California Department of Technology

California Enterprise Architecture Framework

Cloud Computing (CC) Reference Architecture (RA)

Version 1.0 Final
January 2, 2014

This Page is Intentionally Left Blank

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

This Page is Intentionally Left Blank

1 Introduction

Recent years have witnessed a rapid increase not only in interest, but also in the actual adoption of Cloud Computing, or the "Cloud", as it is commonly called. This interest has been driven primarily by the cloud's potential to reduce costs and time to delivery; also, by a possibility of creating new services and new business models, which could not be realized or realized as easily before the cloud.

Although the capabilities of the cloud may be attractive to any business, they are of special interest in the public sector, because of its budgetary constraints and organizational complexities. Because of the potential positive impact of the cloud on the IT in the organization and on its business benefits, the cloud is a key consideration in any Enterprise Architecture (EA). It is clearly of importance for California Enterprise Architecture Framework, Version 2.0 (CEAF 2.0). This Reference Architecture focuses on Cloud Computing in the context of CEAF 2.0 and is intended to point out facets that are attractive or important for the state and state agencies, especially when considering migration to the cloud as a part of elaborating respective EA Roadmaps.

1.1 Purpose

The Cloud Computing (further abbreviated as "CC") Reference Architecture (further abbreviated as "RA") document provides guidelines and options for making architectural decisions when implementing or selecting CC solutions.

The objectives for the document include the following:

- To provide inputs for creating or evaluating architectures for CC-friendly solutions
- To identify building blocks (architectural layers, services, components) when considering applications or systems for CC-based deployments, including partial deployments
- To communicate the key architectural decisions relevant for creating or evaluating CC-based solutions

1.2 Limitations

The document focuses on CC and related concepts in context of CEAF 2.0 and is not intended as a comprehensive introduction to Cloud Computing in general. This document is not intended as a product guide or endorsement of any cloud-related product. Rather, the intention is to provide distinctions helpful when evaluating the fit for CC in a specific business context, and when considering various CC services and scenarios available.

1.3 Intended Users

The primary intended users of this document are Enterprise Architecture practitioners and other architects that contribute to enterprise architecture. This broad group includes architects from other domains/disciplines such as Security, Application, Information, Business, Technology, Infrastructure, and Solution Architects. It is also beneficial to Managers, at senior or operational levels, who are involved with Cloud Computing or related areas, such as Enterprise Application Integration, Identity and Access Management, SOA, and similar areas.

1.4 Document Organization

The CC Reference Architecture documentation is organized as follows:

- Section "CC Overview" provides background for the CC RA by introducing descriptions and definitions of Cloud Computing, discusses the main service types found in CC implementations, and identifies architectural components
- The section "CC Reference Architecture Description" elaborates RA for CC using the Conceptual View (in the section "CC RA Conceptual View") which introduces the necessary capabilities for a CC architecture and how they are supported by Architectural Building Blocks (ABBs)
- The section "Glossary" provides description of the terms and abbreviations used in the document
- The section "References" lists publications used for preparation of the document

1.5 Future Directions

Future evolution of the document includes the following steps:

- Addition of an example or examples of existing realization of the CC RA in the state
- Identification and elaboration of solution sharing opportunities
- Formulation of implementation for CC RA and/or selection guidelines for CC

2 Cloud Computing Overview

This section provides a description of Cloud Computing, its main concepts and key capabilities of CC solutions. This is followed by a short discussion of the intended business benefits of CC.

The section also includes a description of key components at a high level and a summary of main usage scenarios involving CC.

2.1 Definitions

This Reference Architecture adopts the definition of Cloud Computing provided by the US National Institute of Standards and Technology (NIST). This definition is provided below:

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

The NIST model of Cloud Computing identifies a number of Essential Characteristics, Service Models, and Deployment Models that - taken together - characterize Cloud Computing. The following figure shows NIST-*based* visual model of Cloud Computing:

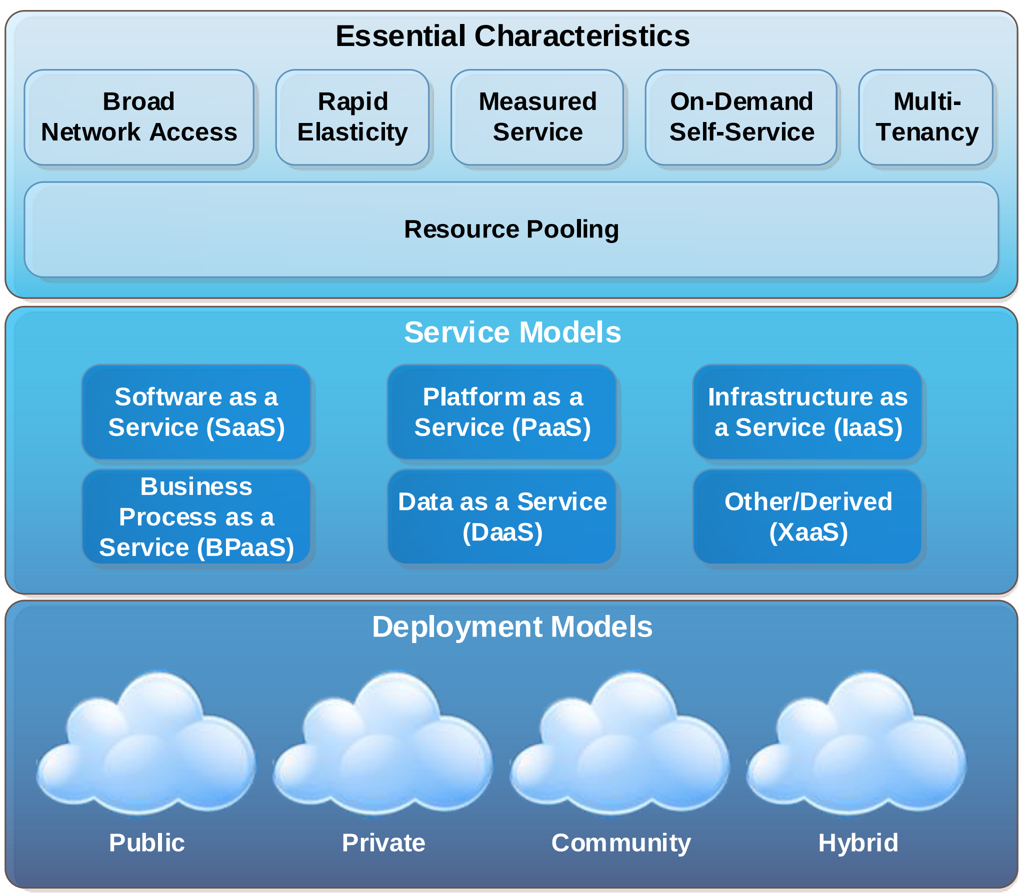

Figure 2-1 NIST-based Visual Model of Cloud Computing

The above figure shows the NIST model for CC with a small number of additions introduced to reflect current trends and to make the model more future-proof. The original NIST model distinguishes the following facets of Cloud Computing:

- Five **Essential Characteristics**: Broad Network Access, Rapid Elasticity, Measured Service, On-Demand Self-Service, and Resource Pooling
- Three **Service Models**: SaaS, PaaS, and IaaS
- Four **Deployment Models**: Public, Private, Community, and Hybrid

In this Reference Architecture, the following elements are also recognized (and shown in the above figure):

- *Multi-Tenancy* as yet another Essential Characteristic (rather than just a side-effect of resource pooling or an aspect of SaaS)
- *Additional Service Models*, including Business Process as a Service (BPaaS), Data as a Service (DaaS)
- *Derived Service Models*: There are a number of emerging *extensions* to the existing service models, such as Desktop-as-a-Service or "DTaaS", Database-as-a-Service or "DBaaS", Security-as-a-Service or "SECaaS", and similar. They are called here "extensions" because they do not introduce fundamentally new service models, but rather are variations or specializations of existing ones (e.g., DBaaS can be viewed as a special case of PaaS)
- *Other (new) Models*: More service models for Cloud Computing may appear and garner attention because of the benefits they could offer, and the label "XaaS" is used as a placeholder for these

The above characteristics, CC service models, and CC deployment models are discussed in dedicated subsections later in the document.

Please note that the terms "Cloud Computing", "cloud", and "CC", are treated as synonyms *in this Reference Architecture document* even though in other contexts they may have distinctly different meanings.

2.2 Characteristics of Cloud Computing

The NIST-based model of Cloud Computing adopted in this Reference Architecture identifies the following *essential* characteristics of Cloud Computing:

- On-demand self-service
- Resource pooling
- Rapid elasticity
- Measured Service
- Broad network access
- Multi-Tenancy

The above characteristics are not independent of one another; rather, some of the characteristics support the others – e.g., broad network access is typically a precondition of usability of the cloud in general, and resource pooling is a precondition for on-demand service, rapid elasticity and support for multi-tenancy. It is important to note that these essential characteristics belong to the cloud together as a whole, and are not open to a pick-and-choose approach.

The characteristics listed above are further discussed in the subsections that follow.

2.2.1 On-Demand Self-Service

According to NIST, on-demand self-service allows consumers "to unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each Service Provider. "

The understanding of on-demand self-service adopted by NIST stresses the unilateral ability of Service Consumers to adjust resource usage to the required level at any time. This is an important feature, which distinguishes CC from e.g., traditional server hosting arrangements.

The above description from NIST can be interpreted as excluding human interaction (rather than just not requiring one). However, when self-service is treated more broadly, in addition to automated provisioning and de-provisioning, it also includes the following facets, regardless of whether they are executed by human users or software (e.g., reacting to some events that are broadcast without human intervention):

- Management of service instances
- Reporting on provisioned services, service usage, etc., typically in near-real time
- Self-service through a web portal for human users. The portal publishes, among others, the catalog of available services (and related APIs), so that:
    - A subscribed consumer may browse the service catalog and provision a cloud resource through portal
    - A subscriber's application may provision new instances as-needed by using the cloud provider's API
- Self-service using standards-based protocols (e.g., https(s), SOAP, etc.) and explicit API for programmatic interaction, which makes it possible to easily automate cloud-related procedures on the cloud Service Consumer side. It is important to note that Service Automation Management is critical to providing self-service capabilities
- Publishing of the service catalog (and related APIs), so that:
    - A subscribed consumer may browse the service catalog and provision a cloud resource through portal
    - A subscriber's application may provision new instances as-needed through API

2.2.2 Broad Network Access

Following NIST, broad network access means that cloud's "capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations)." [D].

It is important to note that Broad Network Access is not limited to broadband network connectivity, although having it available helps make cloud services transparent to the user. Rather, the emphasis is put on heterogeneity of client platforms, and their ubiquity, or at least lack of strong dependency on physical location of Service Consumers. Consequently, Broad Network Access involves the notion that the resources hosted and managed in the cloud are potentially available to any computing device from any Internet connected location.

It is important to point out that in practice, a Service Consumer may want to or even be forced to (e.g., because of legal/regulatory requirements) restrict the availability of cloud-hosted resources to intranet and/or to specific devices in some business scenarios. This requirement is typically addressed in CC by supporting declarative configuration – rather than forcing Service Consumers into creation of dedicated components or mechanisms to manage configuration of the components in the cloud.

Another facet of Broad Network Access is that applications deployed on the cloud may interface to other cloud-based or non-cloud applications. Note that in such scenarios, standardizing the APIs prescribing interfacing to/from cloud-deployed applications is crucial for successful adoption of the solution. Moreover, in those scenarios, using Enterprise Application Integration Bus/Hub (as described in a separate Reference Architecture for EAI) for integration is typically highly desirable and worth considering.

2.2.3 Resource Pooling

According to NIST, Resource Pooling means that "the provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the consumer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing (compute), memory, and network bandwidth."

Resource Pooling is a technique used by Service Providers to support, among others, On-Demand Self-Service, but it is not a feature of CC that is directly visible to Service Consumers in a transparently functioning service. In turn, what makes Resource Pooling feasible is Virtualization, typically in the form of virtualized servers providing for virtualized and scalable resources:

- Virtualized servers are the primary units to be consumed
- Virtual servers constitute a large pool of resources that can be made available when required
- Virtual servers are shared among subscribing consumers but not dedicated to a single consumer exclusively

Usually, pooled resources are owned by the Cloud Service Provider – rather than Service Consumer, in contrast to non-cloud deployments. However, it is also possible for those resources to be owned and/or managed by an organization within the enterprise that is, as a whole, also the consumer of the cloud services. In the public sector, such an option may be attractive in the context of a larger enterprise (such as state), where agencies can use the services provided by a centralized data center to their advantage.

2.2.4 Measured Service

Measured service refers to a metering capability at a level of aggregation that is appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts) in the cloud. Even though the cloud systems can automatically control and optimize resource use by leveraging their internal resources, their usage can be monitored, controlled, and reported. This way, Measured Service contributes to transparency for both the Provider and Consumer of the utilized services.

Measured Service allows service consumers to pay only for what they use. Viewed in a larger perspective, Measured Service has a broader impact, for it affects the basic economics of supporting organization's IT needs: it modifies the traditional ratio between capital expenses and operation expenses related to IT.

2.2.5 Rapid Elasticity

According to NIST, Rapid Elasticity means "capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly out ward and in ward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time."

Dynamic Scaling complements Measured Service and underwrites On-Demand Self-Service. It allows the required loads to be supported by the used service instance, at the time the load is needed, and with the cost of supporting a given load incurred only when it is actually utilized. This automated allocation and freeing of resources decreases the cost of providing Rapid Elasticity and cloud services to the consumer. From the Service Provider perspective, this capability increases reliability of the services without having to over-compensate with a potentially unnecessary investment in the infrastructure. From the Service Provider point of view, ability to support Rapid Elasticity in an economically viable way is connected not only to Resource Pooling, but also to what degree the cloud-deployed resources can make use of Multi-Tenancy.

2.2.6 Multi-Tenancy

Multi-tenancy is a critical technology for Cloud Computing that allows for one instance of application serving multiple customers at the same time, while sharing cloud's resources. This sharing increases operational efficiency and decreases operation costs.

There are a number of options for realizing multi-tenancy, depending on the extent of multi-tenancy sharing available and/or required. The options include the following:

- Shared hardware (such as physical servers , network) and OS
- Shared processing (hardware and shared application platform)
- Shared platform (shared processing and data facilities)
- Shared application instance (shared platform and shared application)
- Shared software code and architecture (shared infrastructure, architecture, and code libraries

The following figure (adapted from Gartner) illustrates these options:

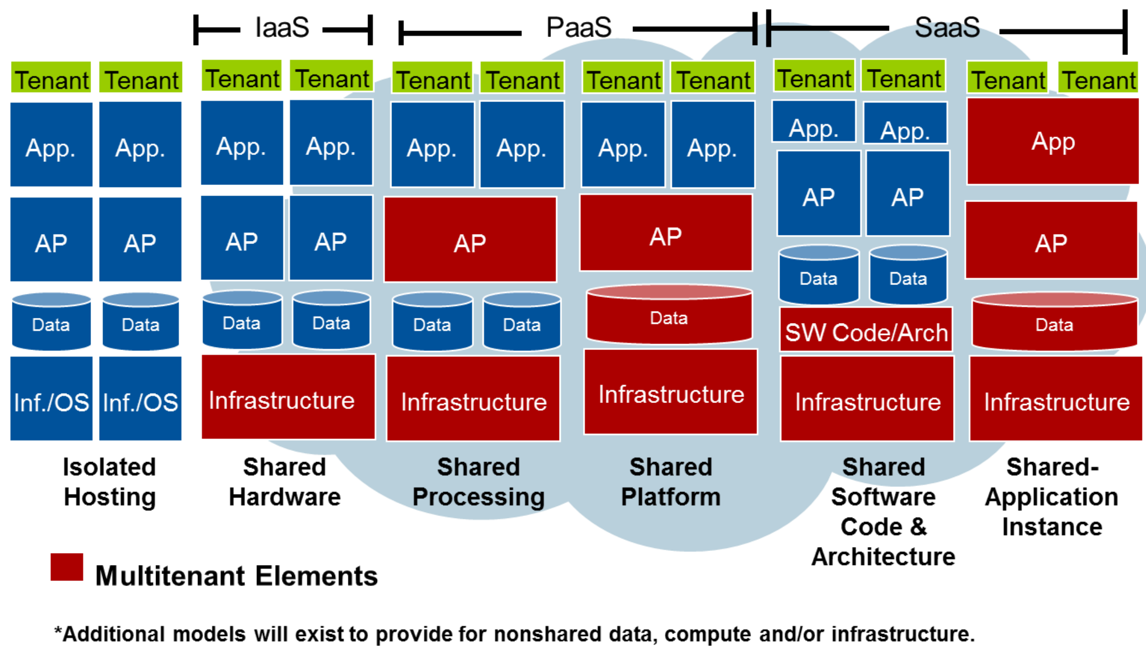

Figure 2-2 Multi-Tenancy (Gartner)

It is worth noting that a given Service Model (IaaS, PaaS, and so on) may have more than one choice with respect to which specific cloud elements are made multi-tenant – for example, in PaaS, there are two distinct choices: one that shares application platform and infrastructure and the other that shares the database platform in addition to application platform and infrastructure, as previously described.

In case of SaaS, the options regarding multi-tenancy become more complex. The options involve design- and implementation- level agreements and conventions that make an application multi-tenant-capable. The primary criterion for a multi-tenant-capable SaaS is that the architecture and the underlying software code of SaaS application do not change from consumer to consumer. The SaaS application may support specific requirements of a consumer through configuration without requiring changes to its architecture and/or its software code. Common choices for SaaS, as shown in the above figure, are *multiple instances* supporting multiple tenants and a *single instance* supporting multiple tenants.

2.3 Business Benefits

The main business benefits of cloud-based solutions can be summarized as follows:

- Increasing the focus on delivering business value by delegating responsibility for running and managing of parts of the infrastructure and technology stack to a specialized third party
- Improving business agility by using services on-demand, integrating services, and by making use of scalable computing
- Operational efficiency, resulting from reduced capital expenditure and on-demand usage of IT resources
- Increased operational reliability and resilience, guaranteed by SLAs and supported by state-of-the-art availability, access control and backup solutions
- Environmental efficiency and sustainability, resulting from decreased energy consumption and reduced IT facility footprint

In the literature on Cloud Computing, the most often quoted benefits of CC are shown in the following figure [3]:

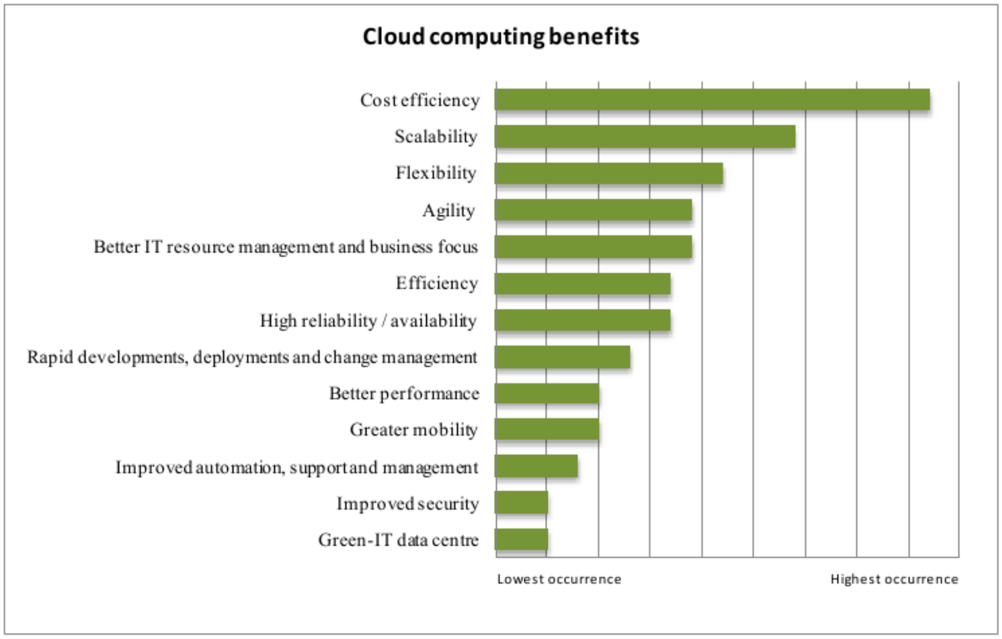

Figure 2-3 Cloud Computing Benefits

The figure can be interpreted as reflecting perceptions of benefits of CC in its users. The most compelling benefit of CC is cost savings, which are achievable in IT as a result of economies of scale. The savings can be realized in the following areas:

- Operational expenditure (OPEX) reduction, resulting from applying measured service ("paying only for what is used"), and from reduction of on-going costs related to infrastructure
- Capital expenditures (CAPEX) reduction, resulting from e.g. server and other hardware acquisition reductions
- Time-to-market reduction

Other gains from successful adoption of CC include the following:

- Shortening time-to-deliver and time-to-change by improvements in application construction, testing and delivery in a standardized environment with readily available resources (such as ad-hoc servers) and services, improvements in automation and support of systems
- Increasing competitiveness by increasing agility and flexibility, reliability of the systems, fast scalability, etc.
- Helping business process focus when infrastructure-related issues get addressed by adoption of the cloud

2.4 Cloud Service Roles

In Cloud Computing, the following roles can be identified:

- Cloud Service **Provider** that is responsible for providing cloud services
- Cloud Service **Consumer** that consumes the provided services

- Cloud Service **Creator** that creates at least some of the services provided by the Provider

It is important to point out that the interactions between the Consumer and the Provider, on one hand, and between the Creator and the Provider, on the other hand, take place using *standardized* APIs, as shown in the following figure:

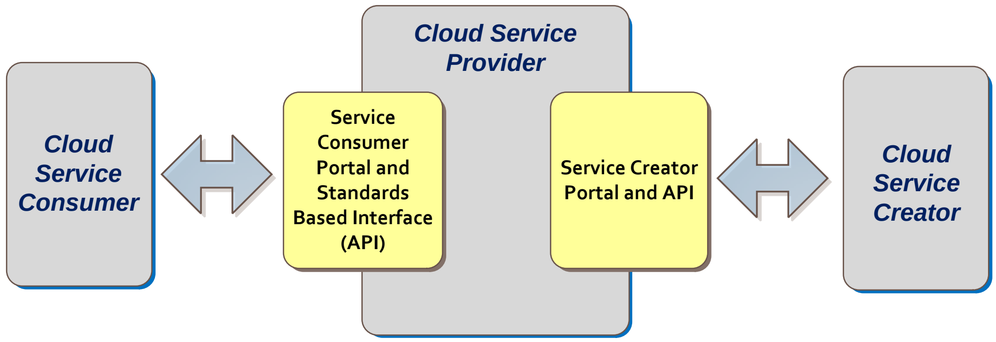

Figure 2-4 Cloud Service Roles Overview

From the point of view of a target EA that involves Cloud Computing, the above distinctions are important, both with respect to the roles (Provider, Creator, and Consumer) and with respect to the related APIs (provider API vs. creator API). In a sufficiently large organization - one with a capability to host its own cloud platform (e.g., the state) - the distinction between the roles may seem blurred on occasion; for example, the same organization that is the Provider of cloud services, may also be largely responsible for creation of those services, and even may be, at the same time, a consumer of some of the services being provided. Nevertheless, a strict logical separation of the roles should be maintained, because logical architecture corresponding to these roles is different - this is discussed later in the document in subsections about internal components of Cloud Service Provider, Service Consumer, and Service Creator. Moreover, Service Consumer and Service Creator roles should not be concerned, beyond the published APIs, with the architectural and technical details of the cloud that are of primary concern for the Service Provider. Similarly, a clear separation must be retained between the APIs (and related services and interfaces) that are utilized by Service Consumers as contrasted with the APIs to be used by Service Creators.

2.5 Cloud Service Models

There are a number of Cloud Service models available. Depending on the purpose and desired characteristics of given cloud services, one model may be a better fit than the other for the business requirements at hand. The models differ in how much responsibility in managing the IT component stack (ranging from Facilities, Network, Storage, Servers, up to Application, its configuration and data) is assigned to Service Consumer as compared to Service Provider. The most common and representative service models are shown in the following figure:

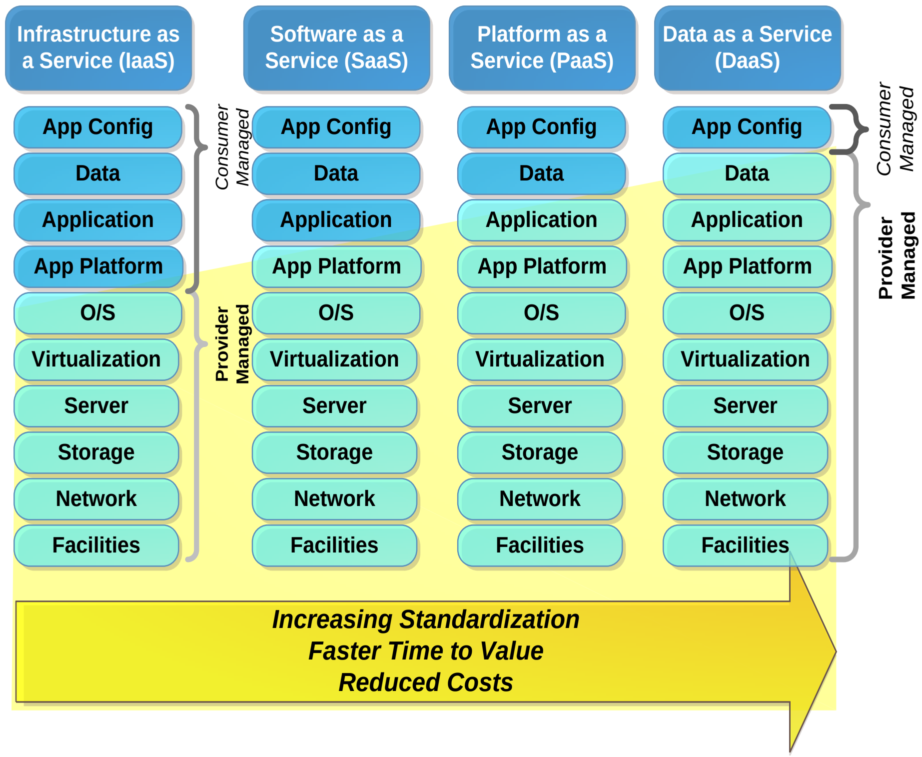

Figure 2-5 Cloud Service Models

The above figure orders service models into a sequence - IaaS, SaaS, PaaS, and DaaS – in which increasing responsibilities are given to the cloud Service Provider and, correspondingly, the extent of responsibilities of the cloud Service Consumer decreases. A useful starting point comparison for the CC service models is the non-cloud service model, in which virtually all responsibility is allocated to the "Consumer " – with the exception of some infrastructural responsibilities (e.g., with respect to network or physical facilities) that are typically assigned to dedicated organizational units. The common CC service models, some of which are shown in the above figure, are described in the subsections that follow.

Please note that the figure showing Cloud Service Models should not be treated as indicative of the capabilities available in the actual deployment of the given service model by Service Provider. For example, availability of a DaaS service model does not necessarily imply that the other service models are provided as well (PaaS, SaaS, and IaaS) in the same cloud solution – this may be the case but does not have to be the case. It is essentially up to the Service Provider to support specific service models. When in doubt, it is the published service APIs that helps determine the actually supported service models.

For further description of the services and examples of those services, please refer to the section "Cloud Services".

2.6 Cloud Service Provider Components

Cloud Service Provider is an organization responsible for providing cloud services to Cloud Service Consumers, which owns the cloud service management platform, and may own the physical resources or partner with another organization. The Provider may have internal Cloud Service Creators in their organization or utilize an external third-party. The Provider typically uses a Provider Portal for cloud service management.

The following figure shows main components related to Cloud Service Provider:

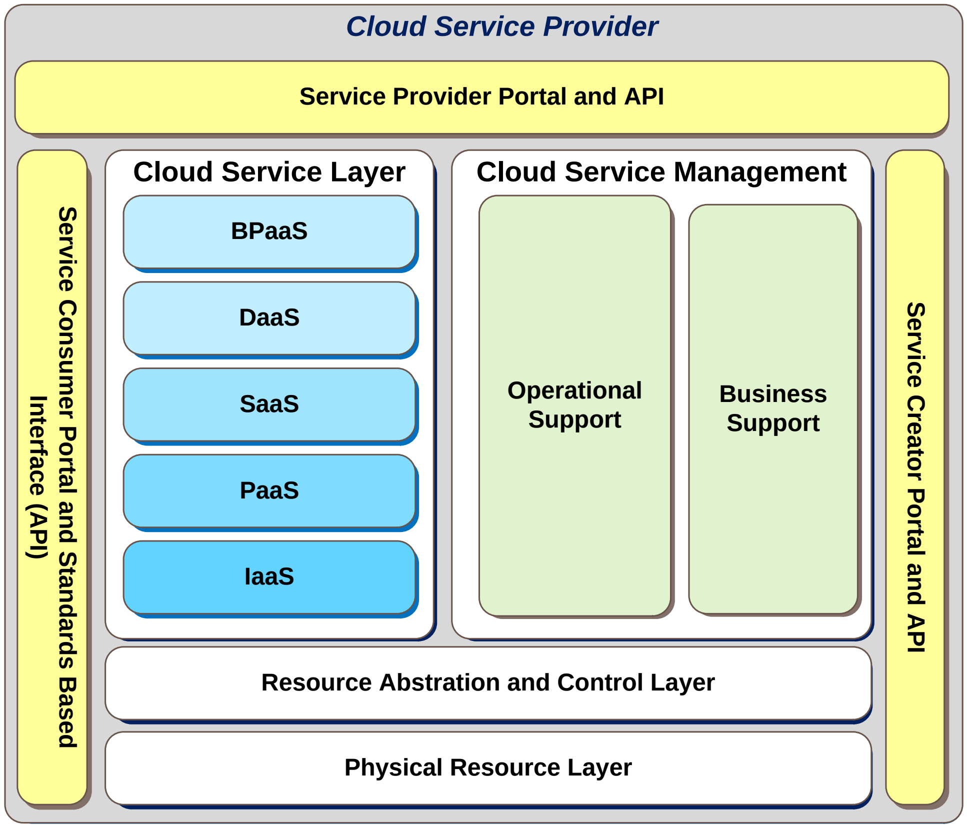

Figure 2-6 Cloud Service Provider Component Overview

Each major area from the above figure is described in the subsections that follow.

2.6.1 Cloud Services

The following figure provides an overview of Cloud Services:

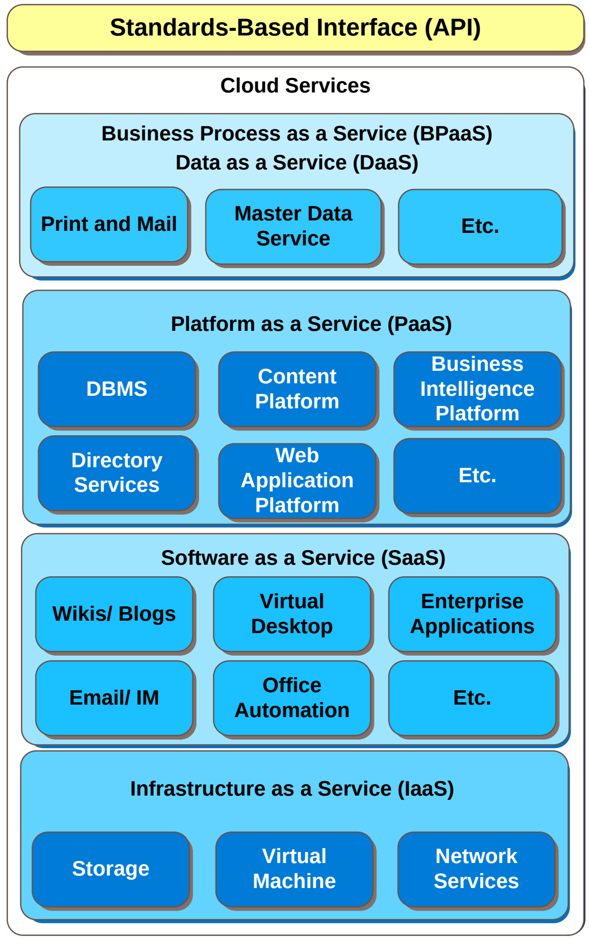

Figure 2-7 Cloud Services Overview

The Cloud Services can be grouped as follows:

- ***Data as a Service (DaaS)***, which supports a multi-tenancy Master Data Management Platform; for example, Party (customer, employee, business organization etc.) Master Data Service can be implemented as a DaaS in the cloud
- ***Business Process as a Service (BPaaS)***, which support Business Process Engine for execution of Business Process definitions, typically with multi-tenancy capabilities. Both stand-alone business processes (such as Eligibility Determination) and sub-business processes (that can be integrated into main business processes managed by a consumer's BPM application, such as Payment Processing) can be implemented as BPaaS in the cloud
- ***Platform as a Service (PaaS)*** allows for the delivery of facilities that are required to support the complete lifecycle of building and delivering applications and services using the cloud infrastructure. PaaS can include a set of programming languages, software and product development tools, and runtime platforms. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems or storage, but has control over the deployed applications and possibly the configuration set-up. PaaS services include application design, development, testing, deployment, hosting, team collaborations, web service integration, database integration, security, scalability, storage,

state management and versioning. PaaS may also include whole functional platforms, such as Enterprise Content Management (ECM), Business Intelligence (BI), Directory Services, DBMS, or Web Application Platform

- ***Software as a Service (SaaS)*** that enables the delivery of applications that are licensed for use, and which are provided to consumers on demand over a public (Internet) or private network. SaaS is most often implemented to provide business software functionality at a low cost while allowing the consumers to obtain the same benefits of commercially licensed, internally operated software without the complexity of installation, management, support, licensing and high initial costs. Examples of SaaS include services such as Email, IM, Virtual Desktop, Wikis and Blogs, and Enterprise Business Applications. SaaS can also service Shared Business Services (such as SSN Verification, or Legal Presence Verification), or Shared Enterprise Applications (such as Employee Benefit Management or Time and Labor Tracking)
- ***Infrastructure as a Service (IaaS)***, which provides for the delivery of computer infrastructure (resources) as a service over a public or private network, including servers, software, data center space, virtualization platforms and network equipment. As a result, IaaS can offer tangible advantages, such as near instantaneous provisioning, scalability, cost- effectiveness and flexibility. Although the simplest of the Cloud Services, IaaS encapsulates potentially complex infrastructure involving the following components:
    - Hypervisors (software than runs virtual machines) installed on managed hardware infrastructure
    - Virtual Machine(s) encapsulating specific operating system and its configuration for the environment
    - Virtualized Storage that provides insulation from the actual physical storage mechanisms and devices
    - Virtual Network (s) that hide from consumers of IaaS the details of physical networks

2.6.2 Cloud Service Management

Cloud Service Management falls into two groups:

- *Operational Support*, including management of virtualization elements, of SLA, of IT assets, etc.
- *Business Support*, involving areas such as Customer Management, Contract Management, Accounting and Billing, or Reporting and Billing

The following figure shows Cloud Service Management components:

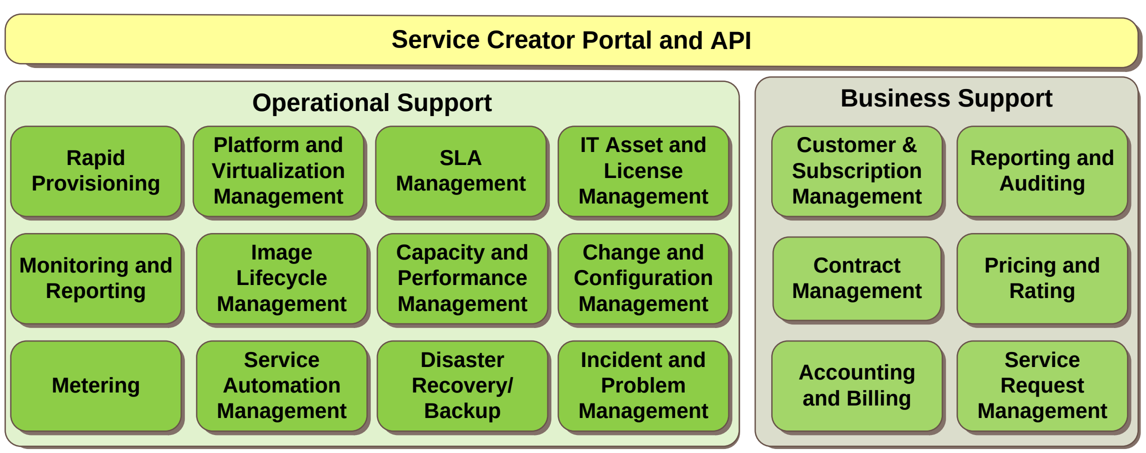

Figure 2-8 Cloud Service Management Overview

As shown in the above figure, both Operational Support and Business Support are typically accessible using two distinct ways of interaction: a portal for human actors and an API for automated/programmatic access (e.g., when used by other applications or systems).

2.6.3 Service Consumer Portal

Service Consumer Portal provides to Service Consumers an easily accessible mechanism for identifying and managing service instances, and then for reporting on actual usage (and possibly usage patterns) of the services. The following figure shows Service Consumer Portal components:

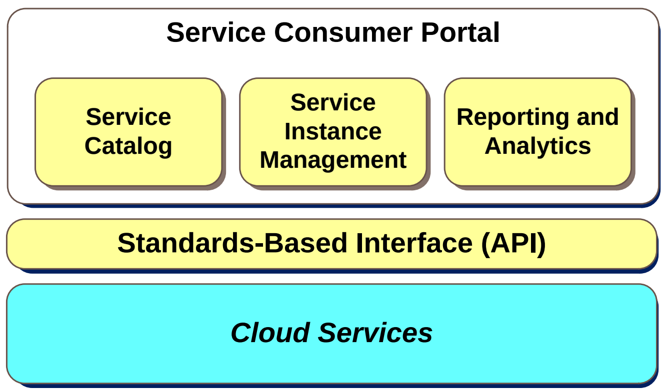

Figure 2-9 Service Consumer Portal Components

The key components of Service Consumer Portal are as follows:

- Service Catalog presents services available to a given Consumer that meet the technical and non-technical constraints (such as SLAs or regulatory constraints) that apply to the Consumer
- Service Instance Management allows for provisioning, starting, monitoring, and stopping of service instances
- Reporting and Analytics provides real or near-real time data on the actual usage of services provisioned by the Consumer

It should be noted that the interaction between the above components and the available cloud services takes place using a published, standards-compliant API.

2.6.4 Resource Abstraction and Control Layer

The following figure shows Resource Abstraction and Control Layer components, namely, Image Metadata, Image (template), and Hypervisor:

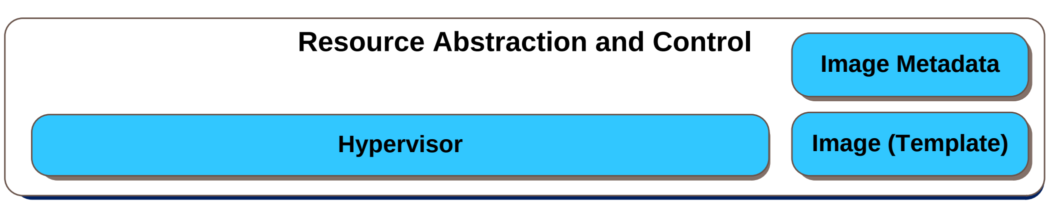

Figure 2-10 Resource Abstraction and Control Layer

The main components of the Resource Abstraction and Control Layer are as follows:

- Hypervisor, which is a component making it possible to execute virtualized operating systems, development environments, application stacks, and so on, depending on the service model
- Image/Template and Image Metadata, which provide for fast provisioning of virtual environments through replication of virtual image replication

2.6.5 Physical Resource Layer

The following figure shows Physical Resource Layer components:

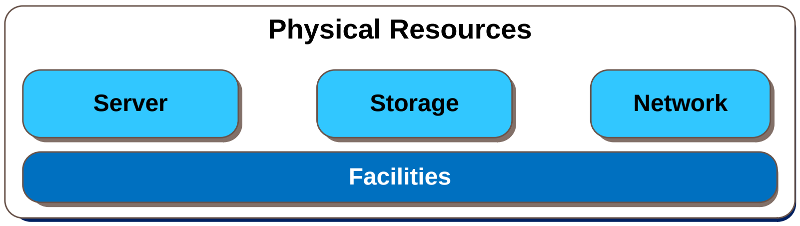

Figure 2-11 Physical Resource Layer

Physical Resources include the following:

- Physical Servers and Physical Storage, typically deployed as server farms at Service Provider's facilities
- Physical Network
- Physical Facilities for housing servers, storage, and elements of the network

2.7 Service Consumer Components

Service Consumer is an organization, a human being or an IT system, which consumes service instances delivered by a Provider. Consumers may continue to have in-house or hosted IT managed in a non-cloud fashion, but they need *Cloud Service Integration Tools* to integrate cloud services with existing in-house or hosted non-cloud IT. Consumers require a *Service Consumer Portal* for self-service delivery and management. Using the portal, Consumers browse the service offering catalog and trigger service instantiation and management from there. It is important to note that Consumers need a well-defined Exit Strategy.

The following figure shows main components related to Cloud Service Consumer:

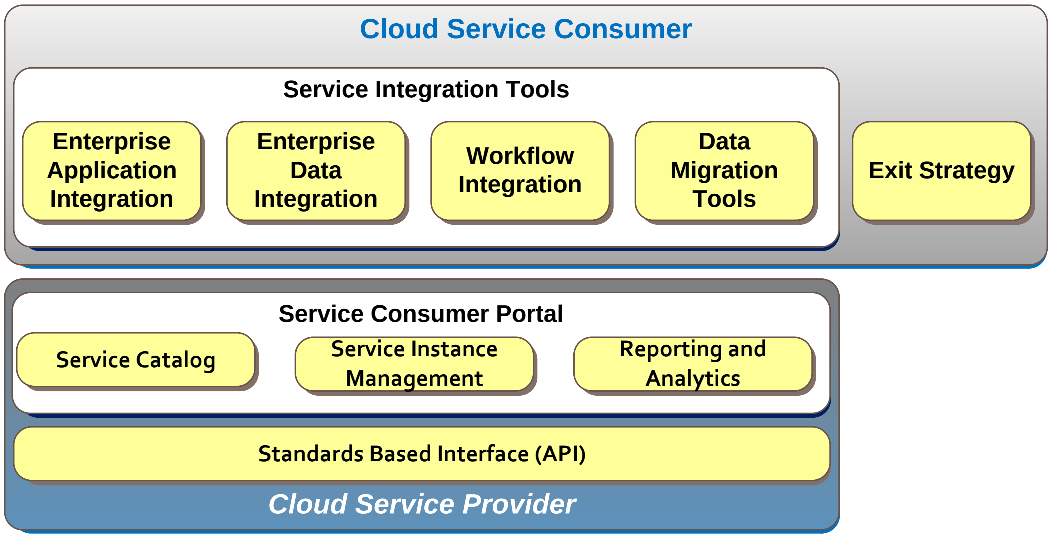

Figure 2-12 Cloud Service Consumers

The main components used by Service Consumer include the following:

- Enterprise Application Integration (EAI) components
- Enterprise Data Integration
- Workflow Integration
- Data Migration Tools

Please note that the above areas are covered in separate Reference Architectures, namely in the Enterprise Application Integration (EAI) RA and the Master Data Management (MDM) RA – both documents are part of the CEAF 2.0 package.

2.8 Cloud Service Creator Components

A Cloud Service Creator is a role that can be assumed by an organization, a human actor or actors, responsible for designing, implementing, and maintaining cloud services that can be made available by a Cloud Service Provider.

The Service Creator role can be assumed by actors belonging to the Cloud Service Provider organization, or to a separate organization providing dedicated Creator services (e.g., an external vendor in an IT Project). Regardless of who actually performs the role, some of the elements of the cloud solution as made available by the Service Provider are expected to be leveraged by Service Creator; e.g., a Service Management portal and/or the Service Creator API as specified by the Service Provider.

The following figure shows the key building blocks relevant to the Service Creator role:

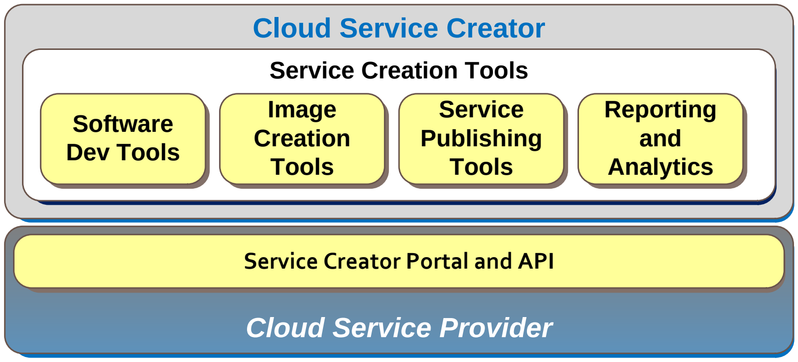

Figure 2-13 Cloud Service Creator

The main components used by Cloud Service creator include the following:

- Software Development Tools that include possibly more than one development *stack* (that is, a consistent set of tools used to create service-related software in a given technology, e.g., JEE-based or .Net-based)
- Image Creation Tools that allow for creation of virtualized packages to be executed by virtual machines/hypervisors
- Service Publishing Tools that support deployment of services
- Reporting and Analytics that allow for tracking and measuring of service usage

2.9 Cloud Deployment Models

The NIST model of Cloud Computing distinguishes four cloud deployment models, as shown in the following figure:

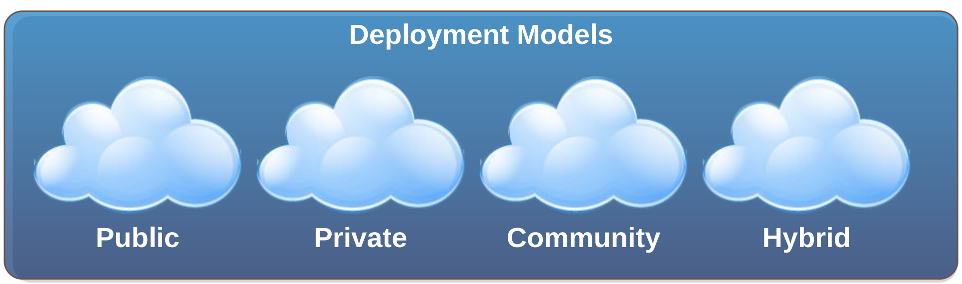

Figure 2-14 Cloud Deployment Models

The available deployment models in the cloud are public, private, community, and hybrid models, as follows:

- A *public cloud* is where resources, such as storage and applications, are made available to multiple consumers by a Service Provider, via a web application or web service over the Internet. The resources are therefore located at an off-site location that is controlled and managed by the Service Provider. These are typically low-cost or pay-on-demand and highly scalable services
- A *private cloud* infrastructure is operated for a single organization. It may be managed by the organization or a third party and may exist at an on-site or off- site location. Private cloud services offer the provider and the user greater control over the cloud infrastructure, improving security, compliance, resiliency and transparency. Private clouds, however, require capital expenditure, operational expenditure and a highly skilled IT team [1, 4]

- *Community clouds* are controlled and shared by several organizations and support a specific community that has shared interests, such as mission, policy, security requirements and compliance considerations. It may be managed by the organizations or a third party and may exist at on-site or off-site locations, and the members of the community share access to the data and applications in the community cloud. Community cloud users therefore seek to exploit economies of scale while minimizing the costs associated with private clouds and the risks associated with public clouds [3, 4]
- A *hybrid cloud* is a combination of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability. Applications with less stringent security, legal, compliance and service level requirements can be outsourced to the public cloud, while keeping business-critical services and data in a secured and controlled private cloud

In each of the Cloud Deployment Models it is important to distinguish the party responsible for architecting, design, deployment, and/or operation of the given cloud solution:

- *Public* Clouds are architected, designed, built and operated by a Cloud Service Provider.
- *Private* Clouds come in the following varieties:
    - Custom Private Cloud: Architected, designed, built and operated by an organization for its own use
    - Packaged Private Cloud: Architected, designed, built and operated by a vendor for a single customer use
    - Virtual Private Cloud: Provider ensures that a customer is completely isolated from others
- *Community* Clouds are cloud infrastructure shared by several organizations that have shared concerns and mission
- *Hybrid* Clouds combine cloud models or cloud and non-cloud (on-premises) models of deployment

In practice, the actual determination of the best fit with respect to the deployment model of the cloud obviously depends on a larger number of factors than just those pointed out above. However, the models already available, and their derivatives (in case of the private deployment) do not introduce artificial constraints when it comes to deciding on the deployment model. Additionally, the existing deployment models make it possible to consider migration between various deployment models as part of an EA Roadmap.

3 Cloud Computing Reference Architecture Description

This section provides a focused description of CC Reference Architecture (RA) using the Conceptual View which provides a summary of logical-level building blocks for CC as presented in the Section 2 above.

3.1 CC RA Conceptual View

The CC RA Conceptual View figure below brings together all major components of a CC solution in the following groups:

- Cloud Service Provider, as described in the section “Cloud Service Provider Components”
- Service Consumer, as described in the section “Service Consumer Components”
- Service Creator, as described in the section “Cloud Service Creator Components”

Note that the resulting view is quite complex. However, from the perspective of either Service Consumer or Service Creator, the architecture required to make use of a cloud-based solution is actually simpler, as it involves only the architectural building blocks in the sections marked correspondingly as “Service Consumer” and “Service Creator”. The central, and the most complex section marked “Cloud Service Provider” is neither directly visible to the other two roles, nor it is their responsibility. However, including in the diagram the architectural building blocks for which only Service Provider is responsible helps visualize the extent of that responsibility, and makes it easier to compare it with typical deployments that do not utilize the cloud.

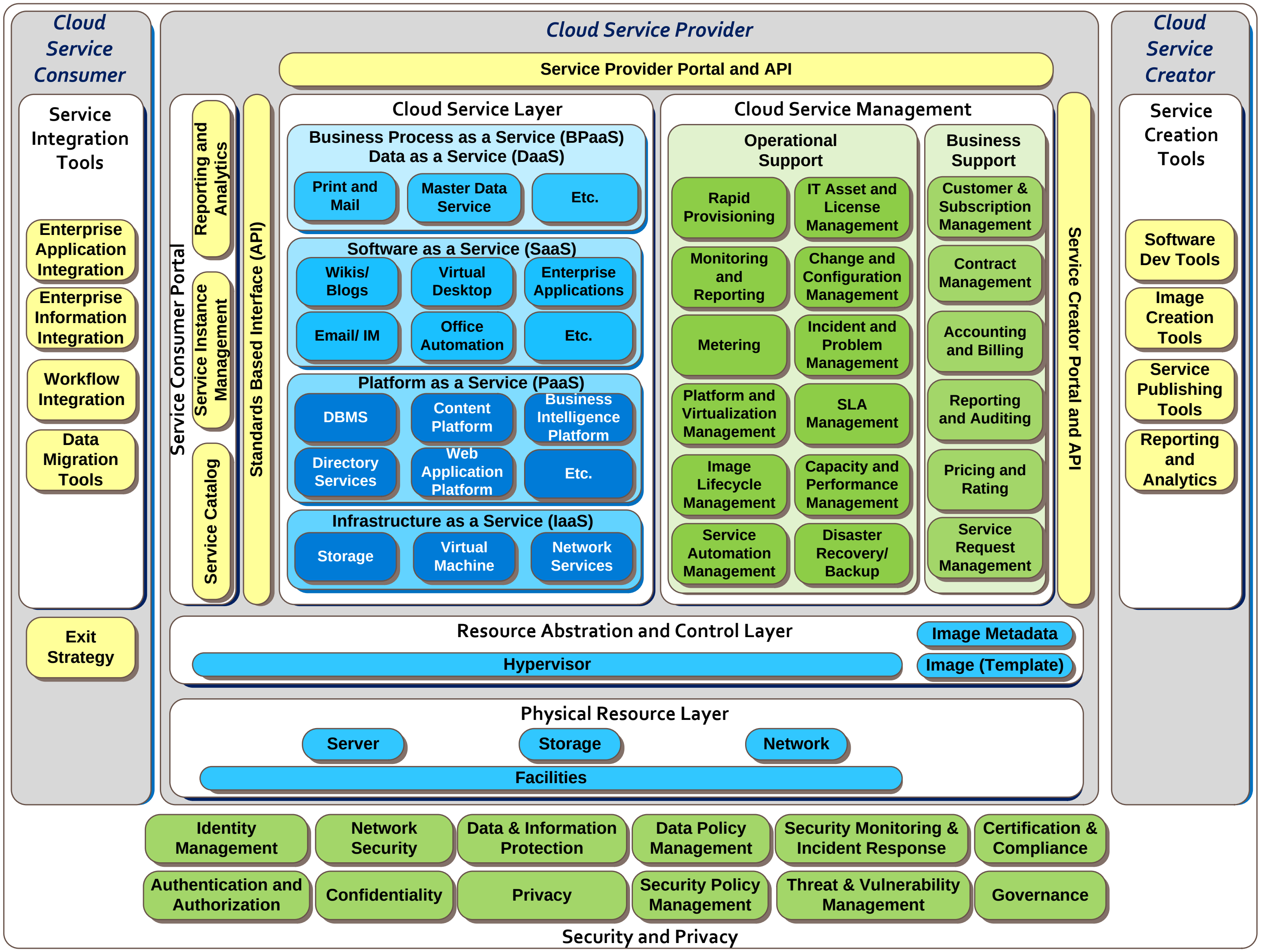

Figure 3-1 CC Reference Architecture – Conceptual View

4 RA Implementation Notes

When specifying target enterprise architecture and considering the cloud in it, one of the key challenges is identifying and evaluating the architectural elements (ranging from platforms to services and applications/systems) as candidates for migration to the cloud. When doing so, CEAF 2.0 can be helpful. More specifically, section 6 of CEAF 2.0, titled “Target Architecture Vision” describes the federated model of the target enterprise architecture and in the context of the target EA, it also introduces the distinction among *distinct, common, and core elements* of the architecture, as follows:

- *Distinct business processes* are *specific to a business unit* with little to no integration or coordination with the processes of other business units. For example, business processes associated with Debt Management, Debt Collection, and Construction Management capabilities may be viewed as distinct business processes in the context of state business operations
- *Common business processes* are those that are *common to some (but not all) business units*. For example, business processes associated with Customer Relationship Management, Inventory Management, and Supply Chain Management capabilities may be viewed as common business processes in the context of state business operations
- *Core business processes* are those that are *common to all business units*. For example, business processes associated with Finance, Human Resources Management, and Procurement capabilities may be viewed as core business processes in the context of state business operations. From state perspective, these core business processes may be found across several state agencies

Although the above distinction, as introduced in CEAF 2.0, pertains to elaborating target enterprise architectures and - more specifically - to classifying business processes in the organization in the context of the target EA, the underlying distinction can also be of used when considering various architectural elements (rather than just business processes) for migration to the cloud.

The following subsections provide a short discussion of how the Distinct/Common/Core distinction can be used when identifying business processes as candidates for migration to the cloud, and then how it can be used for determination of “clusters” of architectural building blocks that together become a candidate for migration to the cloud – because the distinction in question is applicable not just to business processes, but also to architectural building blocks in general.

4.1 Identifying Business Processes as Candidates for the Cloud

Once existing or target business processes are grouped into distinct/common/core categories, the resulting grouping facilitates ROI analysis and prioritization of the intended migration to the cloud:

- The *core* business process category is likely to offer the highest ROI (compared to the distinct and common categories) when migrated to the cloud, but the actual ability to migrate such processes to the cloud strongly depends on the maturity level of the organization with respect to identifying and practical implementation of these core processes

- The *distinct* business process category is likely to offer the lowest ROI when migrated to the cloud, unless there are other factors that affect the estimate, such as available know-how, or constraints in available infrastructure
- The *common* business process category can be expected to produce lower ROI than the core business processes when migrated to the cloud, but at the same time it may offer in practice an attractive combination of significant business gains without the level of EA maturity required for effective migration of core business processes

4.2 The Distinct/Common/Core Classification and Cloud Service Models

As already mentioned in the section “Cloud Service Models”, the cloud offers a number of Service Models. The most commonly mentioned are IaaS (Infrastructure-as-a-Service), SaaS (Software-as-a-Service), PaaS (Platform-as-a-Service), and DaaS (Data-as-a-Service). Viewed at a high level, Cloud Service Models differ in the split of responsibilities between Service Provider and Service Consumer (this is shown in the “Figure 2-5 Cloud Service Models”).

The available Cloud Service Models provide one useful perspective of considering candidates for migration to the cloud, and the CEAF-supplied distinction between distinct/common/core elements provides another perspective, which complements the first one in making it easier to work out EA migration roadmaps involving the cloud. One way of using both perspectives together is as follows:

- For every available Cloud Service Model , corresponding clusters of elements can be identified as natural candidates for migration; for example, in case of IaaS that could be Windows or *nix OS platform, in case of PaaS, specific development environment or a web application platform, in case of SaaS, an office automation solution, and so on. Identifying these candidate clusters is an important step in working out the target EA that involves the cloud
- For every identified candidate cluster, applying the distinct/common/core distinction results in a matrix of “cluster candidates” that can be examined separately with respect to their potential business impact, by following the approach suggested for identification of business processes

The above helps taking first steps in analyzing the business value of the considered migration to the cloud, and it can help rationalize and prioritize steps in the resulting roadmap, when other factors are taken into consideration (specific circumstances affecting feasibility, EA maturity level and similar).

5 Glossary

Cloud Computing (CC) is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Reference Architecture models the abstract architectural elements in the domain independent of the technologies, protocols, and products that are used to implement the domain.

6 References

6.1 Federal and State Documents

A. Mell, Peter, and Timothy Grance. "The NIST definition of cloud computing (draft)." NIST special publication 800 (2011): 145
B. The California Enterprise Architecture Framework, Version 2.0 (CEAF 2.0)

6.2 Books and Papers

1. B. Gold-Bernstein, W. Ruh, Enterprise Integration: *The Essential Guide to Integration Solutions*, Addison-Wesley Professional 2004
2. Themistocleous, M., and Irani, Z., *Benchmarking the Benefits and Barriers of Application Integration*, Benchmarking: An International Journal, 2001, Vol.8, No. 4, pp.317-31
3. Carroll, M., A. Van der Merwe, and P. Kotze. "Secure cloud computing: Benefits, risks and controls." Information Security South Africa (ISSA), 2011. IEEE, 2011
4. IBM Global Technology Services, "Getting cloud computing right", 2011
5. H. Cai, N. Wang, and M. J. Zhou, "A transparent approach of enabling SaaS multi-tenancy in the cloud," in Services (SERVICES-1), 2010 6th World Congress on, 2010, pp. 40–47

6.3 Web Sites

a. California State Chief Information Officer, California Information Technology Council, Committees http://www.cio.ca.gov/ITCouncil/Committees/Committees.html
b. Gartner IT Glossary, http://www.gartner.com/it-glossary
c. Wikipedia entry for "Cloud Computing", http://en.wikipedia.org/wiki/Cloud_computing

7 Document History

Table 7-1 Document History

Release	Description	Date
Version 1.0 Draft	Initial creation	6/19/2013
Version 1.0 Second Draft	Revised based on internal review comments	8/05/2013
Version 1.0 Final Draft	Addresses EAC review comments	10/21/2013
Version 1.0 Final	Final version	01/02/2014