CHAPTER III

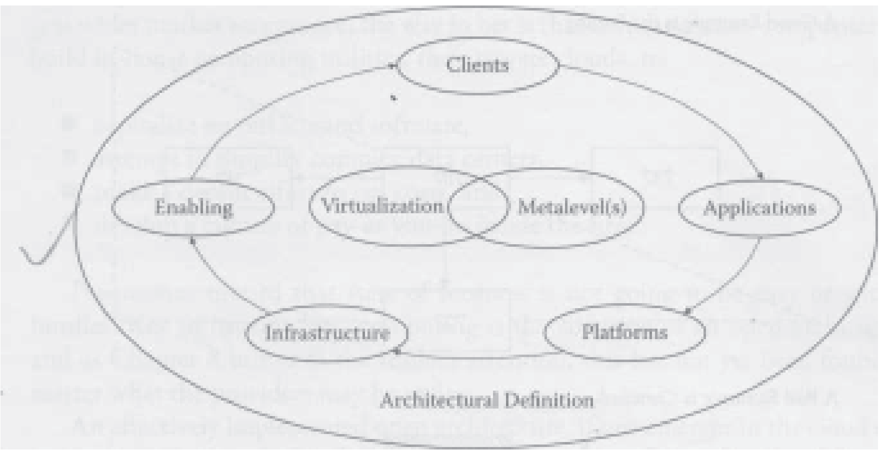
WHAT CLOUD COMPUTING HAS TO OFFER

PUBLIC CLOUDS, PRIVATE CLOUDS, AND CLIENTS

Similar to the policy fol-lowed with extranets and intranets. Proprietary clouds will have many of the public clouds' characteristics:

- being Internet based and dynamically provisioned;
- benefiting from onDemand software, platforms, and enabling services; and
- billing their internal users on fine-grained computing and databasing resources being employed, as providers of external clouds do.

From an architectural viewpoint, this will create *hybrid cloud* environments consisting of multiple internal clouds that, at least in theory, are seamlessly con-nected to external clouds and their providers. As the concept of cloud computing.



the term *clients* stands for a growing variety and type of engines on the end user side? The three main types are

- workstations and PCs with hard disk, hence local storage capability at giga-byte level;
- the so-called thin clients without hard disk, promoted by some cloud providers who offer to map the end users' files into their infrastructural services; and
- a diversity of mobile clients, some with hard disk and others bare, whose accesses, queries, and transactions take place within the cloud environment.
- Thin or thick, the client appears as a single point of access for all the comput-ing needs, and
- cloud computing services are accessible anywhere by every device that has access to the Internet.

Whenever real time is not a competitive advantage or operational requirement, the better option is *asynchronous* operations, which permit minimizing costs and (among other benefits) avoid deadly embraces in database updates and access.

Cloud Applications and Platforms

Vendors active in cloud computing have announced a broad set of applications and tools accessible from anywhere in the world by using an Internet connection. Applications for rental are mainly mainstream routines like customer relationship management (CRM), enterprise resource planning (ERP), accounting, and human resources—the stuff appealing to the broadest possible market. Platforms help user organizations develop their own programming routines.

Originally known as *software as a service* (SaaS) or *software on demand* (SoD)* this inventory of applications emulated onPremises packages focusing on sales, bookkeeping, finance, planning, scheduling, productivity, and communications.

An important point to bring to attention is that today a significant part of onDemand applications are rented—not sold. Tomorrow this practice may well be generalized. That's very good for the user organization because it means that it cannot manipulate the routines to make them fit its internal procedures, a com-mon policy with bought onPremises commercial packages.

- Consumers and companies can access onDemand software fairly easily.
- The likelihood of contextually linked ads makes sense if it is associated with free-of-cost apps.

Small and medium enterprises (SMEs) that are a target market (s^e-Ol rapier 4) can purchase more advanced versions of these applications with no accompanying ads, for a fee. Typically onDemand routines are browser and operating system (OS) agnostic, which, however, should not be interpreted to mean that user organiza-tions cannot be locked in to a cloud computing vendor's suite.

Still, as long as onDemand software is operating system and browser agnostic, its online availability constitutes an interesting development in IT, which is sure to have long-term consequences. Standards have never been the forte of information technol-ogy companies, and without that agnostic feature there would have been an explosion of cloud computing versions—one or more by vendor—and with it the risk that:

- interfaces will be increasingly incompatible, and
- programming modules of similar functionality but different standards would be causing a lot of pain to both software developers and users.

Old hands in the computer industry remember that this has happened not just between vendors but also within vendor establishments and its wares. Programming products have been classically developed by different teams in competition with one another and without appropriate coordination in regard to protocols, operating sys-tems, database management systems (DBMSs), and teleprocessing routine (TPR).

Market niches are usually exploited by start-ups. The view of big vendors like IBM and Microsoft is that not all applications are suitable to be handled through cloud computing. For instance, those with high memory requirements are best when installed locally, temporarily uploaded onto a centralized service (emulating the concept of a cache) for cross-departmental collaboration. This is essentially a hybrid model: OnDemand plus OnPremises.

The alternative is *platforms*, the pillar of cloud computing next to applications software (~~Chapter~~ Platforms are largely intended to encourage the new generation of developers to add to the programming products made available on demand by the vendor. Indeed, cloud applications and platforms complement one another, which explains why the latter have attracted the interest of companies, in spite of the fact that large user organizations have already made investments in platforms from incumbents like Microsoft, IBM, and SAP.

The cloud platform model of software development pro vides interesting economics, being in reality an offshoot of the provider's underlying infrastructure, including its database management system and operating system. User organizations can save money by capitalizing on the facilities platforms offer.

By all evidence cloud platforms will be particularly favored in the case of new application development reflecting the user's willingness to embrace a solution that allows for rapid and cheaper deployment of applications—particularly if the plat-form acquired is user-friendly. Large incumbents are aware of the risk this strategy carries to their base of revenue and develop strategies for responding to it.

For instance, Microsoft's Azure Platform has two layers: The base is proprietary hardware and software serving as a central point in managing switches, routers, server nodes, and connectivity. The upper layer is composed of building blocks like Live services, NET services, SQL services, SharePoint, and Dynamics CRM. All of these are designed for usage in conjunction with applications developed on the base layer, and can help in extending onPremises applications to cloud computing.

Providing the Cloud Infrastructure

The discussion on onDemand software and platforms has given a glimpse on the *infrastructure* that hosts them. The cloud vendors sell time-shared facilities to their clients who consume computing, databasing, routing, and switching resources *as a service*. This way:

- users pay a rental for only the facilities they employ, and
- the pay-as-you-do bill is calculated on the basis of a utility model that works in a way analogous to that of traditional utilities

Pay-as-you-do means invoicing as infrastructural services are consumed. The alternative is to pay on a subscription basis. Vendors of infrastructural services are still on the learning curve on how to reach a dual goal: maximize their profits and still be competitive in pricing.

The underlying assumption is that sharing computing, databasing, and net-working resources improves their utilization rates. Computing engines and data-base servers are not left idle. At least theoretically, this reduces costs as resource usage rises—though not necessarily dramatically because the constraint is conten-tion. Many *providers* claim that this way customers do not have to worry about peak load limits, but this is half true at best.

One of the benefits to be derived by this strategy is better forecasting and planning by vendors of infrastructural services and their clients. Another benefit is the demolition of the simplistic belief that cloud computing services, which are upscale on the food chain, can be equated to electricity, whose bare wires and outlets are way down on the food chain^ The cloud has prerequisites. Two of the most evident are

- high-speed bandwidth for the Internet, making it feasible to receive very short response times from vendor-centered infrastructures and

- steady watch that this infrastructure is upgraded by the cloud computer vendor, to ensure enough capacity, avoid bottlenecks, support high reliability and provide first-rate security—all at a relatively low cost.

As these references demonstrate, and as should be expected, there exists a huge gap between theory and practice. It is alright to say that with cloud computing user organizations may avoid capital expenditures on hardware, basic software, and associated services and instead pay the provider only for the resources they use, but it is wrong to think that all these goodies will fall from heaven as the need arises, like biblical manna.

Prospective users must appreciate that there exist many challenges associated with very large-scale storage. As hard disk capacity has crossed the multipetabyte environment demanded by a cloud infrastructure, data management and availabil-ity, response time, deadly embraces, system failures, blackouts, backups, and plenty of other issues haunt cloud vendors. It matters little that consumption is billed on a utility or subscription basis often with minor or no up-front cost. What is important is to:

- know thedownside connected to these expected benefits and
- gain enough knowledge about the cloud, well before being committed to "this" or "that" vendor, based on promises about "a broad range of IT services.

present cloud computing infrastruc-ture falls short of the outlined requirements and their projected evolution. It gen-erally consists of a more or less scalable infrastructure that indeed allows fairly sophisticated searches (and has cost providers a good deal of money) but has not passed a *stress test.*

For instance, Google says that it has spent an estimated $8 billion building thirty-seven data centers worldwide that house more than a million intercon-nected low-end servers. These are very large systems of unprecedented proportions, addressed to a population of widely variable user profiles. They might be well man-aged, but they have not yet come under stress.

Even the question of the best-fit database management system is not yet settled. Google uses a proprietary distributed file management solution. Other infrastruc-tural vendors bet on relational solutions.

Microsoft's *SQL Services* provides a relational database approach for working with structured, semistructured, and unstructured information elements. It is built on the company's synonymous database technology extending SQL Server capa-bilities to cloud computing. The evidence that this will perform under stress in a cloud environment is not on hand:

- Microsoft's SQL Server was originally developed for relatively contained cli-ent-server solutions.
- By contrast, with the cloud we are talking of requirements related to mega-systems, and as experience teaches, big systems are not small systems that grew up.

Available evidence suggests that the design requirements posed by a cloud com-puting infrastructure still have to be properly settled. These have much to do with *macroengineering* (huge dams, coast-to-coast electricity networks, space travel) and very little to do with portals . Since 1998, Netscape's Netcenter, among other examples, Internet portals have made online interactive information and transactions so much easier, enabling full-service websites that are

- online launching pads,
- entertainment networks, and

- shopping malls rolled into one.

But what cloud computing vendors set out to achieve has no resemblance to daytime-nighttime web-based entertainment or consumer chores. It is requiring extensive expertise beyond that of Internet traffic, with routing utilizing propri-etary algorithms in combination with techniques like caching, compression, route optimization, and load balancing (to automatically direct traffic through the best path). The cloud will severely penalize infrastructure providers who:

- fall behind in resource planning,
- for one reason or another start having dissatisfied customers,
- cannot run faster than their competitors all of the time, and
- employ old systems software in a futile attempt to bridge their cloud product line with their old one (for instance, the case of IBM and Microsoft).

Survival in the cut-throat, dynamic environment of a megasystem like the cloud infrastructure is in no way guaranteed by calling this or that systems approach dynamic. Just renaming the crust of the cake will be like betting on an empty shell, and it will constitute a good prescription for losing contact with the market.

Cloud Computing, Spectral Efficiency, Sensors, and Perspiration

The best applications in cloud computing will not be According to several opinions, me best applications in cloud computing will not be those coming from the conversion of old IT but from a domain still in its infancy: smart phones, personal assistants, mini-laptops—in short, the new generation of personal computers to be connected at almost anytime to anywhere. An infrastruc-tural solution worth its salt must account for this option.

This requirement essentially means that it is wrong to look at cloud computing's infrastructure just as a vast array of interconnected machines managing the data and software that run on all sorts of IT equipment. The better way is to perceive it i s an expanding combination of mobile and fixed technologies that:

- already pose a long list of challenging but fairly stochastic technical require-ments and
- will shape up ongoing advances in the computing universe in the next decade.

Sensors embedded in machinery currently in use can gather data in direct ways about the reliability of component parts as well as temperature, humidity, noise level, wear and tear, and more. Adding digital sensors and remote controls to the transmission and distribution of a power network, for example, and combining their input with data mining and expert systems, will turn it into a *smart grid*whose management is far more responsible and transparent than today's approaches, mak-ing it better able to:

- cope with new sources of renewable power,
- feature coordinated use of different sources of energy,
- provide information to consumers about their usage,t and
- permit utilities to monitor and control their networks more effectively.

In a fairly similar way, sensors and digital relays installed on other types of transmission and distribution will make possible a higher level of supervisory con-trol by optimizing the employment of resources. They will also provide a new array of services that find precedence in the introduction of supply chain management a decade and a half ago.

Project Management Metalayer
Design-to-Order Consultancy
Help-Desk Typically Internet Based

Three layers of enabling services for cloud computing.

The after effect of this real-time input can be so much greater if combined with Six Sigma. Known as reality mining and employing virtual doubles, this combination of facilities can x-ray entire organizations. Eventually the benefits from cloud computing will come from such business x-rays, but they will not flow automatically. Reengineering the organization is the key to opening the door to the benefits. As Thomas Edison had it, the key ingredients of success are 5 percent inspiration and 95 percent perspiration.

The Technology of Enabling Services

The technology of the cloud's enabling services can be divided into three strata, as the most common and least sophisticated is the help desk, currently offered by several cloud providers at the lower end (on a subscription basis). Salesforce.com is an example. The middle layer addresses itself to consultancy and what could be called design to order. This targets both internal and external cloud computing, with its deliverables still at an early stage of a fundamental definition.

While a user organization may have such skills available from its other business lines and previous IT projects, it may still require third-party input specific to cloud computing ; for instance, in connection to:

- Reengineering studies that are a prerequisite to a successful application and
- The distinction that must be made, in the most careful manner, between the clod provider's system architecture and the user organization's business architecture.

Starting with the lower layer of enabling services, help desk activities are quite common these days (though in many cases they are not of satisfactory quality). As such, they don't need explaining in any detail. The reader's attention should, however, be brought to the fact that the quality of help desk services must be contractually guaranteed in the service level agreement (SLA).

Vague promises are not acceptable, no matter who makes them, particularly so as the experience form help desk for e-mail and other Internet services is not the best. It is no secret that people manning help desks are not properly trained through simulators, and they often lack experience. That's bad enough with e-mail, but it can be disaster with cloud computing.

Consultancy services are, in general, of higher quality. As far as computer vendors are concerned (and evidently cloud outfits), services offered under the umbrella title enabling is their spearhead into the cloud market. Their deliverable can best be described as design to order, whose solutions should be tailored to the user organizations; requirements but use commodity on Demand software.

It is essentially this duality that may call for expertise and support to see the project off the ground. But not all skill should be external. User organizations will need plenty of internal skills to be in charge of the quality and cost-effectiveness of what vendors are selling them and to ensure the latter don't engage in what is generally known as vaporware.

In use since the 1990s, the term vaporware describes the practice of public announcement of software products well before they are ready for sales (let alone for installation at client premises). The purpose of this more or less generalized practice is to cause user organizations not to buy a competitor's product that is either currently available or ready to enter the market.

Vaporware practices find their origin in the early twentieth century, in the infamous sales policy of the "broken down cash register" practice by NCR. (At the time National Cash Register was a start-up eager to take market share away from established cash register companies, like Standard.) That practice consisted of first selling a machine under Standard's name, which was a fake, and then , after the merchant had enough with it because it was breaking down all the time, replacing it with an NCR cash register "at no extra cost to the customer."

Enabling services will invariably focus on reengineering the user's business architecture (the master plan must be made by the client), component optimization, custom-made adcf-ons to cloud applications, definition of scaling requirements, studies on growth trajectory, and (most importantly) engineering analysis for power and cooling.

User organizations will be well advised to be on guard for false claims. For instance, promises made by vendors to their clients that cloud computing is "eco-logical" are part of vaporware. As we will see later, computers and com-munications are huge consumers of energy, and even an idle computer that is "on" uses roughly 65 percent of the power of a busy computer.

In principle, general purpose servers consume more power and generate more heat as they are configured to run at maximum level for a specific workload, not the optimal one. Vendors say that by minimizing unnecessary features that create excess heat, and by appropriately provisioning power and cooling components, they will bring energy consumption under control. That, too, is vaporware.

Saving energy is necessary, but it is also more easily said than done, particularly since cost reasons see to it that clouds deploy commercial off-the-shelf gear that is far from being energy savings. If capital investments were no problem, this might have allowed us to take advantage of the latest technology, but cost optimization is not a recipe for minimizing energy usage. A balanced design would trade energy for cost, and vice versa.

Is it possible that a better project management will cut cost, swamp energy con-sumption, and increase quality of deliverables at the same time? Some people say so, and vendors promote the idea, but this statement shows people who either lack project experience or tell an outright lie.

A cloud project, and any other project, is a well-defined activity set up to produce *predetermined results.* As such, it involves the management of resources along four axes of reference:

- know-how, and hence personnel;
- money, and therefore budgets;
- time, projected in timetables; and
- computers and communications power.

As an organizational activity, the project is dedicated to the attainment of a precise, short- to medium-term goal.

If the enabling services to the user organization who gets ready for cloud com-puting apply Monet's principles, *then* they may be worth its salt. *If* project control is asleep at the wheel of governance and there are overruns, *then* the enabling assis-tance and the project itself will be a failure. Let me moreover add that the final responsibility for cloud project management should be with the CEO of the user organization, not with the vendor's consultants.

Cloud and Virtualization

Plenty of enabling studies will be needed to bend the curve of energy consump-tion—if this is at all doable. In reality it is not, because we continue adding more sophisticated services that consume energy, like virtualization. *Virtualization* con-sists of separating software functions from specific hardware devices. The goal is to share hardware capacity between software modules, in a way invisible to:

- end users,
- applications,
- platforms,
- storage systems,
- computing devices, and
- other infrastructural components.

Virtualization is at the heart of cloud computing. Through it resources are being shared while software and hardware facilities provide end users with seamless access to their specific applications and information elements they need for their ark. It is doing so by recasting physical resources into logical facilities.

Each partition created through virtualization (more on this later) acts as an dependent entity that can be assigned an OS and application supported by a given . c ad. However, the act of separating logical from physical resources brings along . - illenges, costs, and energy requirements. On one hand, it is easier to dynamically iid virtual files than to incorporate physical servers as nodes to the system. But on % he other hand, crashes do happen, and recovery becomes more complex.

- Virtualization, which technically abstracts format and programs heterogene-ity from the OS, is producing results but not miracles.
- Operating system heterogeneity limits the number of physical grid nodes available for productive work.
- The management of data centers becomes a demanding enterprise, as more applications are provisioned on a greater number of servers.

What is stated in the above list can be better appreciated by returning to the • andamentals. The *virtual machine* concept has been known since the 1970s. It carted with IBM's VM operating system on mainframes, which permitted a pro-cessor to split itself into several virtual machines, sharing the (high-speed) memory. The twenty-first-century version of virtualization is that of:

- employing networked computers to provide a real-time infrastructure, assured through the *hypervisor*, and
- making use of the hypervisor to enable sharing access to software and hard-ware modules.

In the background of virtual memory developments has been the concept of *waging,* which came to light in the late 1960s.

Further still, apart from ensuring applications mobility within and between clouds, virtualization helps in improving the usage of multicore processors (supposed to save energy). Such multicores are employed to provide more computing power per processor, and their growth has acted as a catalyst for the movement toward virtual-ization. The downside is that most application, database, and system software would have to be rewritten to fully optimize the power of multicore processors.

In addition, there is no reason why virtualization cannot be combined with other technological advances like data source *wrapper agents*, encapsulating various heterogeneous data sources, or broker agents sharing knowledge about, and trans-actions with, data source agents.

The wrapper agents enable plug-and-play third-party software, allowing one agent to communicate on a metalevel with other agents, while exploiting domain- specific software. The broker agents provide information that could answer more effectively user requests. For instance, brokers can accommodate single-occurrence requests, service recurring behavioral objectives, and react dynamically to changes in goals. All this sounds great, but the reader should never forget that much of it is written in a future sense.

Strategic Products and Tactical Product

Because most firms are not technologically self-sufficient, strategic decisions must be made regarding which technologies are core to their business interests and which play a supporting role. With this comes the need to distinguish between strategic and tactical products, focusing on:

- how best to manage sourcing for core technology and
- whether to buy or make tactical software routines.

Starting with the premise that all sorts of products and services today require a good deal of software support to have market appeal and be profitable, the com-puter programs written or bought to support a company's strategic product line should themselves be strategic.

Every case, *strategic products* are longer range as well as key income providers. They are Tew but vital to the company's main business, and they must get most of management's attention. Often, but once again not always, they are the result of in-house R&D characterized by the fact that the company:

- has a commitment to their continuation,
- benefits from a degree of pricing freedom, and
- spends no money for subsidies to keep them on the market.

Typically strategic products have novelty, but not all innovative products become strategic.

Tactical products are shorter range and should be under steady profit and loss (P&L) evaluation as to whether they are continued. Some tactical products result from in-house R&D; others are bought licenses. In either case, they should answer low-cost production and distribution criteria (some may be sold at lower than pro-duction cost to lure clients to the firm and its strategic products, which means subsidized prices). Tactical products:

- are less affected by innovation than strategic products and
- are frequently outsourced to benefit from lower production costs.

These statements are generic in nature. They do not apply only to the cloud or (even less so) only to software. In their substance, they are particularly true in prod-uct lines with high competition.

Semiconductor industry policies, for example, range between those that main- tarn- a large in-house manufacturing technology infrastructure and those that rely almost exclusively on external suppliers.