CONSTRUCTING AND EVALUATING SUPPLY-CHAIN SYSTEMS IN A CLOUD-CONNECTED ENTERPRISE

Donald F. Ferguson
CA Technologies, NY, U.S.A.

Ethan Hadar
CA Technologies, Herzelia, Israel

Keywords: Supply-Chain Enterprise Systems, Cloud Computing, IT Services, Business-IT alignment, Enterprise Service Architecture.

Abstract: An enterprise that exploits its IT-services from the cloud, and optionally provides some of the services to its customers via the cloud, is defined by us as cloud-connected enterprise (CCE). Consumption from the cloud and provisioning to the cloud of IT services defines an IT supply-chain environment. Considering the conceptual similar offerings from different vendors is economical attractive, as specialization in services increases the quality and cost-effectiveness of the service. The overall value of a service is composed of characteristics that may be summarized as QARCC: Quality, Agility, Risk, Capability and Cost. Tradeoffs between implementing services internally and consuming services externally may depend on these characteristics and their sub-characteristics. Regardless of the origin of the services or sub-services, we propose that the construction or consumption of the solution should follow dedicated cloud-oriented lifecycle for managing such services. The proposed incremental and iterative process, fosters an agile approach of refactoring and optimization. It is based on the assumptions that services change their QARCC characteristics over time due to emerging opportunities for replacement of sub-components. It is designed to operate in internal clouds as well as external and hybrid ones.

1 INTRODUCTION

IT departments currently deliver their IT services (Wikipedia, 2010 a) to employees, customers and partners, and manage these deliverables in a business like fashion. Examples include online banking, web commerce, finance and accounting, order tracking, etc. IT systems may be considered as the backbone of the business, controlling manufacturing, supply chains, logistics, customer relationship management, and more. In some cases, IT **is the realization** of an enterprise's business model in which the IT services are the principle implementation of business services. For example, financial services that include transaction billing of phone calls in telecommunication companies.

IT implementations of business services, or any services for that matter, are composites of hardware, software, applications, data, network, etc. A common definition for the end-to-end solutions implementing business services is *composite application* (Wikipedia, 2010 b), where the functionally needed for the service is driven from several different sources, regardless of ownership of these sources. The definition refers mainly to software elements as the main sub-modules; however, in the domain of IT management, composite application includes hardware and other non-software (SW) resources. More specifically, when examining the end-goal of an IT service to cater for business services, the delivered application is the one that is perceived as the service in question. Thus, the resources comprising such a service (e.g. network elements and servers) are a means to delivering the solution. These decomposition levels of IT concerns focus the decision-making on the managed solutions rather on technical elements constituting it.

Figure 1: Composite Application of an on-premise IT systems, analogous to an in-house supply

F. Ferguson D. and Hadar E. (2010).
CONSTRUCTING AND EVALUATING SUPPLY-CHAIN SYSTEMS IN A CLOUD-CONNECTED ENTERPRISE.
In *Proceedings of the 5th International Conference on Software and Data Technologies*, pages 69-76
DOI: 10.5220/0003048000690076

chain. In such a case, the IT resources implementing a business service are internal within the enterprise. The model is logically constructed as a manufacturing company that is vertically integrated, has only an internal supply chain and performs all manufacturing internally. In an on-premise composite application, the enterprise's IT department purchases the raw material (e.g. servers, routers, database SW, packaged applications, namely, the composite application structure elements), then installs, configures, integrates and manages the base resources to form a completely internal solution.

2 MOVING TOWARDS A COMPOSITE APPLICATION IN THE CLOUD

When off-premise or any cloud solution is considered, modifications to the properties of the elements structuring the "online ordering" system described in Figure 2 can be ownership, location, generic usage permits and elasticity.

Ownership of the full structure will define the billing relationships between the IT department and its supply chain such as leasing or renting the assets comprising the structure (Danielson and Hadar 2009). **The cost model** of the offered composite application between the IT department and the internal consumers from the different enterprise business units can be calculated per usage, periodic, or fixed, paid and budgeted in advanced or charged back.

The **location** of the composite application sub-elements, namely internally within the enterprise sites or externally on a third party hosting service, defines the integrability of the elements and the type of networks that binds them: local network, secured wide area networks, or open networks (such as the internet).

Usage limitations of such the composite application elements offered from a remote location and over the internet can be offered to restricted groups as a private offering, or without grouping and limitations to the public consumers.

Consequently, the Composite Application, offered as a service, can be constructed from only **public Cloud** elements, external to the enterprise, or from elements that are offered by **private cloud** vendors, still external to the enterprise. The offered service can even be constructed as a **hybrid cloud** solution, where part of the service is constructed from on-premise, internal assets, and the other parts from external private or public ones.

Thus, composite application **deployment models** define the external types categorized as Public, Private, and Hybrid. These types are derived from the ownership, location and restricted usage of the underlying elements. Cost models and levels of dynamic flexibility (elasticity) of the offerings are affecting the **business models** between the p**roviders** of the cloud services (external or internal), and the **consumers** of these cloud services (either the IT department or a Business Unit of the enterprise). Intermediation and bridging services may enhance the relationship between Providers and Consumers, facilitated by a **broker**, which can be a third business entity, or be part of the Provider or Consumer organizations.

For the sake of clarity, in this document, any discussion termed with "Cloud" refers to Public, Private, or Hybrid Clouds, which are external to the organization. Solutions that address smart datacenter capabilities, where the assets are fully located within the enterprise sites (whether owned or leased) will be referred to as "Internal cloud", or as Physical-Virtualized (PV) solution.

As detailed ahead in prototypical scenarios, it is anticipated by analysts (Fellows and Piraino 2009) that most of the cloud related solutions in 2010 will be on internal clouds for mission-critical or data sensitive Composite Applications. Non-critical services will be consumed from external Public and Private clouds (such as receiving CRM services from SalesForce.com, or e-mail services from Google), and only a small fraction will be constructed in a hybrid manner. Thus, enterprises are incrementally moving to a model in which cloud service providers implement sub-elements of an enterprise's IT services, either having a transaction crossing between the services, or as decoupled standalone external solution. The transformation is similar to enterprises that have optimized their businesses through supply chain management (Wikipedia, 2010 c) and enterprise resource planning (Wikipedia, 2010 d), delegating services to other businesses and gradually consuming such services.

3 THE CLOUD-CONNECTED ENTERPRISE

An enterprise that exploits its IT-services from the cloud, and optionally provides some of the services

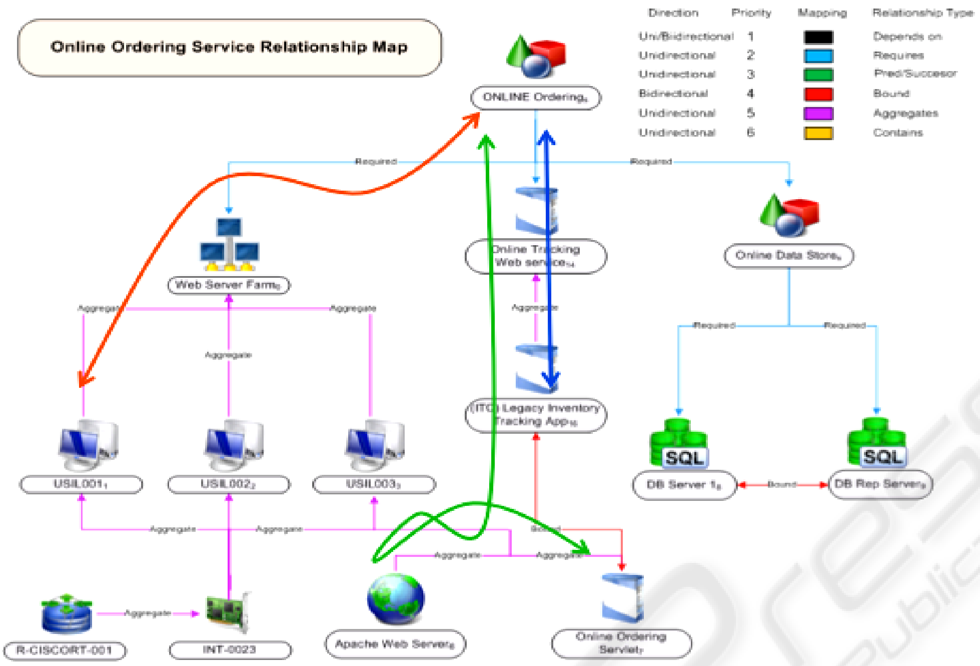

Figure 1: Composite Application.

cloud-connected-enterprise **(CCE)**. Consumption to its customers via the cloud, is defined by us as from the cloud and provisioning to the cloud of IT services define an IT supply-chain environment, as detailed in Figure 2: Cloud Connected Enterprise. The growing offerings from different vendors to the same conceptual offerings, whether Infrastructure or Software as a Service (IaaS, SaaS), generates an economical attraction. Specialization in these services presumably increases the quality of the service, and the need to supply a service as fast as possible supports its agility. Risk is associated with sizing or performing a task which is not the enterprise's main business, and accordingly, all the associated costs are also drivers for selecting a cloud provider instead of constructing commoditized solutions on-premise.

Such benefits can be summarized as QARCC: *Quality, Agility, Risk, Capability* and *Cost.* An enterprise evolving to a value-chain of cloud IT services could achieve tremendous business benefits focusing on the QARCC metrics only.

As a simple example, we can observe the Akamai – Logitech case in which Logitech planned a web marketing campaign that would increase Web traffic to their site by 100 times per day. Logitech used Akamai's content distribution network (CDN) and their Edge Computing technology to implement the solutin.

The QARCC benefits with regard to a solution of implementing the business needs in Logitech internal datacenter where:

- *Quality:* Akamai's technology is far less likely to have transient errors than a quickly deployed, in-house system. Customers would perceive transient errors to be poor quality in the Logitech website.
- *Agility*: Logitech did not need to purchase, install and configure hardware and software. This shortened the time required to deliver the solution.
- *Risk:* A best-of-breed provider's system may be less likely to crash or become overloaded. Logitech's use of the Akamai CDN for a Web promotional campaign handled an unexpectedly large amount of web access without degraded performance.
- *Capability:* The Akamai CDN provides geographic content caching and distribution, and end-to-end performance monitoring over the Internet.

These capabilities would be impossible for Logitech's centralized datacenters.

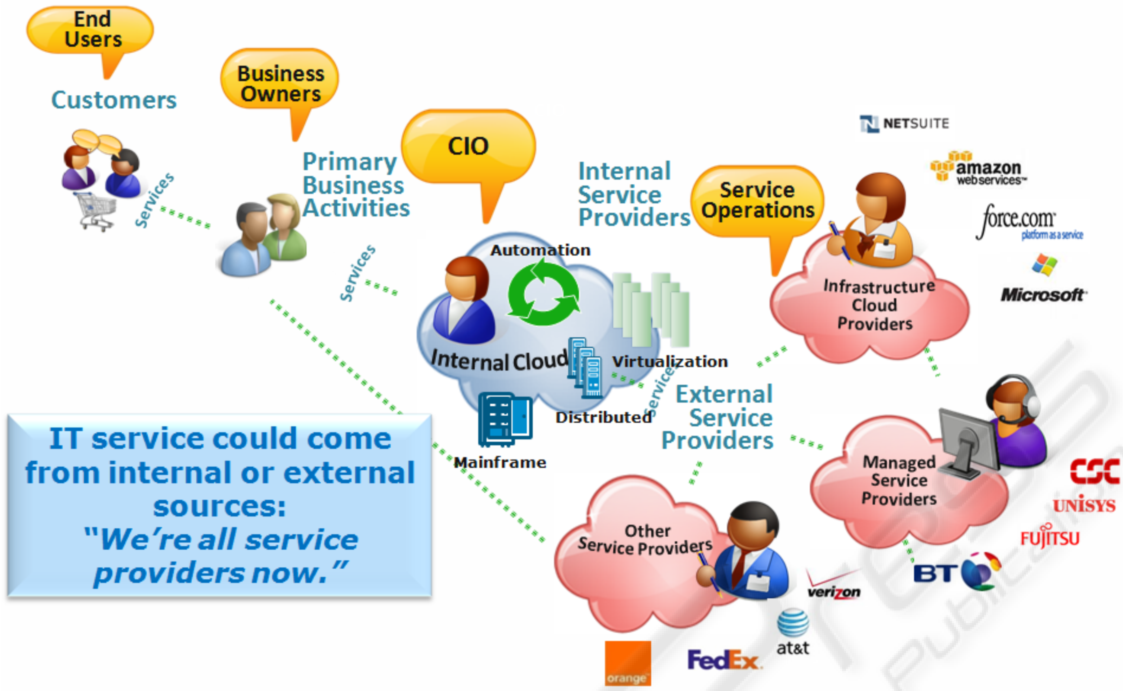

Figure 2: Cloud Connected Enterprise.

- *Cost:* Purchasing, installing and configuring the IT resources would have been prohibitively expensive for a short duration campaign. Logitech simply paid for capacity according to the amount of time the solution was deployed.

Nevertheless, one should be careful when analyzing this example. Becoming a cloud—connected enterprise that gains value by transforming from an IT factory to an IT service value network (SVN) may have negative consequences. Some obvious examples include:

- *Information Security* - An enterprise invoking external business service no longer has direct control over data, which hinders information governance, risk, compliance, and access control.
- *Service Levels Assurance and Agreements* - Evaluating the *quality* of the business service is more difficult. In an on-premise solution, it is possible to debug the end-to-end solution. Consider the example of invoking an external source of demographic information or a business intelligence system. The calling consumer cannot see the input data, calculation formulas, etc.
- *Control* - An enterprise can modify internal hardware and software, and test and analyze configurations. This allows the enterprise to compute confidence in an internal composite application's availability and performance. The enterprise calling an external service has no visibility into how the service provider implements the functions, and cannot assess the reliability of a service level agreement.

Understandably, the next generation enterprises will need to address these concerns of a cloud-connected enterprise. By using QARCC metrics for management decision, on-going improvement of service selection and utilization can increase. By providing tools to control these IT concerns and automating the change, overall assurance of service level agreements is met.

4 THE COMPOSITE APPLICATION AND BUSINESS SERVICES LIFECYCLE (CABS-LC)

Existing IT management processes usually address IT management responsibilities, or recommended steps and best practices for managing IT assets, such as ITIL (IT Infrastructure Library), or other common workflows. Such workflows can be categorized into Operation, Configuration, and Application perspectives:

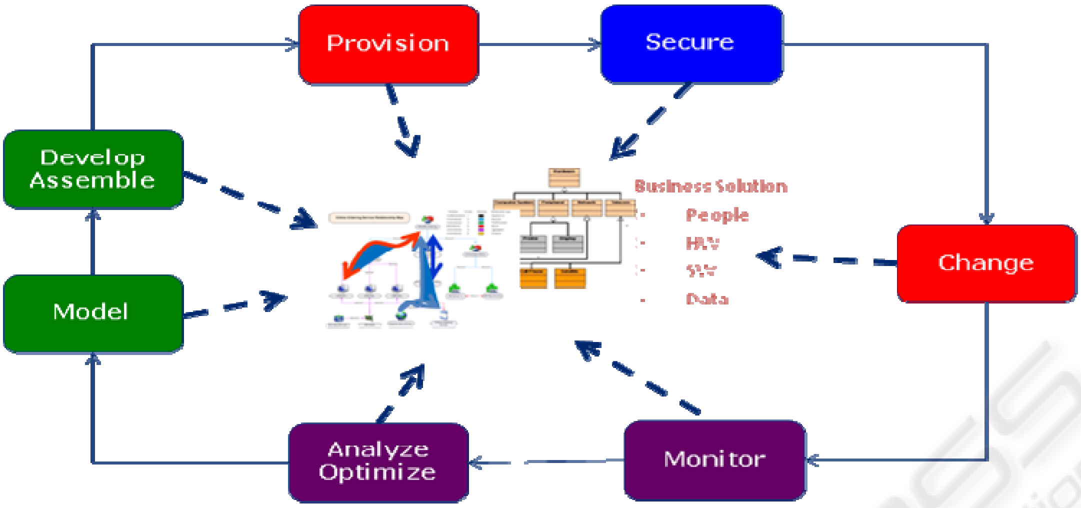

Figure 3: The Composite Application and Business Services Life Cycle (CABS-LC).

Operation perspective: When considering the case of "keeping the lights on", we first start with monitoring the existing state, and accordingly, analyze and plan for execution of a change, coupled with knowledge. Namely, Monitor – Analyze – Plan – Execute with Knowledge (MAPE-K).

Configuration perspective: When dealing with a new service or solution, one needs first to define the rational, and configure an existing solution (such as servers and hardware) followed by maintaining its life duration. Namely, Define – Configure – Deploy – Operate – Monitor – Improve.

Application perspective: When considering providing an application solution, one needs to consider the integration points, and accordingly, construct a solution to be optimized gradually. Namely, Model – Assemble/Develop – Run – Monitor/Optimize.

Consolidating these aspects into a cloud environment, as well as considering that the main approach in services is the construction of a Composite Application and the consumption of a remote service forms a new perspective. This new perspective is The Composite Application and Business Services Lifecycle (CABS-LC), shown in Figure 4.

The ***CABS-LC*** is an incremental and iterative cycle, fostering an agile approach of constant refactoring and optimization. CABS-LC has a dynamic and agile approach based on the assumptions that services change their quality over time and due to opportunities and alternatives that emerge for replacement of sub-components, and business needs change and mutate. CABS-LC is designed to operate in internal clouds as well as external or hybrid ones, distinguished by levels of *Information Security*, *Service Levels Assurance and Agreements*, and *Control*.

The major steps in CABS-LC are Model - Develop/Assemble – Provision –Secure – Change – Monitor – Analyze /Optimize, all acting on a common integration and toolset framework.

CABS-LC deals not just with infrastructure settings, but addresses Composite Applications (and business services) concerns as well. Thus, the lifecycle steps take a new dimension of software development as well as IT configuration of infrastructure, raising the discussion to a holistic business solution one, rather then IT integration conversation. Accordingly, the steps are:

Model – the goal of the modelling phase is to capture the structure of the sub-elements and sub-entities that are part of a Composite Applications or a Business Service, including what metrics are available and needed in order to evaluate the overall QARCC metrics. The models define two main layers: (1) the structure of the Composite Application (components and dependencies), and (2) the QARCC characteristics structure and needed metrics, superimposing on the Composite Application structure. Several models of the same Composite Application, (constructed differently) can be then evaluated and tuned according to the same QARCC metrics. These structures may be defined differently, providing a foundation for evaluation, and eventually, automation of the selected structure assembly and provisioning.

Develop/Assemble – Assembling a solution and configuring its inner components including hardware elements (such as images, VMs, routers, etc) and software services (such as remote web service API calls, configuring Web-service gateways, setting up cross-enterprises identities, etc). This phase may include new designs or adjustments to existing solutions.

Provision (deploy) – Since in a public cloud the notion of provisioning a service actually defines connecting to one provided by a third part, the need is for configuration of agreements. This configuration means setting up a contract with the service provider, and providing the data necessary for the provider to perform the service. Regular provisioning still applies when internal services and applications, as well as infrastructure that the enterprise owns or has access to, are distributed and deployed on regular or virtualized environments.

Secure – Security concerns include identity management that bridges domains as well as applying security interception according to policies. Example are limitations on the deployment locations of servers (known as physical boundary limitations), as well as network segmentations that enable multiple different Composite Applications to co-exist on the same environment. Such a configuration of network firewall and relevant identities depends not just on the theoretical assembly constructed on the modelling environment, but rather self-adapts to a selected location (internal, external, or hybrid), protecting applications and data.

Change – Automation of the change process is comprised of two categories: self-adaptive, and workflow-driven. Workflow-driven implies that change is defined and actuated firstly based on needs and pre-configured triggers, to later on be reused by automation tools. Self-adaptation is a change that usually changes the system's inner state in order to optimize a goal, such as optimize a QARCC defined thresholds. Self-Adaptation changes can be comprised of automated workflow-driven changes, if massive similar changes are applicable. An example for self-adaptation is to structure a system with five servers for load balancing, and a self-adaptation rule that can increase the number of servers up to ten according to load, and decrease the number of provisioned servers to three in case the system is not in use more than one day. The change automated provisioning workflow that can be defined once and reused many times according to the self-adaptation rule.

Monitor – Metrics, thresholds and eventually key performance indicators (KPIs), highlight the status of the monitored Composite Applications by aggregating and accumulating sources of information into manageable knowledge. The knowledge can be provided by automation tools and humans, driven by machine or human perception, and eventually can be transformed into actions. Notice that monitoring includes measuring business processes and real-world outcomes, not just automated data provided by managed machines. An example may be providing an event that informs on the quality of the ground-mail service: "Did the package arrive on time?", in contrast to "What is the average response rate of the package-delivery company Web service API?" Thus, daily operational information can be more than low-level infrastructure driven one by becoming an element of managing the application and business solution. For example, "Do not create a new LDAP ID" or "Add a user to SAP" indicate usage patterns that might trigger a need for improvement that will affect the IT service itself and not part of the underlying system.

Consequently, monitoring is set with composite KPI at the business level, such as "95% of my purchase orders should be completed in 2 business days with no human intervention"

Analyze /Optimize – The result of carefully configured monitors, can serve for analysis and reasoning on the root-cause of a problem, optionally leading to a resolution. An example may be that the system correlates "unresolved requests", with "server transactions load", and analyzes it to "My web application server is aborting requests under load". The triggered resolution of this analysis will be to provision additional capacity in order to reduce the load that will increase the system quality.

Analysis can serve for alternative design between similar services (global optimization), or finding where to improve in a given service (local optimization). Optimization can be done on a single attribute (such as capacity), achieved by automated changes, or done on multiple attributes that will trigger a massive refactoring and replacement of a service.

Changes are part of the dynamic nature of an elastic cloud environment that can expand and contract on need, regulation, or business requirements. Accordingly, using analysis and optimization for evolving and adjusting existing solutions is an inherent part of this dynamic world.

As an example, consider the case of an enterprise running a marketing campaign. In such a case the IT department will need to rent more warehouse space, purchase more pre-order parts for production, change online pricing rules engines, modify the web personalization rules and content database, acquire and provision capacity on IaaS providers such as EC2 and Microsoft Azure, set up end-to-end monitoring with Gomez, etc.

This "marketing campaign" IT project, will be defined as a set of Composite Applications that are connected together, from storage and computing space, to servers, database content changes, recovery, success monitoring tools, business transaction management and more. Accordingly, one needs to:

1. Model the components dependency and define components should be provisioned, including architecture considerations such as availability, scalability, recovery and backup plans.
2. Decide and select implementation components either by buying, pay per use, or building, and accordingly plan how to test the solution in the appropriate testing environment.
3. After the test is successful, actual deployment into an open-network segmentation will commence, requiring additional configurations of the composite application, specifically with access and security permissions. The provisioned solution needs to manage privileges for both administrators as well as regular costumers.
4. Change will happen when the collected marketing broadcast data is updated. Scalability becomes an issue when non-planned users are logging in, and of course, when problems and malfunctions occur.
5. Monitoring the marketing campaign will mean measuring transactions' performance, user experience, and availability as promised by the service providers, and other metrics that could indicate the quality of the Composite Application. These metrics are coupled with additional indications on the actual cost and capacity planned, as well as the ability of the composite application to handle load stress. Namely, the QARCC (Quality, Agility, Risk, Cost and Capacity) metrics. Examples are "how many costumers leave the campaign without any purchase?", "what products are selling the most?" and "is the load balancing system working?", "are there fulfilment problems and bottlenecks in the financial transactions?", "are their hacking attacks to the marketing data?".
6. Analysis of the QARCC metrics of the campaign, as well as its individual sub-elements, required smart threshold setting (KPI) as well as considering alternative services. Analyzing options may identify a need to optimize the solution. Such an optimization may be a decision to increase the service agreement with the provider for more capacity or reduce the risk of information lost by providing additional backup cycle. If the amounts of transactions consumers are activating change, performance might be affected in one of the sub-elements. Triggering and connecting to another sub-service provider to enable the same service may be an option. Consider the case of the sub-service credit-card billing that was not designed to manage such transactional load. This can be achieved by analysis of the underlying elements that construct the QARCC metrics, and finding the root-cause for degradation in performance.

5 DISCUSSION

This paper described the supply-chain concerns of the cloud-connected-enterprise (CCE), affecting the selection of the structure of a composite application that is based on its QARCC quality characteristics. The Composite Application and Business Services (CABS) lifecycle methodology and process supports the selection of the best structure. Consequently, the methodology is driven by a set of loosely-coupled tools that can automate business and architecture rules and assist in optimizing IT management of cloud concerns.

The content and decision on which the automation is acted upon is either driven by community, vendor preferences, or customers' specific needs. It is reasonable to assume that the QARCC needs and CABS automation will differ from one enterprise to another, however, similarities in the QARCC evaluation structure and criteria will emerge.

Our approach for loosely-coupled solutions supporting CABS and QARCC, is propelled by smart integrations of tools that provide capabilities and data needed to compute and assess the structures of the QARCC metrics and values, thus, drive the decision-making of the optimization element of the suggested change, and the following provisioning. This work-in-progress is materialized by the manifestation of such tools, and evidence based results that will be provided as the research progresses.

REFERENCES

Wikipedia, May 2010 a, Business Service Management. http://en.wikipedia.org/wiki/Business_service_management

IBM web site, May 2010, A component-based approach to strategic change, http://www-935.ibm.com/services/uk/igs/html/cbm-intro.html

Wikipedia, May 2010 b, Composite Application, http://en.wikipedia.org/wiki/Composite_application

Wikipedia, May 2010 c, Supply Chain Management, http://en.wikipedia.org/wiki/Supply_chain_management

Wikipedia, May 2010 d, Enterprise Resource Planning, http://en.wikipedia.org/wiki/Enterprise_resource_planning

Danielson D., Hadar E., “Analysis of IT concern landscape as trigger for innovation in cloud computing” , CA’s white paper, May 2009, NY, USA

William Fellows, Antonio Piraino, “2010 preview – Cloud computing”, 14 Dec 2009, 451 group.