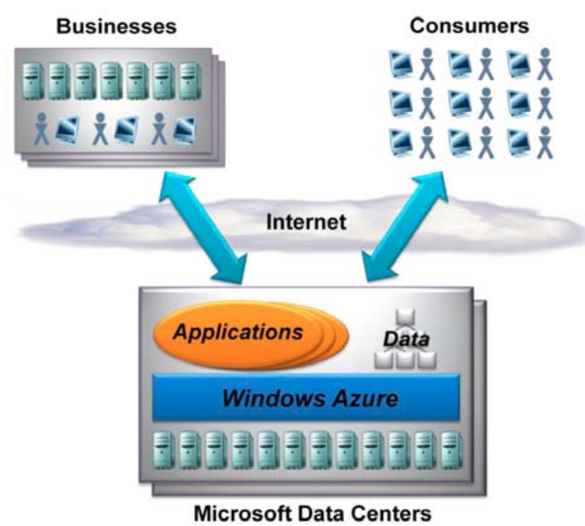

Figure 2.4-1. Windows Azure in Microsoft Data Centers[23]

In contrast, a Worker role does not include IIS and is intended for intensive application processing that occurs entirely within the cloud environment. An XML configuration file defines to the Azure control fabric the required Web and Workers to fulfill an application's needs.

Azure's architecture accounts for several types of data storage with differing characteristics. First, SQL Azure provides a fully relational SQL-based database service offering, familiar to MS SQL Server programmers, as described further in Section 4.4. Second, Azure allows for the storage of binary large objects (BLOBs) which can represent anything from data files to whole logical *XDrives* file systems. Third, Azure has a queuing function available to send asynchronous messages between applications in any machine role and this can be very useful for systems integration. Finally, Azure provides highly scalable non-relational table structures as well. This broad and multi-faceted approach supports a wide variety of programmer and architectural requirements. Microsoft's David Robinson explains, "With SQL Azure, developers building Web 2.0, ASP.NET and PHP applications can use familiar tools and data models to develop on a pay-as-you-grow, secure, scalable and highly available database service at minimal infrastructure cost." [24]

Microsoft provides development tools to support the remote application development process. For example, the Azure environment supports both the managed code inherently produced by Visual Basic .NET and C# compilers, and unmanaged code produced by other compilers which don't have Intermediate Language (IL). Microsoft also provides a tool to support the local development of applications intended to execute in the cloud, known as the *Azure Development Fabric* as shown in Figure 2.4-2. David Chappelle explains, "One obvious difference between the cloud and on-premises worlds is that Windows Azure applications don't run locally. This difference has the potential to make development more challenging. To mitigate this, Microsoft provides the development fabric, a version of the Windows Azure environment that runs on a developer's machine." [25]

In summary, Microsoft has provided a development environment that is friendly to developers of legacy software systems, while still offering new services such as cloud-focused multi-tenant non-relational databases. The developer toolsets are comprehensive and support the remote development of cloud applications. The pay as you go model when combined with familiar development tools and databases could be a motivating factor for many legacy programs considering a cloud software release.

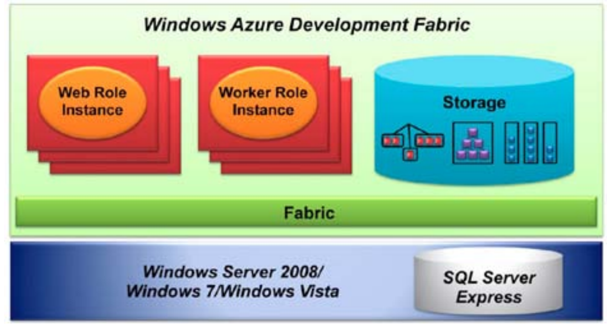

Figure 2.4-2. Windows Azure Development Fabric[26]

3.0 Database Platforms

This section discusses the market segment increasingly referred to as Database as a Service (DaaS). DaaS, a form of PaaS, is currently found in the marketplace in three broad capabilities—online general relational databases, non-relational databases, and the ability to operate pre-defined virtual machine images loaded with common open source databases such as MySQL or similar commercial databases. These approaches give a wide range of capability and potential complexity.

> *"Cloud computing is disrupting traditional database economics. By leveraging multi-tenancy—the ability to share a storage infrastructure across multiple users—the cost of a large-scale storage infrastructure can be amortized across multiple companies. This replaces the fixed costs (Capital Expenditures) and variable operational costs (Operational Expenditures) inherent in an in-house data center, with a variable cost-only economic model."*
>
> *– Mike Hogan, CEO ScaleDB[27]*

Many of the cloud DaaS solutions in this section are unique to a particular vendor and do not follow conventional standards for schema description or query languages. This raises the concern of vendor lock-in once substantial databases are created with significant investment, and lowers the ability to leverage prior schema development work and entity relationships in some situations.

Current market differentiators in this cloud service category include items such as:

- Use of the relational database model in the database
- Ability to use and roll back transactions for database integrity
- Use of proprietary versus standards-based query languages
- Ability to use database schemas and data types
- Data update propagation patterns across distributed implementations

3.1 Machine Images with Existing Database Server Engines

One of the most conceptually straightforward ways to put a database in the clouds is to lease a bare server and to load a machine image of an existing database engine. Many vendors including Rackspace™ [28], Terremark™ [29], and GoGrid[30] offer this type of cloud-based bare virtual server for lease. For example, Amazon EC2™ has pre-made machine images for IBM DB2™, Microsoft SQL Server™, MySQL™, Oracle™, PostgreSQL™, Sybase™, and Vertica™. This method relies on leasing the underlying computational and storage resources from an infrastructure-as-a-service (IaaS) provider and retaining the expertise to load and remotely operate the database from within your organization. The benefit of this general approach is the complete control it affords the service consumer to configure the database as they will. The downside of this approach is that with control comes total responsibility for characteristics such as physical tuning, licensing, failover, replication, and continuity of operations (COOP). In contrast, several of the DaaS options whose descriptions follow in this section resolve these types of architecture capabilities on behalf the service consumer.

MySQL is a notable option in this machine image category due to its common use in the LAMP software bundle stack, and its GNU General Public License. Currently, a number of vendors including Amazon, Elastra, and Rightscale offer MySQL images on their infrastructure offerings.

3.2 Amazon SimpleDB™ [31]— Example of a Non-Relational Cloud Database

Like Google Apps' BigTable™, Amazon's Simple DB is an example of a non-relational cloud-based database. It works on a simple design premise, with a key/value pairing architecture, and has the benefit of being able to leverage the dynamic computing power and storage underlying Amazon's offerings. The Amazon web site explains, "For database implementations that do not require a relational model, and that principally demand index and query capabilities, Amazon SimpleDB eliminates the administrative overhead of running a highly-available production database, and is unbound by the strict requirements of a RDBMS. With Amazon SimpleDB, you store and query data items via simple web services requests…" [32] SimpleDB offers benefits for programmer consumers who leverage its simplified application programmer interface (API), but does not have higher order capabilities such as database transactionality.

A SimpleDB consumer sets up a domain which includes a set of value/attribute pairs. In effect this data store is structured like a set of columns associated with a single key. SimpleDB is designed in a collection of services that follow the general design pattern of a Web service request/response. The services include *PutAttributes, GetAttributes, DeleteAttributes,* and *Select.* The PutAttributes and GetAttributes services allow the data records to be updated and retrieved respectively. The Select is a form of a data set query function structured in a proprietary format, not adhering to a standard such as Structured Query Language (SQL). Attributes are always strings (e.g., UTF-8 text), which translate well into the underlying textual service exchanges. Note that binary information, dates, and numeric data do not have native formats in this architecture.

An additional benefit of the SimpleDB is the multi-site operation built into the infrastructure of the service offering. Currently, for reasons of performance and network topology, SimpleDB is offered in particular regions, which equate to physical data center locations. The current list of regions includes US-East (Northern Virginia), US-West (Northern California), and EU (Ireland). The consumer of the SimpleDB services selects a region for a particular data set when it is instantiated. Amazon states that, "Amazon SimpleDB stores multiple geographically distributed copies of each domain to enable high availability and data durability."[33] Consequently, SimpleDB can update data in two consumer-selectable methods. The Eventually Consistent Reads mode enables a faster data store update operation, but can lead to momentary data integrity issues until the changes are propagated across all the geographic instances. The Consistent Reads mode trades off slower update performance for consistency in queries. The consumer of the service has to understand the application's requirements in order to select most appropriate update mechanism.

SimpleDB is a minimalist solution for particular kinds of straightforward database problems. The downside of its simple structure and API is that some capabilities common to a relational system would be hard to implement, and would cause some amount of re-inventing common capabilities. For example, complex queries, transactions, multi-table joins, query result sorting and other functions will fall on the consumer's programmer to develop to use this capability. In the correct situation though, this is a highly available, redundant, and simple to use tool, with service access based on open SOA Web Service standards.

3.3 Google Apps BigTable[34]—Example of a Non-Relational Cloud Database with Transactions

Like SimpleDB, Google Apps BigTable is a non-traditional, non-relational database service offering yet it has some additional functionality. BigTable was originally built as an internal tool for Google, and works well to solve the types of problems that Google frequently confronts in the parallel searching of massive databases. Writing for *TechCrunch*, Mark Hendrickson sums up BigTable's role as follows, "Google started development on BigTable in early 2004 and began using it actively in February 2005. The non-relational, proprietary system was designed internally to fulfill Google's peculiar need for access to massive amounts of data at very high speeds (millions of read/writes per second). BigTable is based on the Google File System (GFS) and designed for distribution across thousands of commodity servers that collectively store petabytes of data. Services that rely on it include Google Search, Google Earth and Maps, Google Finance, Google Print, Orkut, YouTube, and Blogger."[35] BigTable became available externally to Google in 2008 as part of the Google App Engine.

Google, like Amazon, has created a proprietary query language, referred to as GQL, which is analogous to SQL, though with a reduced feature set. Similar to SimpleDB, the GQL does not support the concept of a multi-table join, because the data found in their data store is de-normalized. The big table can be thought of as a large sparsely filled matrix.

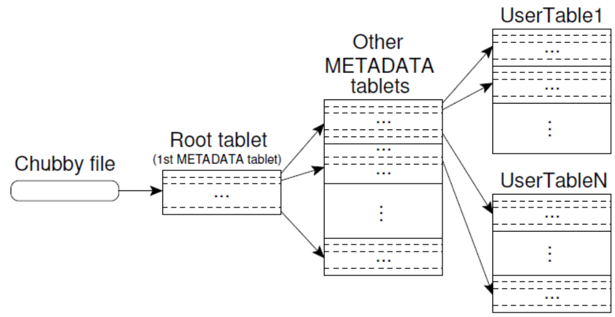

Figure 3.3-1. BigTable—Tablet Location Hierarchy[36]

Google has a complex internal table management method as shown in Figure 3.3-1. This is largely hidden from the consumer programmer, but aids in understanding the exposed functions. David Hakla explains, "In order to manage the huge tables, Bigtable splits tables at row boundaries and saves them as tablets. Each tablet is around 200MB, and each server saves about 100 tablets. This setup allows tablets from a single table to be spread among many machines. It also allows for fine-grained load balancing, because if one table is receiving many queries, it can shed other tablets or move the busy table to another machine that is not so busy. Also, if a machine goes down, a tablet may be spread across many other machines so that the performance impact on any given machine is minimal."[37] The ability to divide the search problem across a collection of machines, each built from commodity hardware, provides advantages to Google when searching very large data sets.

APIs for the Google data store are provided in Java and Python. In either API, the data store is "schemaless" meaning that it imposes no typing or structure on the data stored. The application programmer must ensure that typing occurs in the application logic, outside of the database itself.

Unlike SimpleDB, Google's data store has allowed for the concept of a transaction. A collection of data store operations can be performed as a group, thus ensuring that they all succeed or fail together. This is an important feature for maintaining the data store integrity for more complex sets of application data.

3.4 Microsoft SQL Azure™ [38]— Example of a Relational Cloud Database

Many vendors like Amazon™, with their Relational Database Service (Amazon RDS)™ [39], and Xeround™ [40] offer relational Structured Query Language (SQL)-based database engines as a PaaS. For example, Microsoft is offering a Structured Query Language (SQL)-based relational database in the cloud with its *SQL Azure Database*. SQL Azure's similarity with existing Transact-SQL products is a clear benefit to a development community that relies in large part on traditional relational models of data sets and hopes to model normalized entity relations in its data stores. The Microsoft cloud database model should look familiar to users of MS SQL Server, and a high level of compatibility is suggested between the products. For example, familiar concepts such as tables, schemas, indexes, views, stored procedures, and triggers can be found in the online service.

Like the SimpleDB and BigTable examples, the SQL Azure database is designed to be fault tolerant by use of the underlying Microsoft cloud infrastructure to host copies of data on multiple servers. Further, the Microsoft model allows for the physical region of the hosting to be selected, homing the data to select physical data centers. Interestingly, Microsoft requires databases to be under 10 GB per instance, with multiple instances on multiple machines being described as database sharding.

Figure 3.4-1. Windows Azure Platform

The use of the remote service in a cloud paradigm removes some prior concerns of MS SQL server administrators. For example, Microsoft writes, "Unlike administration for an on-premise instance of SQL Server, SQL Azure abstracts the logical administration from the physical administration; you continue to administer databases, logins, users, and roles, but Microsoft administers the physical hardware such as hard drives, servers, and storage. … SQL Azure automatically replicates all data to provide high availably. SQL Azure also manages load balancing and, in case of a server failure, transparent fail-over. To provide this level of physical administration, you cannot control the physical resources of SQL Azure." [41] By giving up control of physical administration details, the service consumer then relies on Microsoft to control a high availability architecture.

SQL Azure differs from the SimpleDB and BigTable implementations in that it supports traditional schemas and data typing. Figure 3.4-2 depicts the wide range of basic data types available. However, note that a smaller set of usual geography and geometry types have been removed.

Data Type Category	SQL Azure Support
Exact numerics	Supported: **bigint**, **bit**, **decimal**, **int**, **money**, **numeric**, **smallint**, **smallmoney**, **tinyint**.
Approximate numerics	Supported: **float**, **real**.
Date and time	Supported: **date**, **datetime2**, **datetime**, **datetimeoffset**, **smalldatetime**, **time**.
Character strings	Supported: **char**, **varchar**, **text**.
Unicode character strings	Supported: **nchar**, **nvarchar**, **ntext**.
Binary strings	Supported: **binary**, **varbinary**, **image**.
Spatial data types	Supported: **geography**, **geometry**.
Other data types	Supported: **cursor**, **hierarchyid**, **sql_variant**, **table**, **timestamp**, **uniqueidentifier**, **xml**.

Figure 3.4-2. SQL Azure Database—Supported Data Types [42]

4.0 Middleware and Integration Platforms

> *"Going forward, PaaS may find a nice niche in inter-company problem domains, meaning that it's easier to create shared applications on a PaaS that exists between the organizations, versus creating application on one physical platform on another."*
>
> *– David Linthicum [43]*

In an enterprise portfolio of IT systems, integration services can take many forms. Historically, integration took place in pre-defined formats, such as the text-based electronic data interchange (EDI) documents, over value-added networks. These service types can still be found in cloud-based platform services today. However, the integration definition has expanded to include services for message queues, business process modeling, service-oriented architecture (SOA) wrappers, and a wide variety of application specific connectors. This section gives examples of the wide variety of types of integration services in the cloud platform marketplace.

Since many of the integration functions discussed in this section are inherently vendor or solution specific, tying together specific vendor products, the reader should consider the risk of vendor lock-in when investing in the enterprise use of these service solutions. Due the nature of the integration problem being tackled, this risk may be unavoidable in some degree.

4.1 Appian Anywhere™ [44]—Example of a Cloud Business Process Modeler

Business process modeling (BPM) is an important part of any substantial IT system capability and often fulfills an integration role among multiple data sources or application systems. BPM supports workflow through an organization's business processes and enables process checkpoint and approval mechanisms. Several firms including Iceberg™ [46] offer cloud-based business process tools. Appian offers a BPM tool from the cloud, meaning that it is available across the public Internet as a set of configurable services able to connect to a variety of systems. Though the company interchanges the use

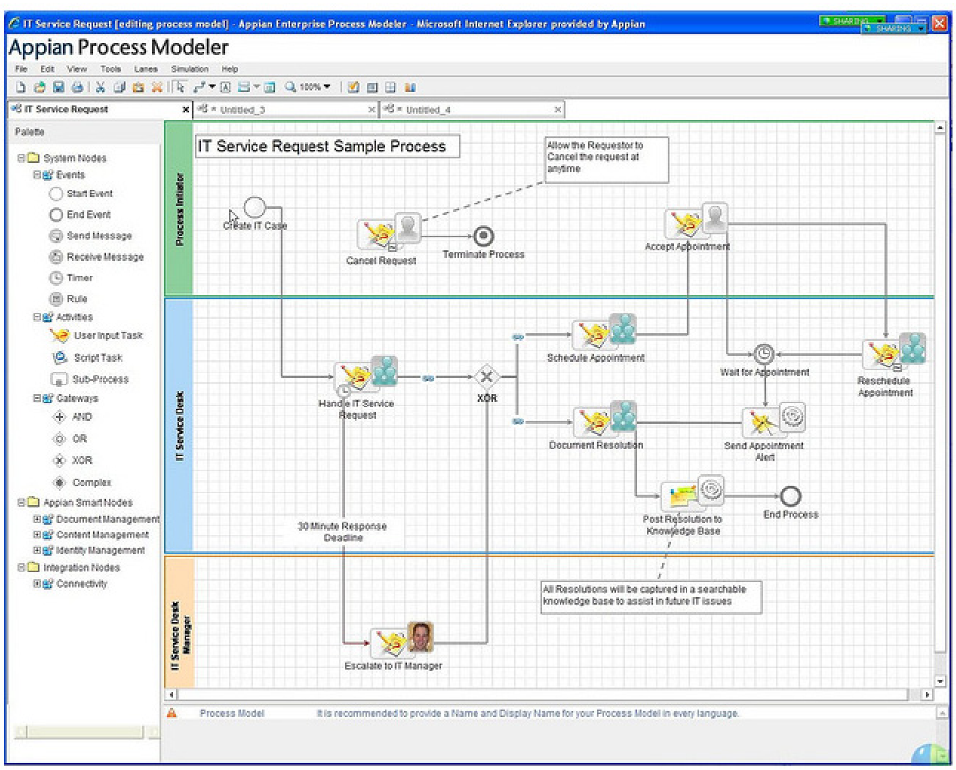

Figure 4.1-1. Appian Browser-based Process Modeler [45]

of the terms "SaaS" and "platform" in their product descriptions, we have included the platform aspects of the BPM services in this report, because they are an important enabler for organizations building and offering SaaS applications on the Internet.

Appian Anywhere is the on-demand Appian-operated version of the company's central BPM tool, which is also available as a three-tier Web application that can be run privately inside a customer's data center. Though the basic online tool shown in Figure 4.1-1 is a platform that can be customized to provide end-users with a SaaS, Appian has created instantiations for particular vertical problem sets such as Federal acquisition processing, human resources processes, case and grants management processes, and micro-purchasing. An Appian-sponsored *Datamonitor* whitepaper states, "The presence of a large number of SaaS-based applications creates the necessity for code free integration enabled through major pre-configured web services. As business users are increasingly exposed to services delivered over the web, they will start expecting BPM to help them seamlessly integrate SaaS applications into their processes."[47] The online tool offers series of functions in addition to core BPM, and they include report generation and collaboration tools.

The core of the Appian product is the process modeler which is Web-based. The process modeler provides a graphical interface to allow a process designer to layout and test workflows. In support of these workflows, Appian Anywhere uses a shared in-memory data area known as KDB to tie together information to support a business process flow. BPTrends writes, "Kdb is a non-proprietary commercial database platform that is optimized for real-time, in-memory performance [...] Traditional approaches that try to combine a streaming or in-memory database with a reporting or OLAP database can't deliver the performance necessary for real-time business. [...] Appian's use of the Kdb database platform prevents this time gap from occurring."[48] Appian also has a forms designer which allows the modeler to build screens for data gathering and presentation.

4.2 Amazon Simple Queue Service (SQS)™ [49] —Example of a Message Queue Service

Message queue capabilities have been an important integration tool for enterprise architects for decades because they support a common design pattern for the asynchronous exchange of information between software applications that do not need to be in direct communication at the same moment. For example, one application can leave an addressed message, and another can pick it up at a later time. The approach is flexible because the timing of the exchange, and how often an application checks or posts to its queues, is a tunable parameter for most application developers, and consequently, message timing can range from seconds to hours based on application requirements. Queues give an additional buffering benefit in some situations, accommodating uneven or 'bursty' production and consumptions rates of the source and destination applications. Often in a Federal enterprise, groups of applications will elect to use the same system-to-system message queuing function, and in some organizations this eventually becomes an enterprise-wide function for application message exchange.

Several firms such as Amazon™, OnlineMQ™ [50] and StormMQ™ [51] offer cloud message queue services. For example, Amazon provides the SQS as a basic message queue service that can create queues and post and retrieve messages among many participants. In keeping with their general design approach, Amazon provides Web service SOAP interfaces for the SQS functions. Figure 4.2-1 depicts the message queue in relation to the service consuming applications. With SQS messages are guaranteed to be delivered at least once, though not necessarily in the order they were sent. Queues support many readers and writers simultaneously. Like other Amazon services, queues can be defined in particular general physical regions, such as the US East (Northern Virginia), US West (Northern California), EU (Ireland) and Asia Pacific (Singapore) regions.

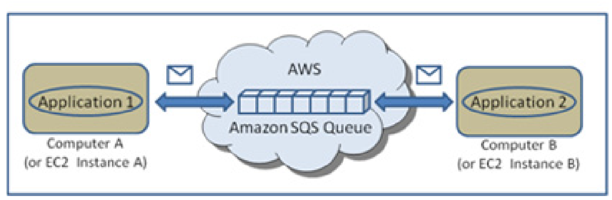

Figure 4.2-1. Amazon SQS—Enabling Message Exchange Between Applications[52]

Queues offer interesting design pattern options for architects of systems-of-systems. For example, queues can enable the development of asynchronous business processes distributed across the cloud's network, across a series of mission-related systems. Applications can assign work to each other, and a series of queues can form an assembly line of waiting work packages. SQS's SOAP and HTTP interfaces will allow it to be called from just about any contemporary development environment, noting that the authentication mechanisms rely on the related Amazon account system.

Due to the underlying scalability of Amazon's EC2, and its ability to launch additional applications, measuring attributes of the queue itself can be used to automatically scale the applications working on the queue content.[53] For example, the length of the queue and the time that objects spend in the queue can be good indicators of the health of the overall application system. Amazon explains, "Because each message in the queue represents a request from a user, measuring the length of the queue is a fair approximation of the load on the application. By trial and error, you can determine the optimal length of the queue (the length at which you have just the right number of servers running to cover the demand). At any time, if the current length of the queue exceeds the optimal length, then you should start an additional server instance."[54] Using this conceptual approach, responding to dynamically changing demand can become an automated function of a system.

4.3 Hubspan™ [55]— Example of an Integration Hub

System to system integration comes in many forms, ranging from business process integration to message exchange and translation. Like Pervasive's *Integration Hub*™ [56] and Virtual Logistics' *Hosted EDI Services*™ [57], Hubspan provides a cloud-based version of what was traditionally called a value added network (VAN). The purpose of a third-party integration broker is to assure messaging between the participating systems, offering non-repudiation of the message content, and assuring delivery of the messages to their intended recipients. Messaging in this context means the delivery of content between software applications, usually in an asynchronous manner in which timing is not urgent.

Often the sender and receiver of messaging traffic do not recognize the same format messages and underlying protocols, and a mapping is established between the incoming and outgoing formats. This translation is often an expensive labor-intensive task

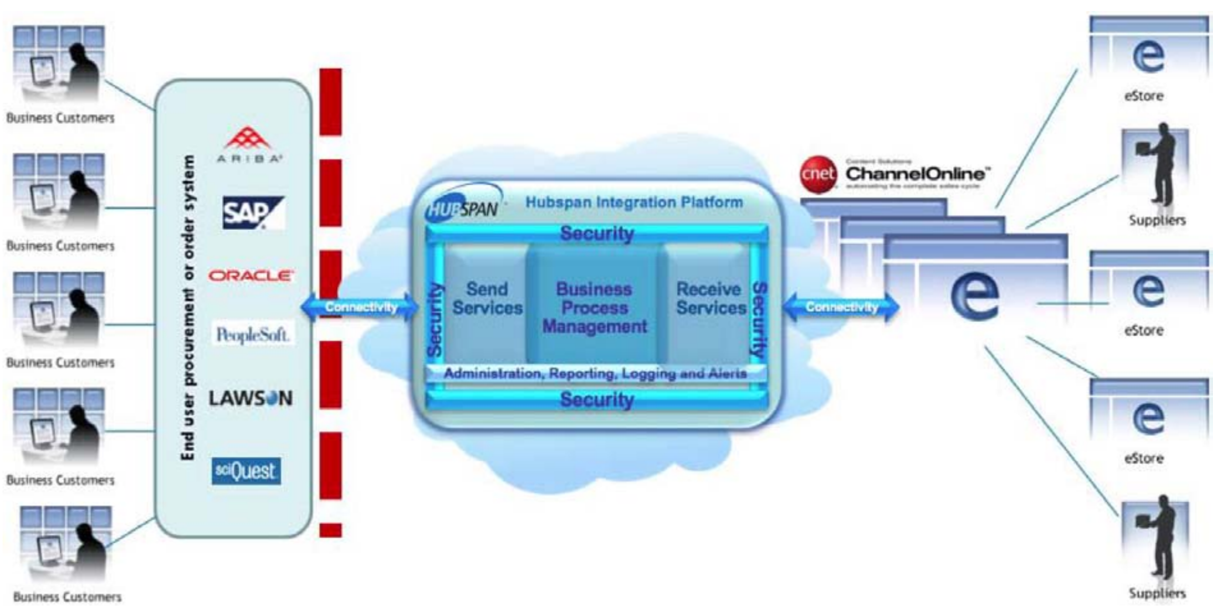

Figure 4.3-1. Hubspan as an Integration as a Service Platform[59]

Protocols Supported	Message Formats Supported	E-Procurement Systems Supported		Sample Processes (and many more!)
• HTTP/S • FTP/S • SFTP • SMTP • AS2 • RNIF • EDI VAN • REST • SOAP	• **PunchOut** (cXML PunchOut, OCI RoundTrip, Oracle XML, TapOut) • **EDI** (ANSI X12, EDIFACT, Spec2000, Tradacoms, etc.) • **Flat File** (CSV, ForcedWidth, MultiRecord, etc.) • **XML** (cXML, OAGIS, UBL/ ebXML, customized, etc.) • **HTML Forms** • **Vendor Specific** (OCI, CIF, iDOC, etc.)	• Ariba • Infor • Epicor • ePlusB2B • Exostar • MRO • Oracle • PeopleSoft • Perfect Commerce	• SAP • SciQuest • Science Warehouse • Market4Care • Unity • SupplyNet • Ketera • Oracle Exchange • IBM ShopOut	• Product availability • Price check • Order status • Shipment scheduling • Shipment status • Product tracking • Code lookup • Contract status check

Figure 4.3-2. Protocol and Message Integration Platform[60]

to do the first time, and having a service capability with many existing pre-built translations can reduce overall integration costs for a program. ThinkStrategies' Jeffrey Kaplan adds, "Hubspan offers a multi-tenant integration platform which extends beyond basic mapping and simple translation capabilities within corporate departments to address the growing challenges of integrating on-demand applications across and between enterprises." [58] This capability goes beyond a message box or a queue because the service inspects the structure and content of the messages, often applying business rules in the transformation of content before delivery. Figure 4.3-1 depicts an example scenario where Hubspan platform capabilities are used to connect a number of SaaS and back-end systems. An underlying addressable network that connects all parties is a basic assumption in this scenario.

The value in an integration hub or VAN is in part dependent on the number and types of protocols and message formats that are supported in the mapping and translation engine, which is the core of the service. Figure 4.3-2 enumerates the types of interactions that Hubspan can support. Please note the wide range of message types, from the more recent cXML PunchOut exchanges to the legacy ANSI X12 EDI interchanges. The key to selecting a service platform for integration is understanding the potential and likely message exchange requirements for the systems with which an organization intends to integrate. This is a requirements generation task focusing on integration requirements. Most vendors will support the addition of new protocols and customer-unique messages, though often this drives cost, as compared to reusing existing translations.

Systems integration is increasing both in complexity within organizations and across external organizations. We can expect this trend to continue as we combine greater numbers of data sources to provide higher value information. Collections of enterprise systems of reasonable complexity will almost certainly need some type of translation and integration engine of this class, offering non-repudiation of message content and assured message delivery functions. Often this type capability becomes an enterprise-wide offering since it crosses so many functional domains. For example, the Government-run legacy system Global Exchange Service (GEX) fulfills this role for both DoD and Federal customers.[61]

5.0 Other Notable Platform Examples

"The idea is not to force-fit IT onto cloud computing platforms; it's about rebuilding the core IT infrastructure as many components parts: data, services, processes, images perhaps bundled into virtual appliances that could be portable among cloud platforms."

– David Linthicum [65]

5.1 Amazon Elastic MapReduce™ [62] — A Hadoop Framework Service

Another key use of cloud computing is the capability to leverage a very large collection of processing power on demand, as needed, for big computational problems. For example, Amazon's *Elastic MapReduce* is a Hadoop framework that operates on Amazon's underlying infrastructure including Elastic Compute Cloud (EC2)™ and Simple Storage

Service (S3)™. Competition in this area includes Softlayer™ [63], who offers a pre-built Hadoop machine image. The Hadoop framework is a platform that enables work on large data-intensive or computationally-intensive problems across a substantial pool of processor and/or storage resources, by dividing up a problem into many repeatable smaller jobs and aggregating the processing results. Amazon's Web site explains, "Amazon Elastic MapReduce automatically spins up a Hadoop implementation of the MapReduce framework on Amazon EC2 instances, sub-dividing the data in a job flow into smaller chunks so that they can be processed (the "map" function) in parallel, and eventually recombining

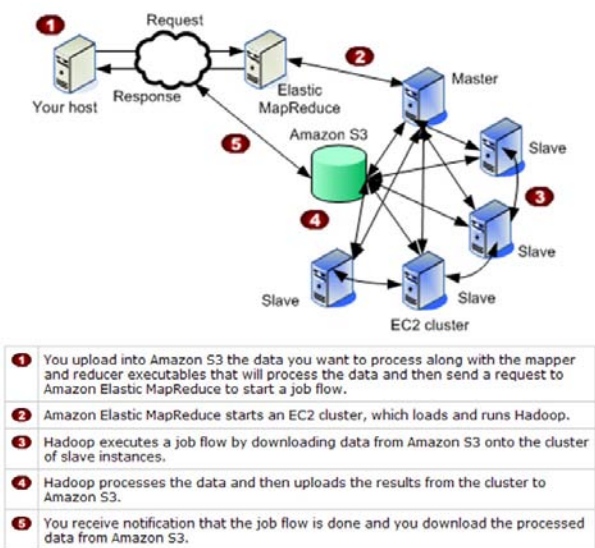

Figure 5.1-1. Using Amazon's Elastic MapReduce[66]

the processed data into the final solution (the "reduce" function). Amazon S3 serves as the source for the data being analyzed, and as the output destination for the end results." [64] This general process for employing the Elastic MapReduce is depicted in Figure 5.1-1 from Amazon's *Developer's Guide.*

Elastic MapReduce handles many of the details in establishing and terminating the Hadoop framework on behalf of the service consumer, and can utilize both the Hadoop Distributed File System (HDFS) and the Amazon S3 Native File System (S3N) for data access. Leon Katsnelson notes, "Hadoop is the kind of application that, at least on the surface, is a natural fit for the elastic nature of cloud computing. Instead of procuring large computing clusters one can just go to Amazon to run a job and pay only for the resources use by that job." [67]

A key consideration for many programs in using a service like this, which is often most valuable with very large data sets, is how to initially get those data sets into the vendor's operational environment. For most vendors a special set of service charges, often called "transfer charges," apply for use of the network in addition to the storage and computational billings that one might expect to receive. Leon Katsnelson states, "…running such a job will require transfers of very large volumes of data in to and out of the cloud. And, while compute charges on EC2 and storage charges on S3 are quite low, data transfers charges can really add up." [68] In fact, for very large data sets, some vendors have provided for the physical shipment of drives, by couriers such as Fedex, in order to reduce network-based transfer charges. Obviously this tradeoff depends on the timing and size of the data set in question.

Amazon has established a REST-based Web service API and SOAP Web Service Definition Language (WSDL) interfaces for queries into the Elastic MapReduce capability. Public Key Infrastructure (PKI) digital signatures are also available to protect data in the service calls. Service endpoints, code samples, and sufficient developer information are defined in the online *Developer's Guide.*

5.2 SOASTA™ [69]— Example of Cloud Computing Testing

SOASTA is a commercial firm uniquely focused on applying cloud computing resources to the on-demand testing of Web-based software such as SaaS. SOASTA's CEO Tom Lounibos states on his blog, "SOASTA led the innovation around cloud computing by using the cloud to test applications, deployment platforms, and end user experience. The founders realized early on that the best way to simulate real world conditions of unpredictable and spiky traffic, which requires enormous amounts of servers to replicate, is through the cloud. […] It turns out that testing today's applications, with their dynamic content, distributed architectures and millions of users, is a very big data problem, which makes testing, difficult, and in many cases impossible, if not for on demand services such as SOASTA

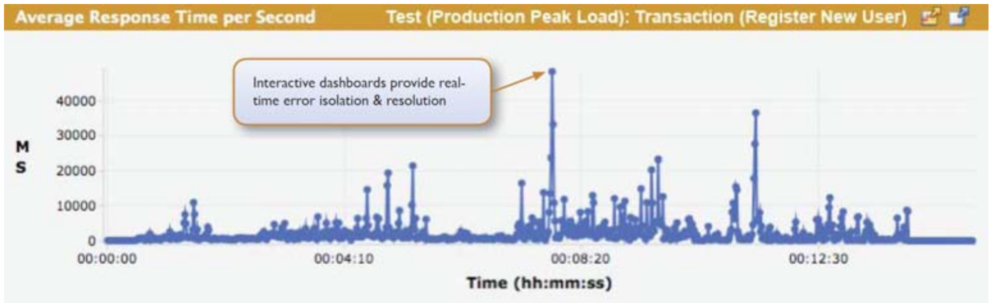

Figure 5.2-1. Response Time Summary from SOASTA[71]

CloudTest."[70] SOASTA's offerings support network-based testing with an emphasis on load testing. Their basic capabilities mirror an automated testing process with test development and recording, test management, playback, and data results analysis as shown in Figure 5.2-1.

Part of the value proposition for SOASTA's approach is that they can leverage an on-demand infrastructure for the testing capability, launching as many resources as required to complete stress testing, performance testing, or load testing. In addition, they have the results loaded into an OLAP data store which supports the deeper analysis of the test results by a virtual team and allows for large quantities of test data to be collected.

SOASTA eliminates the need to set up dedicated hardware servers for testing and can leverage Amazon's EC2 as needed. An example Amazon case study details, "Intuit used SOASTA's proven best practices for testing what, for them, had been a previously unreachable number of virtual users. SOASTA employed an iterative approach starting with a 1,000-user test. Generating load from Amazon EC2, they ramped the volume with each subsequent test—finally reaching the test goal of 200 percent of their anticipated peak—all while real customers accessed the production site at the same time. […] During the testing process, the Intuit and SOASTA teams located, fixed and optimized issues with the application, load balancer settings, and log in, among other components. Soon, Intuit was confident that every component of their application was ready to go for tax day."[72]

The suite can support SOA-based direct service testing as well as SaaS application testing, with or without remote agents. Agents can be required in cases where network access is restricted due to policy or by device configurations such as restrictive firewall rules. Timing of protocol interactions, such as HTTP request/responses to the browser, can be recreated as needed to mimic interaction with user or application-oriented software.

5.3 Zuora™ [73]—Example of Billing and Payments as a Service

Though it may be more difficult to think of billing and payments as a service in a traditional Federal Acquisition Regulation (FAR)-focused environment as compared to other platform services in this paper, several firms are offering these services as a way to recoup cloud costs in the commercial marketplace. In fact, the Government often needs

Figure 5.3-1. Example Billing Services in Relation to Accounting and Customer Support[74]

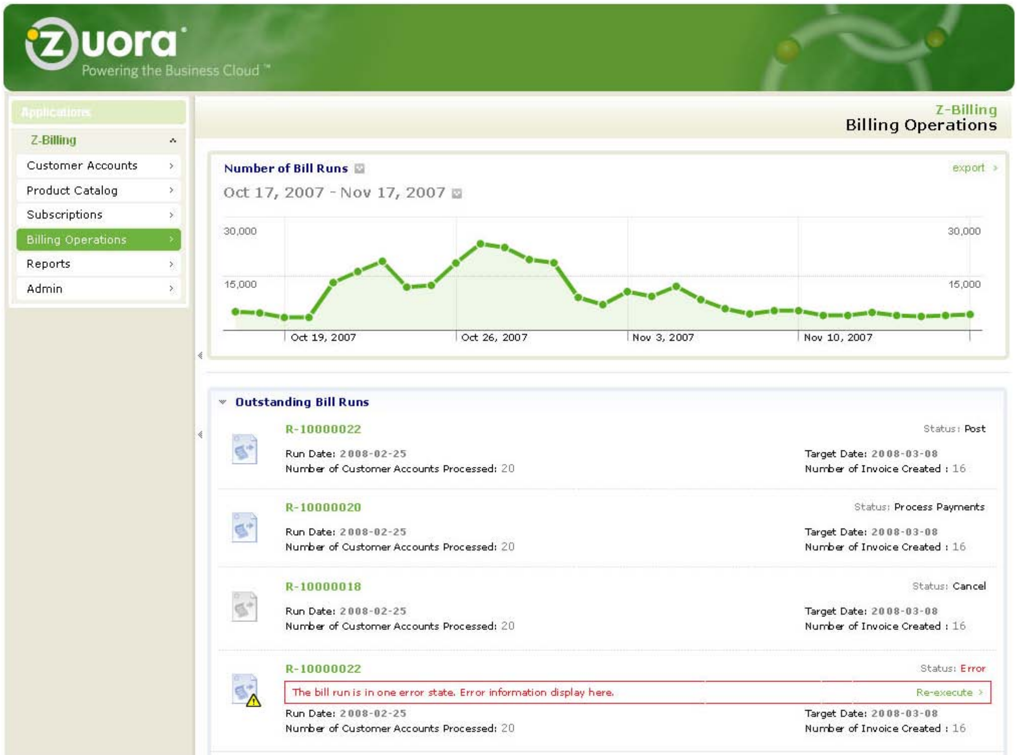

Figure 5.3-2. Example Cloud Billing and Accounting Platform[79]

to perform some form of cost recovery for its IT services and this category of billing and payments services could be applicable to Government providers who desire to provide services to other parts of the Government, either through catalogs or as an IT line of business (LOB).

Like Apprenda™ [75], and Evapt™ [76], Zuora provides a number of on-demand cloud services to support billing, subscriptions, and charge backs for online services. The Zuora services are applicable to Government IaaS, PaaS, and SaaS if applied in the correct context, with a supporting legal and policy framework. Beagle Research states, "As companies—especially technology companies—liberate themselves from the need to build custom billing systems, the internal resources initially dedicated to billing system development, maintenance, and operation can be deployed to core business opportunities. […] Cloud-based billing and payment solutions improve back-office business processes that have too often been afterthoughts for cloud-based companies." [77] Figure 5.3-1 depicts an example Zuora product line, showing the relationship between billing and subscription service to the underlying accounting and customer support systems.

The Zuora services are implemented as a collection of traditional SOA services, with support for one-time, recurring, usage and metering charges, and volume tier pricing. As shown in Figure 5.3-2 the platform provides support for billing, billing disputes and invoicing. The Zuora approach aligns with standards such as SOAP 1.1 and WSDL 1.1, and the specifications for the service calls are detailed in a developer support site.[78] Integration with Salesforce.com and several market leading accounting packages is provided. In addition account histories and account metrics can be generated from a Zuora data store.

6.0 Conclusion

The cloud computing PaaS market is a healthy marketplace with a wide range of service offerings covering key functions desired by an architect for enterprise solutions. As the number of service providers grows, the options to the IT decision maker are expanding. We can expect the market to differentiate further in the coming year. By our informal count, during the past year over one hundred PaaS companies have entered the marketplace with new services. Groups of directly competing services are beginning to emerge. For example, the PaaS application development market is now becoming very competitive, with services differentiating over features such as vendor lock-in, and the support for legacy systems and code.

Of course, the suitability of particular service offerings for individual Government needs goes beyond the usefulness of the functionality described in the previous sections and will also depend ultimately on the program requirements of an organization. Services will also be considered for characteristics such as security, reliability, and operational performance. Terremark's CEO Bruce Hart suggests, "Federal IT leaders cannot afford to ignore the underlying physical architecture from which Cloud offerings are launched and just hope for the best. They must assure at least the level of availability, security and performance that they realize from traditional hardware-based IT architectures—ideally, they should be able to interconnect those traditional systems to the new Cloud services that they acquire. This creates leverage from all of the benefits of Cloud infrastructure—on-demand capacity and massively scalable elastic architecture, which can bring a new level of flexibility and agility to IT leaders, and with it a compelling economic model that eliminates lumpy capital expenditure and precisely aligns IT infrastructure spend and capacity with the real-time needs of the organization—but it does not sacrifice the power and reliability of controlled, standards-based systems."[80] In this context, the federal IT architect must look at the service both in terms of its functional usefulness, meeting needs in the overall architecture, and its ability to satisfy a range of engineering, statutory, and policy requirements.

Appendix A—Acronyms

Acronym	Definition
API	Application Programmer Interface
BLOB	Binary Large Object
BPM	Business Process Modeling
cXML	commerce eXtensible Markup Language
CEO	Chief Executive Officer
CTO	Chief Technology Officer
COOP	Continuity of Operations
CRM	Customer Relationship Management
DoD	Department of Defense
DAAS	Database as a Service
EC2	Elastic Compute Cloud
EDI	Electronic Data Interchange
GAE	Google App Engine
GB	Gigabyte
GEX	Global Exchange Service
GFS	Google File System
GQL	Google Query Language
HDFS	Hadoop Distributed File System
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IaaS	Infrastructure As A Service
IIS	Internet Information Services
IL	Intermediate Language
IT	Information Technology
JSP	Java Server Pages
JSR	Java Specification Request
LAMP	Linux (operating system), Apache HTTP Server, MySQL (database software), and PHP
LOB	Line Of Business
M&A	Mergers and Acquisitions
MIME	Multipurpose Internet Mail Extension
NIST	National Institute of Standards and Technology
OLAP	Online Analytical Processing
PaaS	Platform as a Service
PKI	Public Key Infrastructure
RDBMS	Relational Database Management System
SaaS	Software as a Service
SAS	Statement on Auditing Standards
SOA	Service-Oriented Architecture
SOQL	Salesforce.com Object Query Languag
SQL	Structured Query Language
SQS	Simple Queue Service
S3	Simple Storage Service
S3N	S3 Native File System
VAN	Value Added Network
VM	Virtual Machine
WSDL	Web Service Definition Language
XHTML	Extensible Hypertext Markup Language
XML	eXtensible Markup Language

Appendix B—Products and Services Mentioned

Product/Service	Website
Amazon SimpleDB	http://aws.amazon.com/simpledb/
Amazon Elastic Compute Cloud (EC2)	http://aws.amazon.com/ec2/
Amazon Simple Queue Service (SQS)	http://aws.amazon.com/sqs/
Amazon Elastic MapReduce	http://aws.amazon.com/elasticmapreduce/
Amazon Relational Database Service (RDS)	http://aws.amazon.com/rds/
Amazon Simple Storage Service (S3)	http://aws.amazon.com/s3/
Appian Anywhere	http://www.appian.com/bpm-saas.jsp
Apprenda	http://apprenda.com/saasgrid/features/billing-and-metering/
BungeeConnect	http://www.bungeeconnect.com/
Caspio	http://www.caspio.com/
EngineYard	http://www.engineyard.com/
Etelos	http://www.etelos.com/
Evapt	http://www.evapt.com/
Force.com (from SalesForce.com)	http://www.salesforce.com/platform/
GoGrid	http://www.gogrid.com/
Google App Engine	http://code.google.com/appengine/
Google Apps BigTable	http://static.googleusercontent.com/external_content/untrusted_dlcp/labs.google.com/en/us/papers/bigtable-osdi06.pdf
Hubspan	http://www.hubspan.com/
IBM DB2	http://www-01.ibm.com/software/data/db2/
Iceberg	http://www.geticeberg.com/
LongJump	http://www.longjump.com/
Magic Software	http://www.magicsoftware.com/en/
Microsoft Azure	http://www.microsoft.com/windowsazure/
Microsoft Internet Information Services (IIS)	http://www.microsoft.com/windowsserver2008/en/us/internet-information-services.aspx
Microsoft SQL Server	http://www.microsoft.com/windowsazure/
Microsoft SQL Azure	http://www.microsoft.com/windowsazure/sqlazure/
Microsoft Visual Studio	http://msdn.microsoft.com/en-us/vstudio/default.aspx
MySQL	http://www.mysql.com
NetSuite	http://www.netsuite.com/
OnlineMQ	http://onlinemq.com/
Oracle Database	http://www.oracle.com/us/products/database/enterprise-edition/index.html
Pervasive's Integration Hub	http://www.pervasiveintegration.com/SiteCollectionImages/Solutions/Cloud_Integration.png
PostgreSQL	http://www.postgresql.org/
RackSpace.com	http://www.rackspace.com
Salesforce.com	http://www.salesforce.com
SOASTA	http://www.soasta.com/
Softlayer.com	http://www.softlayer.com/services/software/
StormMQ	http://stormmq.com/
Sybase Database Management	http://www.sybase.com/
Terremark	http://www.terremark.com/
Vertica	http://www.vertica.com/
Virtual Logistics' Hosted EDI Services	http://www.virtuallogistics.ca/services/hosted-edi-services.aspx
Xeround	http://www.xeround.com/
Zuora	http://www.zuora.com/

References

[1] Rymer, John R. , "Platform-As-A-Service Is Here; Can It Help You?",
http://www.forrester.com/rb/Research/platform-as-a-service_is_here%3B_can_it_help_you/q/id/47335/t/2

[2] Mitchell, Dave, "Defining Platform As A Service",
http://www.infoq.com/news/2008/03/as-a-service-gains-acceptance

[3] Engine Yard, "Engine Yard AppCloud",
http://www.engineyard.com

[4] Netsuite, "NS-BOS: the NetSuite Application Development Platform",
http://www.netsuite.com/portal/developers/ns-bos.shtml

[5] Etelos, "Etelos SaaS Application Platform",
http://www3.etelos.com/etelos/platform/esap-intro.html

[6] Force.com, "Apex Code Developer's Guide",
http://www.salesforce.com/us/developer/docs/apexcode/salesforce_apex_language_reference.pdf

[7] Tony Stubblebine, "An Introduction to Salesforce.com's AppExchange",
http://www.oreillynet.com/pub/a/network/2006/11/13/an-introduction-to-saleforcecoms-appexchange.html

[8] Salesforce, "Salesforce.com Object Query Language (SOQL)",
http://www.salesforce.com/us/developer/docs/api/index_Left.htm#StartTopic=Content/sforce_api_calls_soql.htm

[9] LongJump, "Longjump, Cloud Applications Platform",
http://www.longjump.com/

[10] BungeeConnect, "Integration and Interactivity",
http://www.bungeeconnect.com/

[11] IS Tools, "Solutions",
http://www.istools.com/solutions/

[12] Magic Software, "uniPaas",
http://www.magicsoftware.com/en/products/?catID=70

[13] Caspio, "Create Your Online Database Fast",
http://www.caspio.com/

[14] LongJump, "Category: Building Applications",
http://lj.platformatyourservice.com/~platfor1/wiki/index.php?title=Category:Building_Applications

[15] Gardner, Dana, "LongJump eyes fully portable, best-of-breed PaaS holy grail for ISVs going to the 'open' cloud",
http://blogs.zdnet.com/Gardner/?p=2962

[16] LongJump, "Developer Suite",
http://lj.platformatyourservice.com/~platfor1/wiki/index.php?title=Developer_Suite

[17] MartinButlerResearch.com, "LongJump PaaS",
http://www.martinbutlerresearch.com/index.php/2010/longjump-paas/

[18] Google, "What Is Google App Engine",
http://code.google.com/appengine/docs/whatisgoogleappengine.html

[19] Byteonic.com, "High Level Overview of Google App Engine For Java",
http://www.byteonic.com/wp-content/uploads/2009/04/gae-1.jpg

[20] Google, "What Is Google App Engine",
http://code.google.com/appengine/docs/whatisgoogleappengine.html

[21] Microsoft, "Windows Azure Platform",
http://www.microsoft.com/windowsazure/

[22] ZDnet, "Microsoft's Azure cloud is officially open for business",
http://blogs.zdnet.com/microsoft/?p=5085

[23] David Chappell and Associates, "Introducing Windows Azure",
http://view.atdmt.com/action/mrtyou_FY10AzureWhitepaperIntroWindowsAzureSec_1

[24] Microsoft, "Microsoft SQL Azure Team Blog",
http://blogs.msdn.com/ssds/archive/2009/08/18/9874133.aspx

[25] David Chappell and Associates, "Introducing Windows Azure",
http://view.atdmt.com/action/mrtyou_FY10AzureWhitepaperIntroWindowsAzureSec_1

[26] Ibid

[27] Hogan, Mike, "Database Virtualization and the Cloud",
http://www.scaledb.com/pdfs/Cloud_Databases_WhitePaper2.pdf

[28] Rackspace, "Managed Hosting Solutions",
http://www.rackspace.com/managed_hosting/index.php

[29] Terremark, "Terremark Worldwide Inc.",
http://www.terremark.com

[30] GoGrid, "Cloud Hosting, Hybrid Hosting, Cloud Servers",
http://www.gogrid.com

[31] Information in this subsection is drawn in part from the online manuals provided by Amazon.com,
http://aws.amazon.com/documentation/simpledb/

[32] Amazon.com, "Amazon SimpleDB",
http://aws.amazon.com/simpledb/

[33] Ibid

[34] Google, "Datastore Java API Overview",
http://code.google.com/appengine/docs/java/datastore/overview.html

[35] TechCrunch.com, "Google To Launch BigTable As Web Service",
http://techcrunch.com/2008/04/04/source-google-to-launch-bigtable-as-web-service

[36] Fay Chang et al., "Bigtable: A Distributed Storage System for Structured Data",
labs.google.com/papers/bigtable-osdi06.pdf

[37] ITManagement.com, "What Is Google's Bigtable System?",
http://www.itmanagement.com/features/google-bigtable-073108/

[38] Microsoft.com, "Overview of Microsoft SQL Azure Database",
http://go.microsoft.com/?linkid=9686976

[39] Amazon.com, "Amazon Relational Database Service (Amazon RDS) (beta)",
http://aws.amazon.com/rds/

[40] Xeround.com, "A SQL Cloud Database",
http://www.xeround.com/

[41] Microsoft.com, "SQL Azure Overview",
http://msdn.microsoft.com/en-us/library/ee336241.aspx

[42] Microsoft.com, "Data Types (SQL Azure Database)",
http://msdn.microsoft.com/en-us/library/ee336233.aspx

[43] Linthicum, David, "Are We Moving to 'Platform as a Service?'",
http://intelligent-enterprise.informationweek.com/blog/archives/2007/07/are_we_moving_t.html

[44] Appian, "Appian Resources",
http://www.appian.com/bpm-resources/downloads.jsp

[45] Kemsley, Sandy, “Modeling Processes in a Browser with Appian”,
http://www.ebizq.net/blogs/column2/2006/11/modeling_proces.php

[46] Iceberg, “Iceberg – Software Has Evolved”,
http://www.geticeberg.com

[47] Appian, “SaaS BPM: Silencing the Skeptics”,
http://www.appian.com/bpm-resources/downloads.jsp

[48] Appian, “Business Process Trends”,
http://www.bptrends.com/publicationfiles/07-09-Appian%20BPMSuite%20Ver.%205.7%20_3_-Malcolm1.pdf

[49] Amazon, “Amazon Simple Queue Service”,
http://aws.amazon.com/sqs/

[50] OnlineMQ, “OnlineMQ - Cloud Message Queueing”,
http://onlinemq.com/

[51] StormMQ, “A Message Queue Manager Service”,
http://stormmq.com/

[52] Ibid

[53] Amazon, “Auto-scaling Amazon EC2 with Amazon SQS”,
http://developer.amazonwebservices.com/connect/entry.jspa?externalID=1464&categoryID=102

[54] Ibid

[55] Hubspan, “Hubspan Global Integration on Demand”,
http://www.hubspan.com/solutions/webspanservices/

[56] Pervasive, “Cloud Integration”,
http://www.pervasiveintegration.com/SiteCollectionImages/Solutions/Cloud_Integration.png

[57] Virtual Logistics, “Hosted EDI Services”,
http://www.virtuallogistics.ca/services/hosted-edi-services.aspx

[58] ThinkStrategies, “Hubspan Addresses Growing Cross-Entity Integration Challenges in the On-Demand World”,
http://www.hubspan.com/wp-content/uploads/2009/08/hubspan-on-demand-integration.pdf

[59] Hubspan, “Hubspan White Paper: eCommerce Integration”,
http://www.hubspan.com/wp-content/uploads/2009/09/Hubspan-eCommerceIntegrationWP-final-Web.pdf

[60] Ibid

[61] BTA.mil, “Global Exchange Services”,
http://www.bta.mil/gex/index.html

[62] Amazon, “Amazon Elastic MapReduce”,
http://aws.amazon.com/elasticmapreduce/

[63] Softlayer, “Software Services”,
http://www.softlayer.com/services/software/

[64] Amazon, “Amazon Elastic MapReduce”,
http://aws.amazon.com/elasticmapreduce/

[65] Linthicum, David “How and why the military should adopt the cloud”,
http://www.infoworld.com/d/cloud-computing/how-and-why-the-military-should-adopt-the-cloud-484?page=0,1

[66] Amazon, “Amazon Elastic MapReduce Developer Guide”,
http://docs.amazonwebservices.com/ElasticMapReduce/latest/DeveloperGuide/

[67] Katsnelson, Leon, “Cloud database market is heating up with Amazon announcement”,
http://freedb2.com/2009/04/02/cloud-database-market-is-heating-up-with-amazon-announcement/

[68] Ibid

[69] Soasta, "CloudTest by SOASTA",
http://www.soasta.com/

[70] Soasta, "Cloud Corner",
http://www.soasta.com/blog/

[71] Soasta, "Cloud Test On Demand",
http://www.soasta.com/product/cloudtestondemand.pdf

[72] Amazon, "AWS Case Study: SOASTA",
http://aws.amazon.com/solutions/case-studies/soasta-intuit/

[73] Zuora, "Welcome to the Z-Commerce API Documentation Wiki",
http://apidocs.developer.zuora.com/index.php/Main_Page

[74] Venturebeat.com, "Zuora says it has a bold plan to save journalism—subscriptions",
http://social.venturebeat.com/2009/09/22/demo-zuora-says-it-has-a-bold-plan-to-save-journalism-subscriptions/

[75] Apprenda, "Billing and Metering",
http://apprenda.com/saasgrid/features/billing-and-metering/

[76] eVapt, "Subscription Management Cloud Billing",
http://www.evapt.com/

[77] Beagle Research Group, LLC., "Cloud-based billing and payments takes its place as a strategic process", September 2009

[78] Zuora, "Z-Commerce API Documentation",
http://apidocs.developer.zuora.com/index.php/Call_Basics

[79] Zuora, "Create Subscription Products, Packages, and Rate Plans",
http://www.zuora.com/flash/Zuora_ScreenShot_Flash/pics/Screenshot_BillingOperations_b.jpg

[80] Hart, Bruce,
www.mitre.org/cloudcomputing

MITRE

www.mitre.org