Gartner

Research

Publication Date: 22 March 2010 **ID Number: G00174718**

Building Private Clouds With Real-Time Infrastructure Architectures

Donna Scott

Private cloud services are attractive to IT organizations because they enable self-service ordering of frequently requested services as well as dynamic provisioning, thus, increasing the agility and reducing the cost of delivering IT services. Real-time infrastructure architectures can be used to dynamically optimize private clouds as well as shared hosting environments.

Key Findings

- While real-time infrastructure (RTI), IT service management and cloud services can operate independently of each other, they are best operated together to provide the highest level of intelligence and optimized runtime services, particularly for private cloud services.
- RTI architectures dynamically optimize runtime services through active intervention, taking analytical feeds from IT service management and using service management to take its actions.
- Cloud attributes like self-service ordering and dynamic provisioning are attractive to IT organizations desiring to offer more-flexible IT services at lower labor costs.
- IT service management adds value to cloud implementations in order to perform functions like dynamic provisioning, as well as performance and availability management.
- Many cloud service implementations are not production services.
- IT service management functionality is typically less rigorous for nonproduction IT services versus production-based RTI architectures.

Recommendations

- Start with development/test lab provisioning, Web services (e.g., self-service requests and dynamic provisioning for Web environments) and database services for private cloud implementations.
- Use the dynamic optimization that comes with element management, for example, in hardware, middleware, databases, hypervisors and operating systems.
- Add IT service management on top of element management capabilities to deliver the end-to-end IT service with RTI architectures.

- Pilot a private cloud implementation to gain support for shared services and to build transparency in IT service costing and chargeback.
- Implement change and configuration management processes and tools prior to implementing private cloud and RTI architectures.

TABLE OF CONTENTS

LIST OF FIGURES

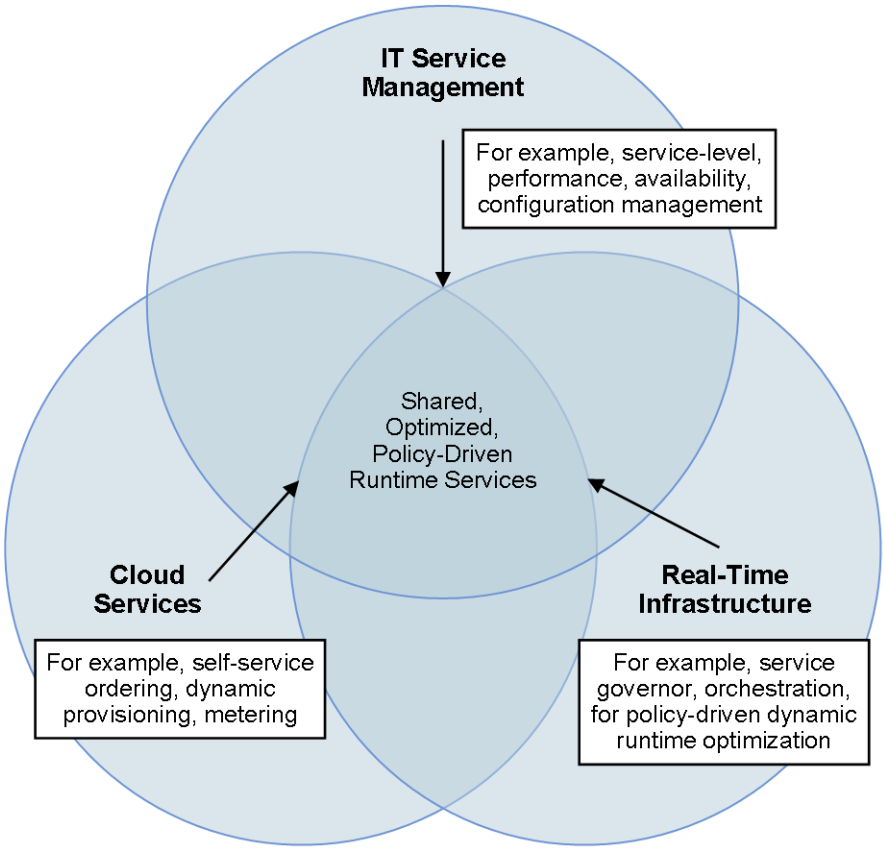
1.0 Introduction

RTI architectures have made progress during the last few years, especially because of the advancement and ubiquity of virtualization technology. However, service management and cloud services have played a role in advancing RTI as well. Service management processes focus on managing end-to-end IT services and component piece parts. As a result, without service management, RTI would not have end-to-end service-level agreements (SLAs) to use to manage and optimize the runtime environment, nor would it have an accurate picture of the health and performance of the services. Cloud services offer attributes such as self-service ordering and dynamic provisioning for order fulfillment — and they offer an example of what the IT department could do internally to deliver services better, faster and at a lower cost.

However, figuring out which standardized services could be offered is another story. This research helps organizations understand which types of use cases should be considered for private cloud services, as well as the interrelationships among cloud, RTI and service management and when they should be implemented together.

Figure 1 shows the intersection of the three areas of IT service management, cloud services and RTI.

Figure 1. Three Different Disciplines Coming Together to Optimize Runtime Services: IT Service Management, Cloud Services and RTI



Source: Gartner (March 2010)

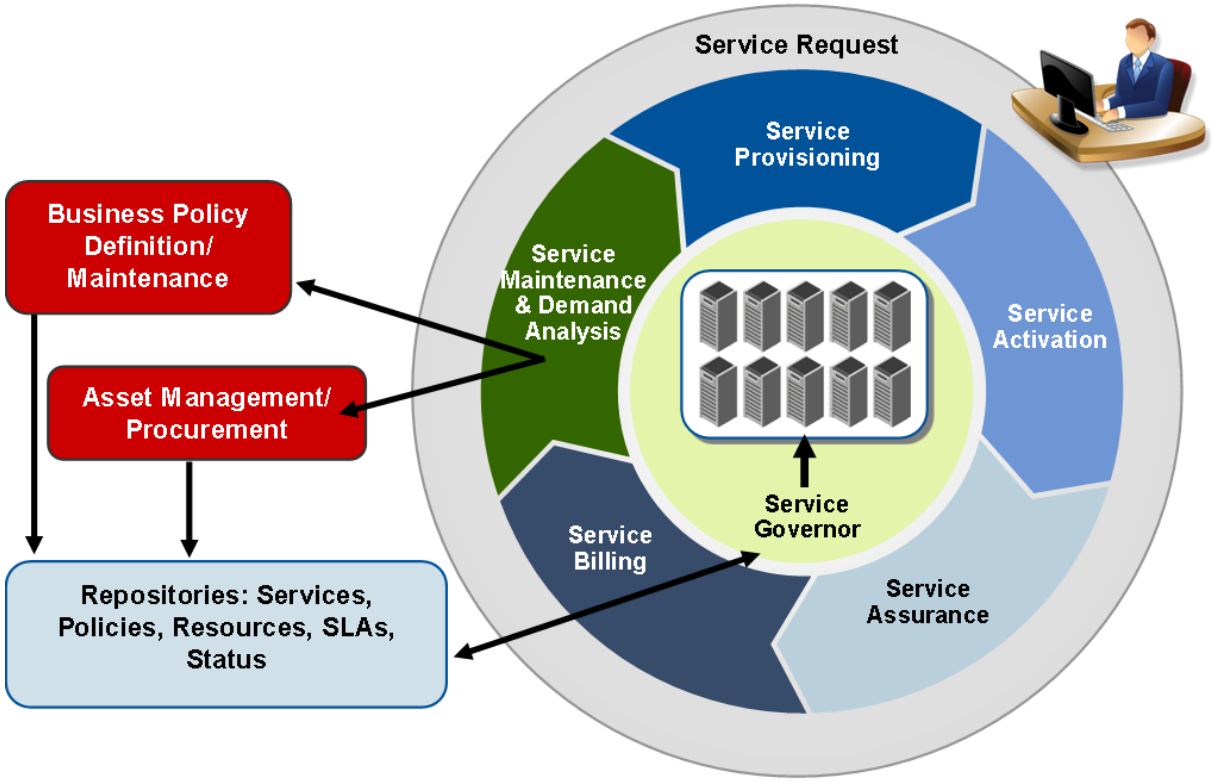
While each are can operate independently, they are best operated together to provide the highest level of intelligence and optimized runtime services:

- **Cloud services** provide service orientation, self-service ordering, dynamic provisioning, elasticity and metering (see "Five Refining Attributes of Public and Private Cloud Computing").
- **IT service management** provides the intelligence layer so that service performance, availability and capacity can be analyzed and acted on proactively.
- **RTI** provides runtime optimization and execution for services and resources. RTI maps the demand for shared services and resources to the supply of shared resources, enabling a more efficient and cost-effective runtime environment, including providing dynamic elasticity of resources to services to meet SLAs. Service management provides the intelligence to optimally manage runtime services.

2.0 IT Service Life Cycle Management and Runtime Optimization

The key parts of the IT service life cycle integrated into an RTI architecture are shown in Figure 2.

Figure 2. RTI, Cloud Services and Service Management Life Cycle Flow



Source: Gartner (March 2010)

IT service management represents the service life cycle phases depicted inside the circle while RTI is depicted in the center of the circle — indicating the runtime environment and real-time actions taken on shared resources to meet SLAs and service demand. RTI is the runtime execution environment with the service governor taking input from service management (i.e., analytics), as well as orchestrating actions on resources and services using service management tools (i.e., configuration management and/or orchestration tools). On the outside of the life cycle sits the service request, a requirement for cloud services. The following describes the phases of the IT service life cycle:

- In the service provisioning phase, the planning, modeling (including setting policies) and service levels are defined.
- Service activation allocates resources, provisions and configures the service, and starts it (by linking with the RTI service governor and underlying resources). When the service starts, usage information is recorded for chargeback purposes.
- Service assurance manages its health and performance, replacing failed components when necessary (again, through the service governor), and fine-tuning/optimizing the environment as necessary to meet the SLAs and priorities. The service governor takes feeds from the repositories and assurance, and takes appropriate optimization actions based on pre-established policies. In taking action, the service governor links to service management automation, typically configuration management tools, to enable configuration changes and elasticity.

- Service maintenance focuses on updating the service, SLAs and priorities (e.g., through incident, change, release and configuration management processes and tools), while demand analysis is focused on the capacity management of the service and the pool of resources — much like material requirements planning (MRP) is in an ERP system. Its methodology is based on the IT service demand forecast versus the aggregate available resource supply, while considering policies for resource provisioning (for example, how much excess resources should be provisioned to be sure that all IT service goals are met).
- Service billing is a requirement for cloud implementations — with metering based on service usage. This is not a requirement for RTI architectures (see Section 3).
- Procurement of IT assets (e.g., infrastructure), such as to increase the capacity, would be handled through an asset management/procurement function.
- A set of repositories is required to maintain information about and the status of business policies, IT services, SLAs, schedules and resources (components). The RTI service governor requires a real-time state to analyze and initiate actions; therefore, the information it acts upon must be as real-time as possible. The repositories can be one unified set or disparate systems with real-time integration, federation and reconciliation. IT service management functionality would include a real-time service and component status (in use, available), so that available resources could be readily provisioned when needed by an IT service.

Without RTI, IT service management focuses on runtime service and resource monitoring, but does not intervene in the runtime to optimize it. With RTI, the two work in conjunction with each other to dynamically optimize the runtime so that service levels are met. Optimization actions will depend on the architecture of the application and platform, but could include scaling up, scaling down, scaling horizontally, moving workloads or services, failing over workloads or services, and performance tuning actions. Furthermore, the service governor will bidirectionally communicate with all the federated or subservice governors that may be in place, for example, at the hypervisor layer (see Section 5).

3.0 Attributes of Cloud and RTI

Figure 3 denotes the differences between cloud (which is a style of computing) and RTI architectures — while they have some common attributes, such as service orientation and shared resources, other attributes differ, for example, run time policies and intervention.

Figure 3. Differences Between RTI and Cloud Attributes

Attributes	Public/Private Cloud	RTI
Service-based	Yes, removes complexities from service consumer, who defines what is meant by service	Yes, but not necessarily end to end; could be component-/ tier-level service
Shared-infrastructure	Yes	Yes
Self-service ordering	Yes for programmatic interface	Not required
Dynamic provisioning	Yes for order fulfillment; but not required for ongoing optimization	Yes and required — both dynamic and scheduled, based on policy and SLAs
Elastic	Yes, but manually initiated is acceptable	Yes and required — both dynamic and scheduled, based on policy and SLAs
Runtime policies and intervention	Not required	Required
Metered	Yes	Not required

Source: Gartner (March 2010)

Cloud services draw from RTI to do dynamic provisioning and enable elasticity. However, cloud services have attributes that are not required for RTI architectures. For example, cloud services require self-service ordering and metering/billing for the services provided. An RTI architecture does not require self-service ordering, because you may just want to optimize the IT services or workloads that are running (or hosted) in your data center. Moreover, while metering is desired, it is not required to have an RTI architecture, but it is required to have a cloud service. You may be focused on optimizing your runtime with RTI and don't have a chargeback model or mechanism in your enterprise. If that is the case, you would not be implementing private cloud services, but you would be implementing RTI.

Note the other nuances: While a cloud service can have dynamic elasticity, it would rely on an RTI service governor to fulfill the capacity increase or decrease. Moreover, a cloud service does not have to be dynamic — it may require manual intervention (which may also be required to get authorization for increased billing). RTI requires dynamic intervention and management of the runtime environment, including dynamic elasticity based on business policies and SLAs.

Cloud services mask the underlying complexities from the service consumer, because the service administrator predetermines which cloud services are self-orderable and writes the automation to enable dynamic provisioning. RTI generally optimizes like groups of shared-runtime resources, applications or IT services. In other words, RTI provides dynamic runtime optimization (including elasticity) for cloud services; however, cloud services may exist in a more static way, with manual intervention required to add resources to services (because budget approval may be required). Moreover, RTI architectures may exist independently of cloud services (to optimize the runtime).

3.1 Public/Private Cloud Versus RTI Membership, Ownership and Control

There are other differences between private cloud and public cloud services and how they relate to RTI architectures. Public cloud services — for example, Amazon Elastic Compute Cloud (Amazon EC2) or Google Apps — have an open membership, and anyone can join or buy the service. A private cloud service has limited or exclusive membership, typically by one enterprise, one business unit or one type of enterprise (for example, a payment-clearing service used by member banks). RTI is orthogonal to membership, because it is the architecture behind scaling and elasticity of the service — thus, it doesn't really care whether the membership is open or exclusive.

RTI also does not care about asset ownership or control, but consumers of cloud services do. A public cloud service has assets owned by the cloud service provider and controlled by the cloud provider. These providers determine the terms of the service, and any consumer of the service can sign up for the standard terms. In a private cloud service, the ownership of assets can be over a spectrum — from internally owned assets to outsourced or public cloud assets. Control of the service is either internal; or, if a service provider is involved, control is defined and negotiated in a contract. An example of control may include data placement, defining who performs what administrative and management tasks, and similar activities. A service governor implementation would enforce such policies across resources that are defined and shared across the spectrum of the public and private cloud.

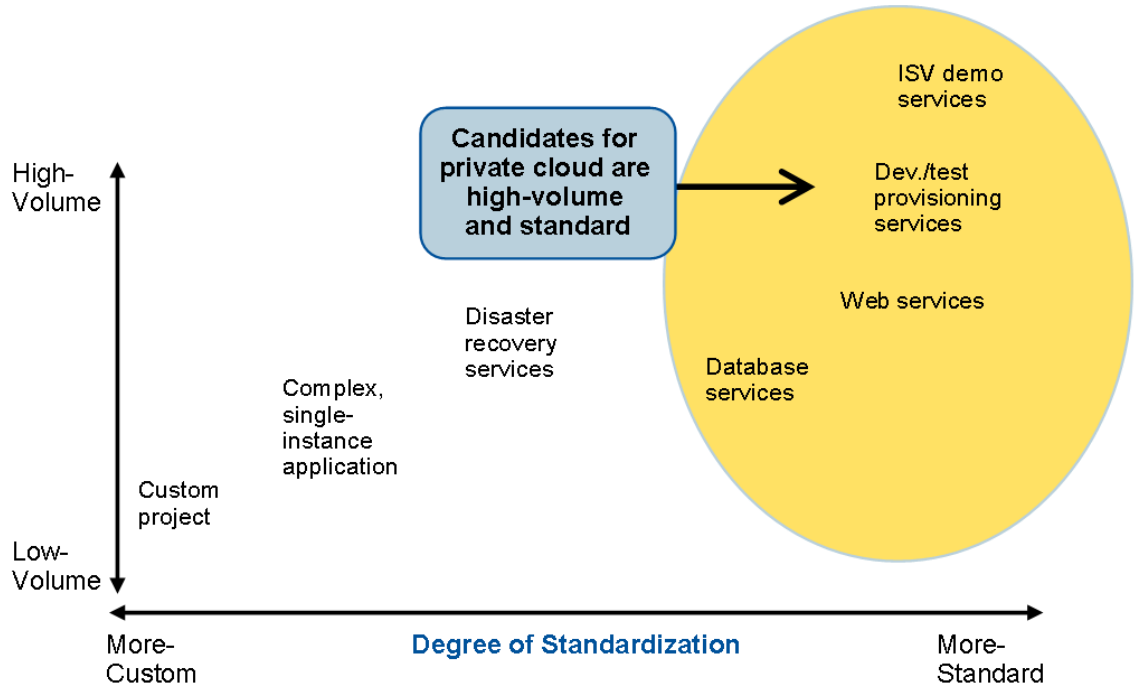
4.0 Use Cases for Private Cloud Services

Enterprises seeking to offer private cloud services should look for standardization opportunities, that is, where they get the same request for services frequently. An obvious example is development and test organizations that need lab management solutions for which they can provision the operating system, middleware and database services very quickly (thus increasing the efficiencies of the developer/tester while taking the labor out of provisioning) from a shared pool of resources and return the resources to the pool when complete. Another example is for independent software vendor (ISV) demo services or for corporate training initiatives. Production services may be appropriate, too, and some enterprises set up private cloud services for applications or services that may be used by many departments or customers in their organizations. These can then share the pool of resources, thus cutting the cost of service delivery.

Private cloud services are not just anything you can order in a service catalog. For example, password reset and moves, adds and changes (MAC) services typically offered by a well-run service desk organization are not considered private cloud services. Nor would adding employee user access to an application (in other words, employee onboarding and access would be part of user provisioning, identity and access management, and service desk/desktop management processes). While private cloud services may not be appropriate for custom projects or single-instance applications, RTI architectures can be implemented to optimize these less-standard environments with others (such as to optimize Java Platform, Enterprise Edition [Java EE] applications on a shared platform).

The level of service management implemented will vary with private clouds, from minimal for nonproduction environments to complete life cycle management for mission-critical production environments (see Figure 4).

Figure 4. Candidates for Private Cloud Services



Source: Gartner (March 2010)

4.1 What About Embedded Element and Virtualization Managers as Service Governors?

Many of the technologies you purchase, such as servers, operating systems, hypervisors, storage, networks and software (for example, middleware, databases and applications) come with embedded element management functionality. The embedded functionality typically enables configurability, monitoring and platform-level functions that may differentiate that platform from another (for example, in dynamic optimization or tuning capabilities). Sometimes these functions are "free" with the platform, and sometimes they involve additional charges. Analysis of the functions, benefits and return on investment (ROI) of these tools should be performed, as they may provide significant value at low cost and with few impediments.

Because these tools often are used by one group of engineers or administrators (such as database administrators [DBAs], server engineers, application administrators and virtual machine administrators) who are responsible for a portion of the infrastructure, they are often easier to implement because they are more easily installed, focus on a specific subset of components and do not require process changes that cross departmental boundaries. Moreover, there may be enough standardization in the portion of the infrastructure controlled (for example, a specific application or a specific platform for a defined set of resources) to get value from the automation without having to standardize across the entire data center.

While element management provides value, enterprises must realize that optimization at the element level is not the same as optimizing at the IT service level, and that it will not deliver on the vision of an automated IT service management life cycle. Multiple element managers will exist through 2014, and the need for integrated automation will be addressed by RTI technologies (with embedded automation and orchestration) and/or run book automation (RBA) and orchestration tools. Moreover, there is a chance that element managers could optimize the element but

suboptimize the end-to-end process, so care should be taken to minimize these risks through the implementation of service-level monitoring and management. Additionally, IT service management, used in concert with RTI architectures, enables integration between the RTI service governor and the subservice governors, for example, element managers or hypervisors. This way, optimization can occur at multiple levels in the hierarchy, but with the confidence that service levels and business priorities take precedence.

5.0 Decision Framework/Bottom Line

When assessing your needs for private cloud and RTI architectures, use these guidelines in your decision-making processes:

- **Sharing resources and implementation of private cloud/RTI:** Evaluate which resources you want to share for which IT services. If the services are repeatable and frequently requested, consider implementing a private cloud. If the services are more custom (i.e., for specific application hosting, which is not repeatable), then implement them on an RTI architecture in order to optimally share the runtime resources among services.
- **IT service management issues:**
  - For production hosting services, implement IT service management to gain intelligence on service performance and availability as input to the RTI service governor to optimize the runtime based on actual data.
  - Configuration automation is a prerequisite to enable dynamic service provisioning for private cloud and RTI architectures. Implement change and configuration management processes and tools prior to implementing private cloud and RTI architectures.
  - Match the service management "rigor" to the requirements (and cost) of the service. For nonproduction private cloud implementations, many service management functions will not be required. For example, resource monitoring may be required (e.g., in order to ensure enough resource availability for the defined services and resource allocations), but service monitoring and performance typically are not required. Moreover, change, release and configuration management usually are performed in a much more relaxed environment, and often with different sets of tools (e.g., they're often controlled by the application development and testing groups). Metering and chargeback may or may not be required, depending on organizational requirements. However, capacity management and forecasting will be required to ensure enough shared resources to meet the demand.
- **Business management issues:**
  - Offering shared infrastructure services requires a change in thinking about the acquisition of infrastructure (in anticipation of demand), delivery of services (automated and fast) and chargeback/pricing (in order to ensure future capacity to meet demand). Pilot a private cloud as a means to gain support for these new business models. This will enable you to offer standard services, self-service ordering and dynamic provisioning (at lower labor costs), which, if done well, will drive the right behavior of customers (to shared environments). We recommend starting with nonproduction services, which require less rigor prior to moving to production services.

- Moving toward RTI requires a culture change and maturity of infrastructure and IT management processes toward service-alignment (see "The Current State of IT Infrastructure and Operations Maturity: Immature!"). Start with a focus on standardization of technologies, software configurations and IT processes, so that as much as possible can be automated. Private cloud — especially in nonproduction environments such as lab management and training — purposes are much easier to implement and can be used as a starting point, without having to integrate with all the service management processes. However, production-level private clouds will require integration with service management processes.

- Set appropriate expectations as you move toward RTI, because most of the technologies are at the Hype Cycle Technology Trigger point or are not yet mature.

RECOMMENDED READING

"Using ITSM to Facilitate the Adoption of External Cloud-Computing Services"

"Toolkit: Planning Performance Management and Capacity Planning Best Practice Implementation"

"The Architecture of a Private Cloud Service"

REGIONAL HEADQUARTERS

Corporate Headquarters
56 Top Gallant Road
Stamford, CT 06902-7700
U.S.A.
+1 203 964 0096

European Headquarters
Tamesis
The Glanty
Egham
Surrey, TW20 9AW
UNITED KINGDOM
+44 1784 431611

Asia/Pacific Headquarters
Gartner Australasia Pty. Ltd.
Level 9, 141 Walker Street
North Sydney
New South Wales 2060
AUSTRALIA
+61 2 9459 4600

Japan Headquarters
Gartner Japan Ltd.
Aobadai Hills, 6F
7-7, Aobadai, 4-chome
Meguro-ku, Tokyo 153-0042
JAPAN
+81 3 3481 3670

Latin America Headquarters
Gartner do Brazil
Av. das Nações Unidas, 12551
9° andar—World Trade Center
04578-903—São Paulo SP
BRAZIL
+55 11 3443 1509