Cloud Computing Reference Architecture 2.0: Overview

CC RA team / presented by Mike Buzzetti

Agenda

- Executive Summary
- Introduction & Overview
  - Historical evolution of the Reference Architecture
  - Structure of the RA 2.0
  - Target audience
  - How to apply the RA for cloud implementations
- Cloud Computing Reference Architecture Details

IBM Cloud Computing Reference Architecture

The IBM CC RA represents the aggregate experience across hundreds of cloud client engagements and the implementation of IBM-hosted clouds

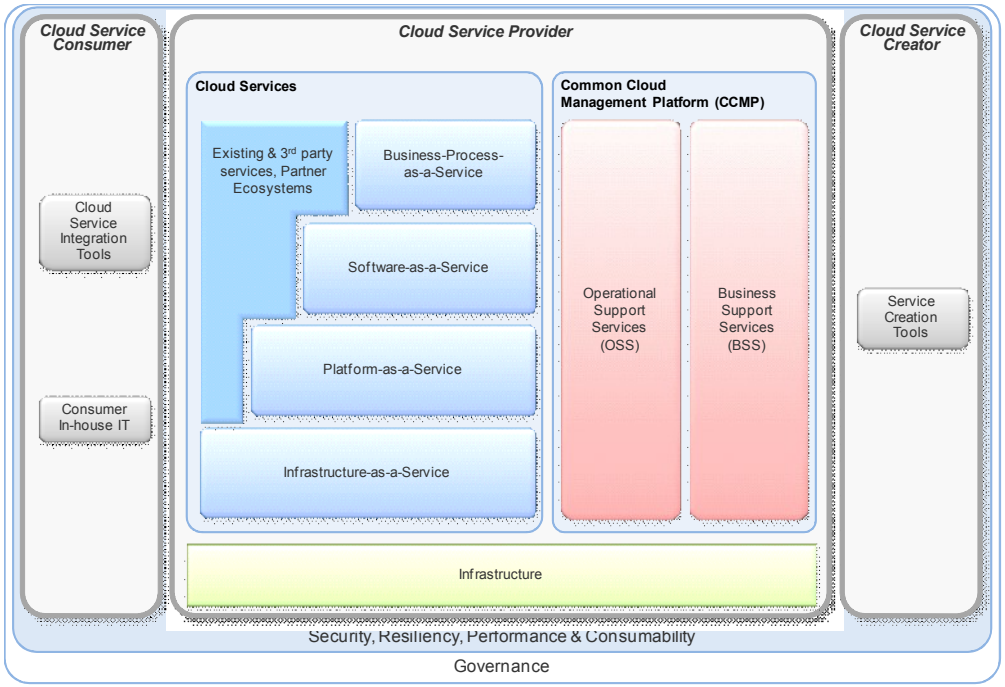
- Based on knowledge of IBM's services, software & system experiences, including IBM Research

The IBM Cloud Computing Reference Architecture (CC RA) is reflected in the design of

- IBM-hosted cloud services
- Clouds IBM implements for clients
- IBM cloud appliances
- IBM cloud service management products

The CC RA focuses on cloud specifics such as radical cost reduction while achieving high degrees of security, reliability, scalability and control

The CC RA consists of 21 detailed documents representing best-of-industry knowledge and insight on how to architect, design and implement clouds



The IBM Cloud Computing Reference Architecture ensures consistency & quality across IBM development and delivery projects

The IBM Cloud Computing Reference Architecture …

- Is based on open standards
- Delivers enterprise-class security to meet local, regional and national compliance for privacy and governance
- Combines powerful automation and services management (low touch) with rich business management functions for fully integrated, top-to-bottom management of cloud infrastructure and cloud services
- Supports the full spectrum of cloud service models, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS) and Business Process as a Service (BPaaS)
- Enables the flexible scaling and resiliency required for successful cloud economics and ROI
- Facilitates seamless integration into existing customers' environments
- Is based on our industry leading expertise with SOA for building services and service-oriented architectures

IBM Cloud Computing Reference Architecture: Development Process

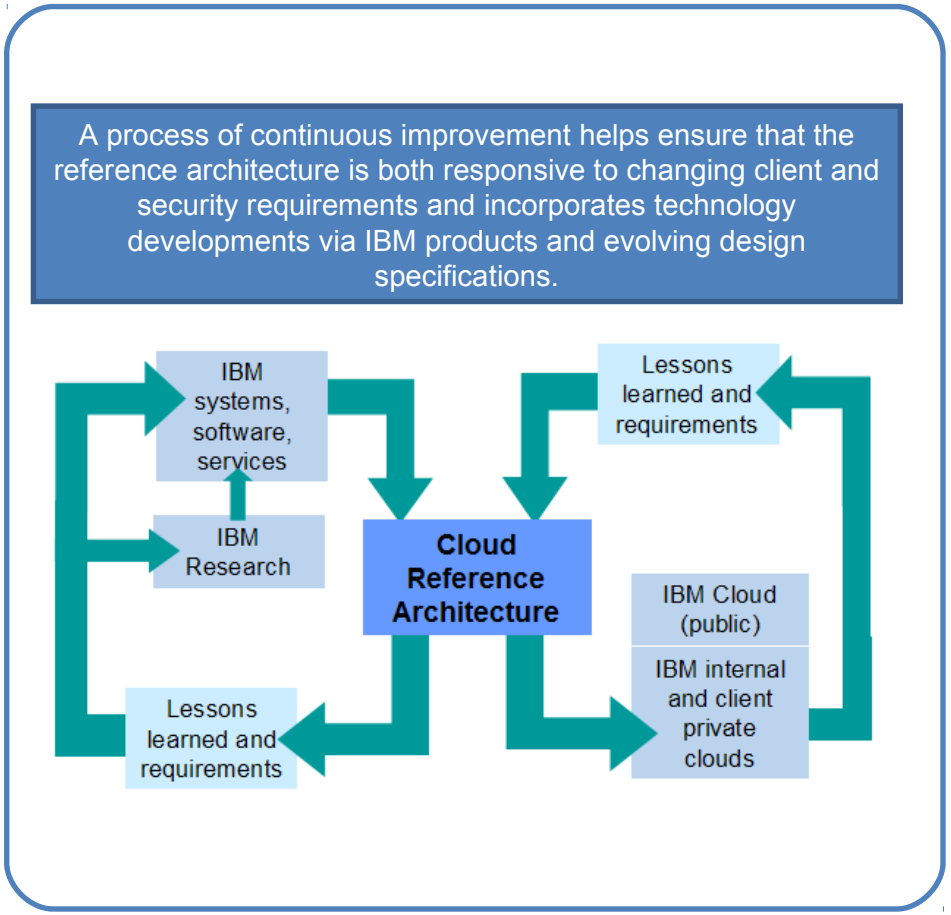
Development led by the IBM Cloud Computing Architecture Board

- Comprising technology leaders from IBM Research and IBM's software, systems and services organizations
- >50 of IBM's top cloud computing experts represent the core team

Derived from extensive client interaction combined with IBM's extensive capabilities and experience in building enterprise-class IT systems.

The CC RA provides specifications for

- the physical components of a cloud implementation (network, compute, storage, virtualization)
- Software components required to run management
- Operational processes
- Governance policies tailored for the environment or enterprise.



The IBM Cloud Computing Reference Architecture covers are broad range of important cloud-specific topics

Architectural Principles	Architecture Overview	Standards & Terminology	Cloud Service Creation	Use cases & Roles
Non-functional Requirements	Consumability	Common Cloud Management Platform	Component Model	Operational Model
Architectural Decisions	Management Processes	Security	Resiliency	Performance & Scalability
Multi-tenancy	Production Clouds	Virtualization Management across server, storage, network	Hybrid Cloud	Metering, Accounting & Rating

Structure of the IBM Cloud Computing Reference Architecture

There are two deliverables for each work product of the RA

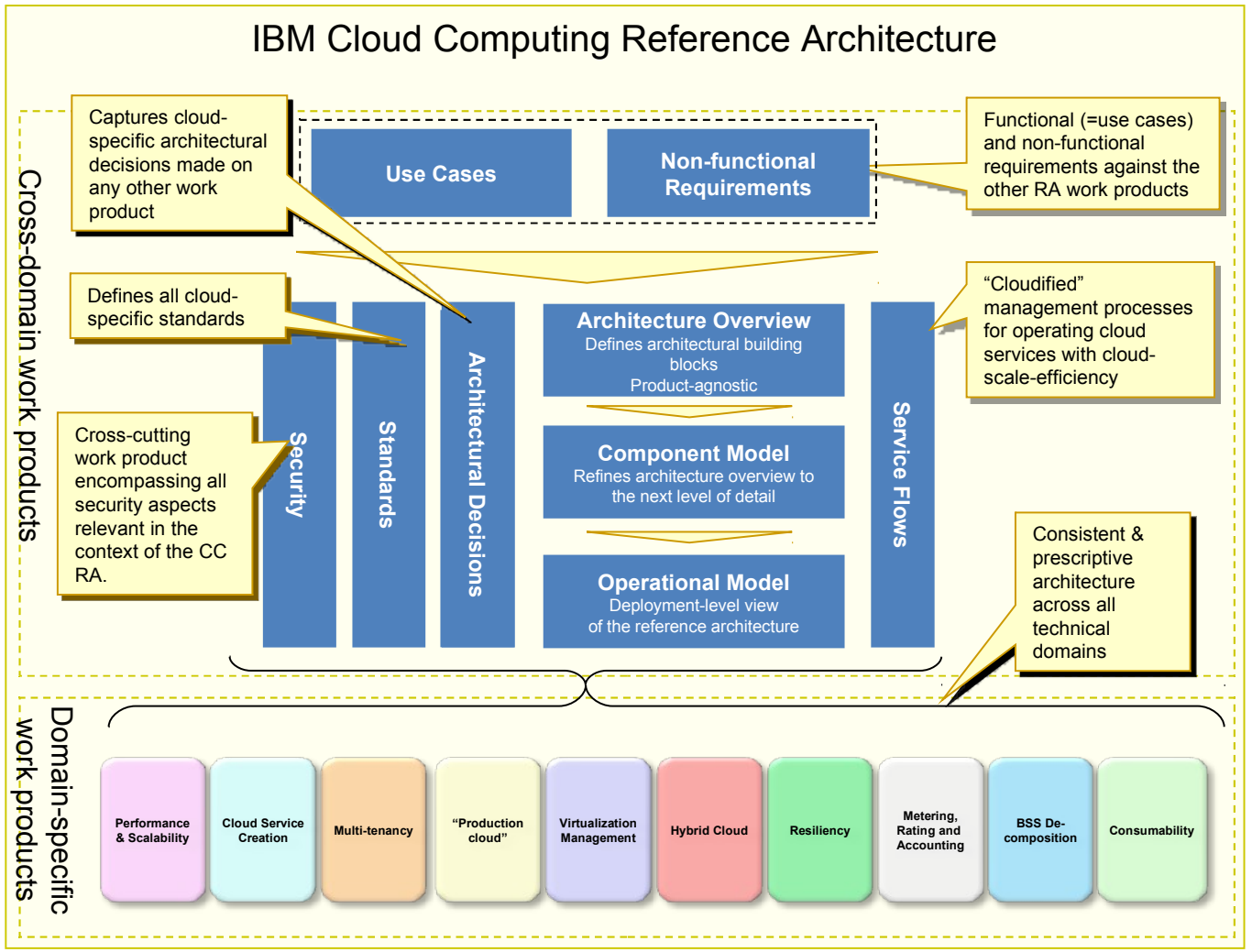
- **Document**: Detailed description of the work product, important for in-depth work
- **Presentation**: Summary of the document, for simplified consumption and quick start

The RA is defined according the Unified Method Framework (UMF)

→Simplifies field adoption since all IBM field architects attend UMF training by default

Each work product includes one or more "applied patterns"

- An applied pattern illustrates how the respective work product could be used / was used in a specific implementation



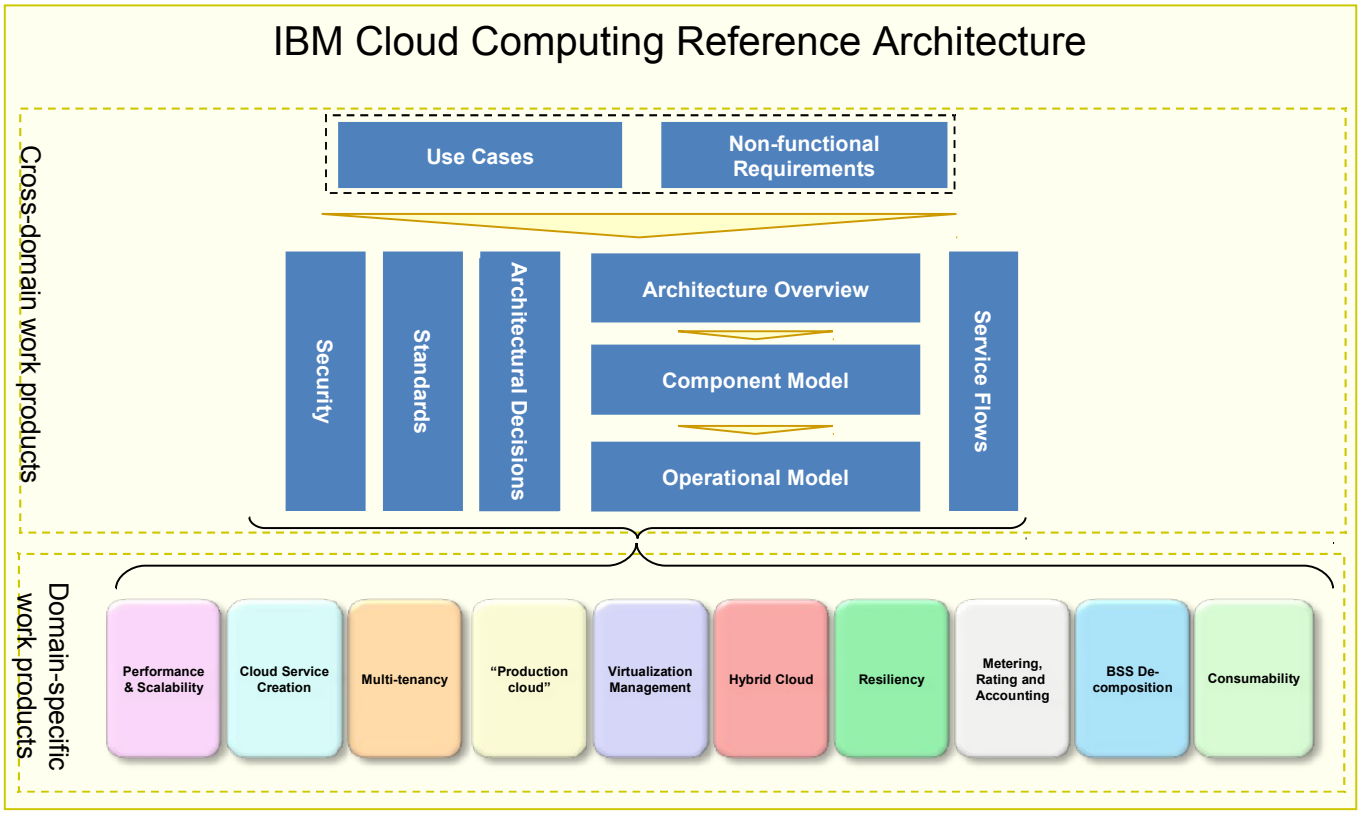
IBM Cloud Computing Reference Architecture: Domain work product details

How to implement a cloud to achieve optimal performance & scalability

Guide for creating cloud services and how to exploit management functions.

Considerations for implementing multi-tenancy in a cloud environment

Example for how to implement a cloud management system for offering production-level cloud services



Definition for how to use virtualization management across server, storage and network

Architecture & implementation considerations for implementing a hybrid cloud

Resiliency considerations for implementing a cloud

Details on metering, rating and accounting, including prescriptive guidance with respect to implementation

Detailed definition of BSS domains and how they should be implemented from a product perspective

How to optimize the end to end experiences related to cloud services

The IBM Cloud Computing Reference Architecture – Aggregating past & present experience and future requirements

CC RA

=

Harvesting
("Learning from cloud deployments in the ***past***?")

- Harvest learnings & material from cloud implementations while factoring out short cuts taken by implementation team (e.g. due to constrained time lines).
- Continuously ongoing harvesting process due to additional learnings during each phase

+

Refinement
("Improve the CC RA content **presently** available")

- Continuous improvement of CC RA content based on feedback from CC RA exploiters & stakeholder
- Be clear about which parts of the RA are already addressed in today's IBM cloud offerings and which aspects still have to be addressed

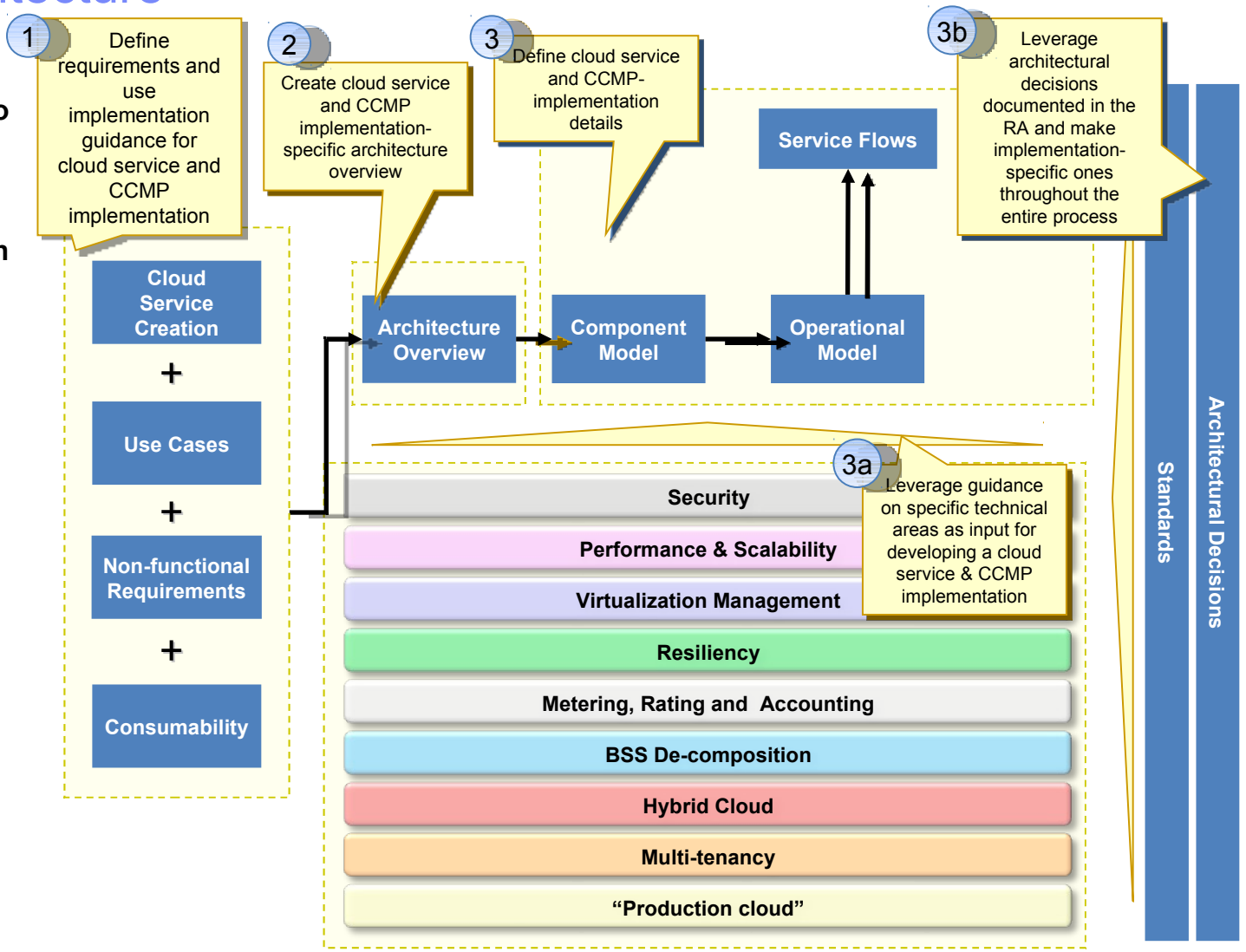
+

Forward looking
("Which ***future*** topics are relevant for cloud computing")

- Today's cloud implementations are not necessarily optimal implementations (due to time & resource constraints)
→ Need forward-looking guidance on ideal target implementation
- Examples are:
  - generalization of CCMP services
  - layered cloud services
  - NFRs, e.g. security, resiliency
  - Hybrid clouds

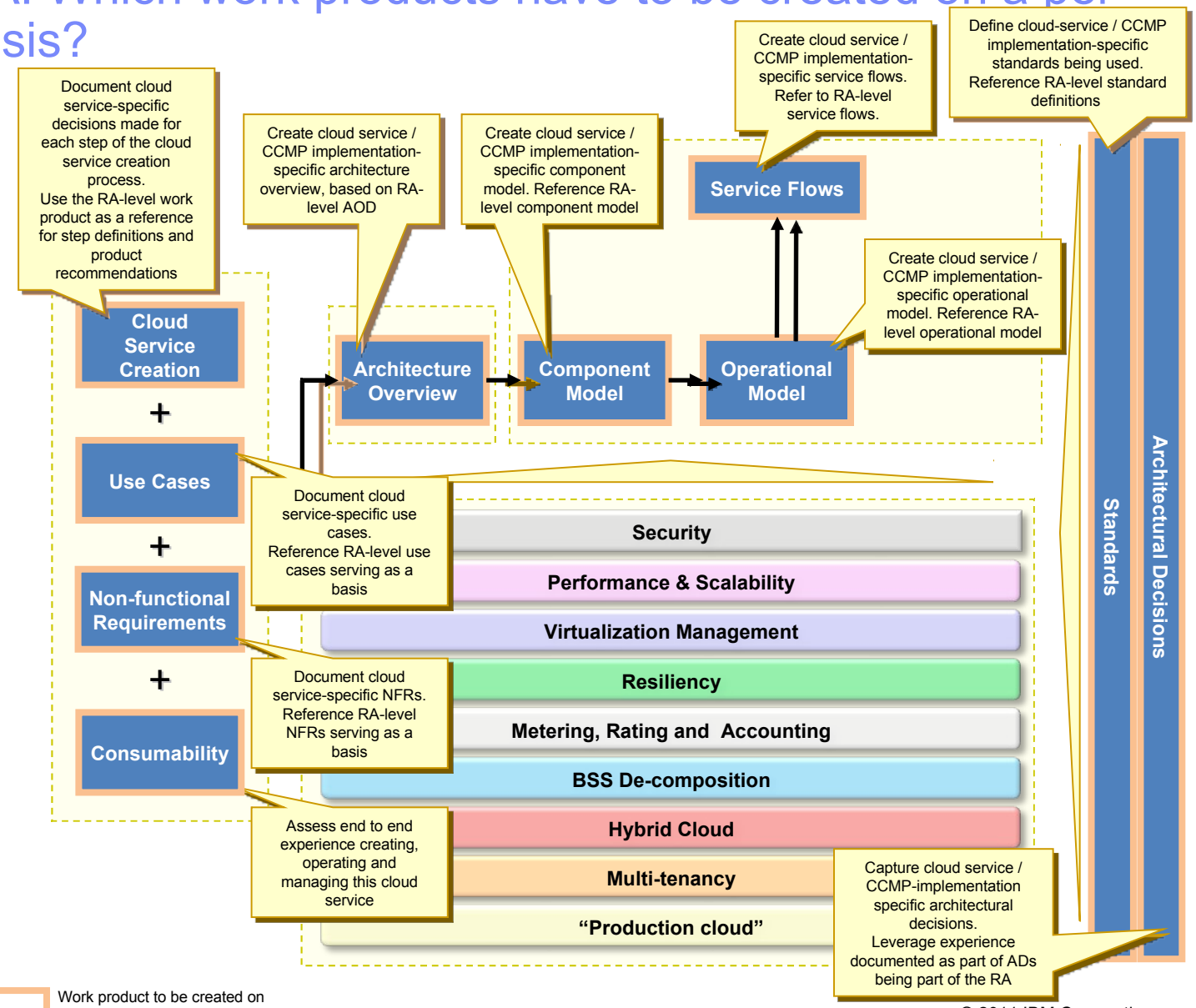
How to implement a cloud using the IBM Cloud Computing Reference Architecture

- **There is a well-defined process for using the RA to implement a cloud service**
  - **A cloud is constituted by a CCMP implementation and an arbitrary number of cloud services delivered and managed by it**
- **A dedicated presentation and document is available describing the process for using the RA to develop the management aspects for a cloud service**
- **RA work products also serve as authoritative reference on specific technical topics for education purposes**

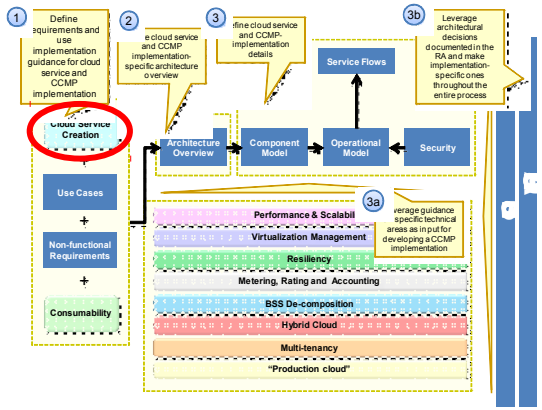


Using the CC RA: Which work products have to be created on a per cloud service basis?

- **The RA contains two types of work products**
  1. **Work products only serving as a reference documentation for specific aspects of implementing a new cloud: All "domain-specific" work products except "Cloud service creation"**
  2. **Work products serving as the basis for a cloud-specific work product: All "cross-domain" work products plus "Cloud Service Creation"**



Cloud Computing Reference Architecture Details



Cloud Service Creation

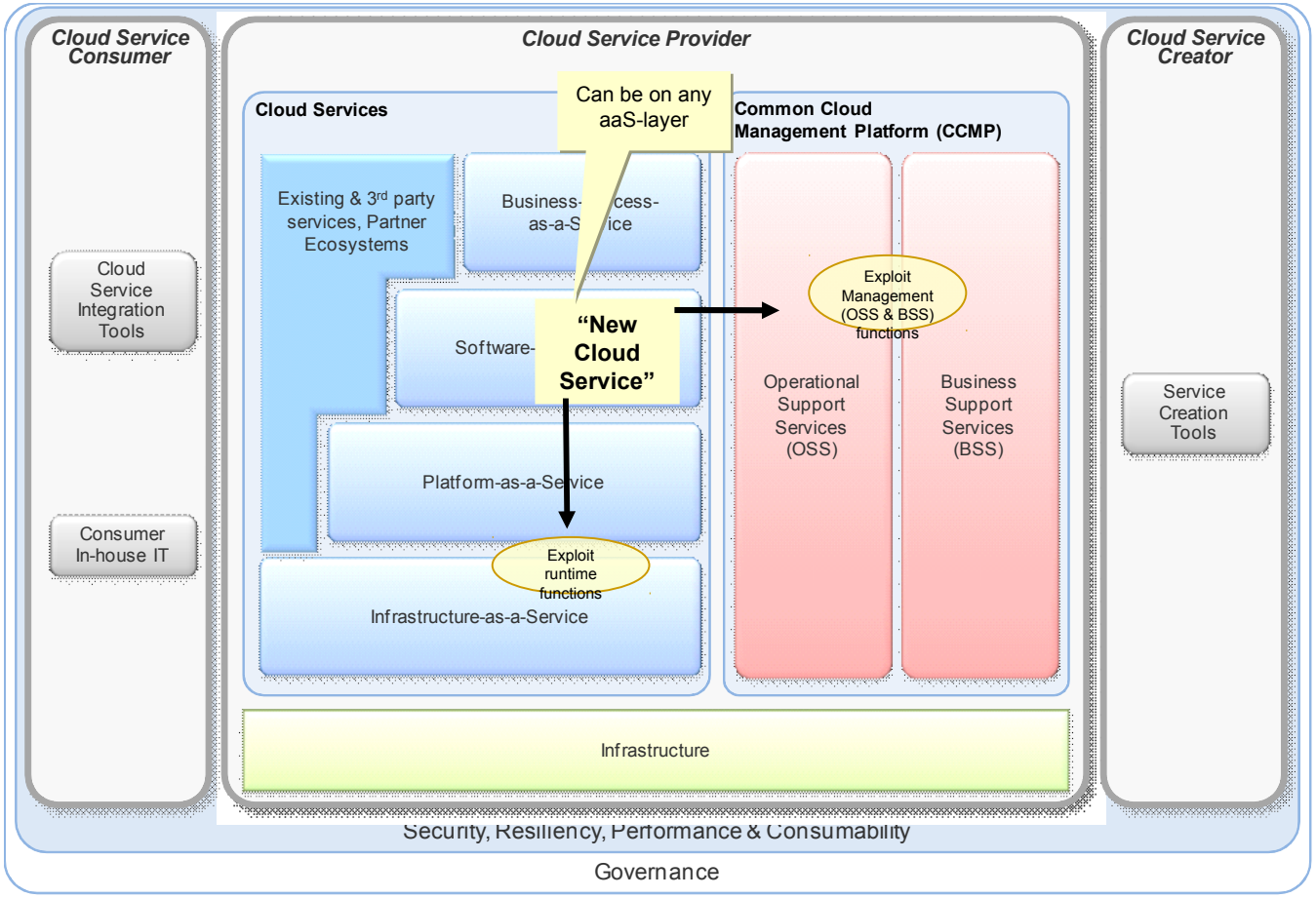
“Cloud Service Creation” – Scope and Overview

Objective:

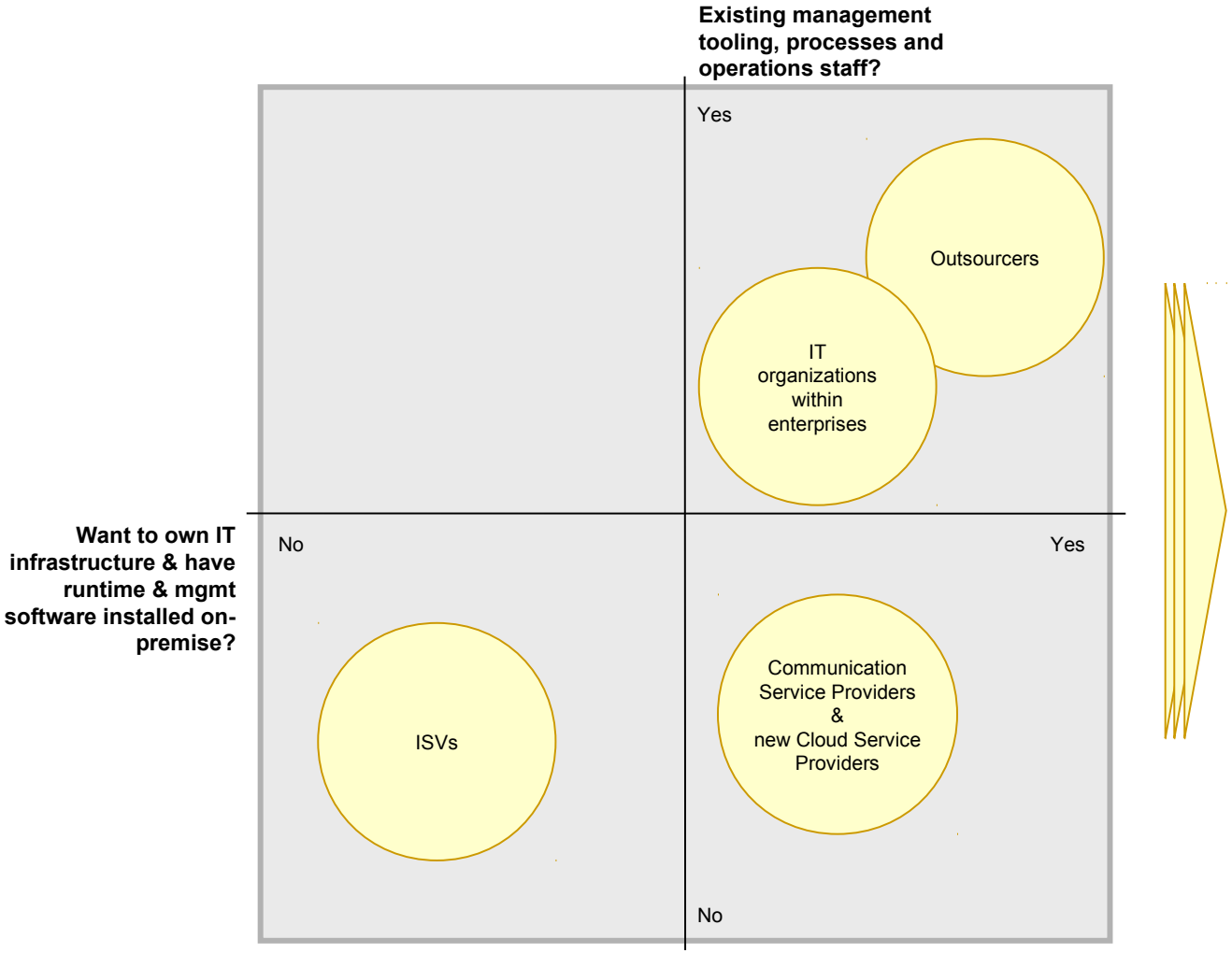
Describe all aspects around developing a new cloud service

- Scope is end-to-end process for developing a new cloud service.

  Focus is on the management aspects as they apply generically across all types of cloud services
- Provide detailed description of basic terminology relevant for creating cloud services
- Provide prescriptive product recommendations relevant for each cloud service creation step and tooling around that
- Provide illustrative examples (“applied patterns”) for how a new cloud service can be created



Developing cloud services – four major segments: Overview

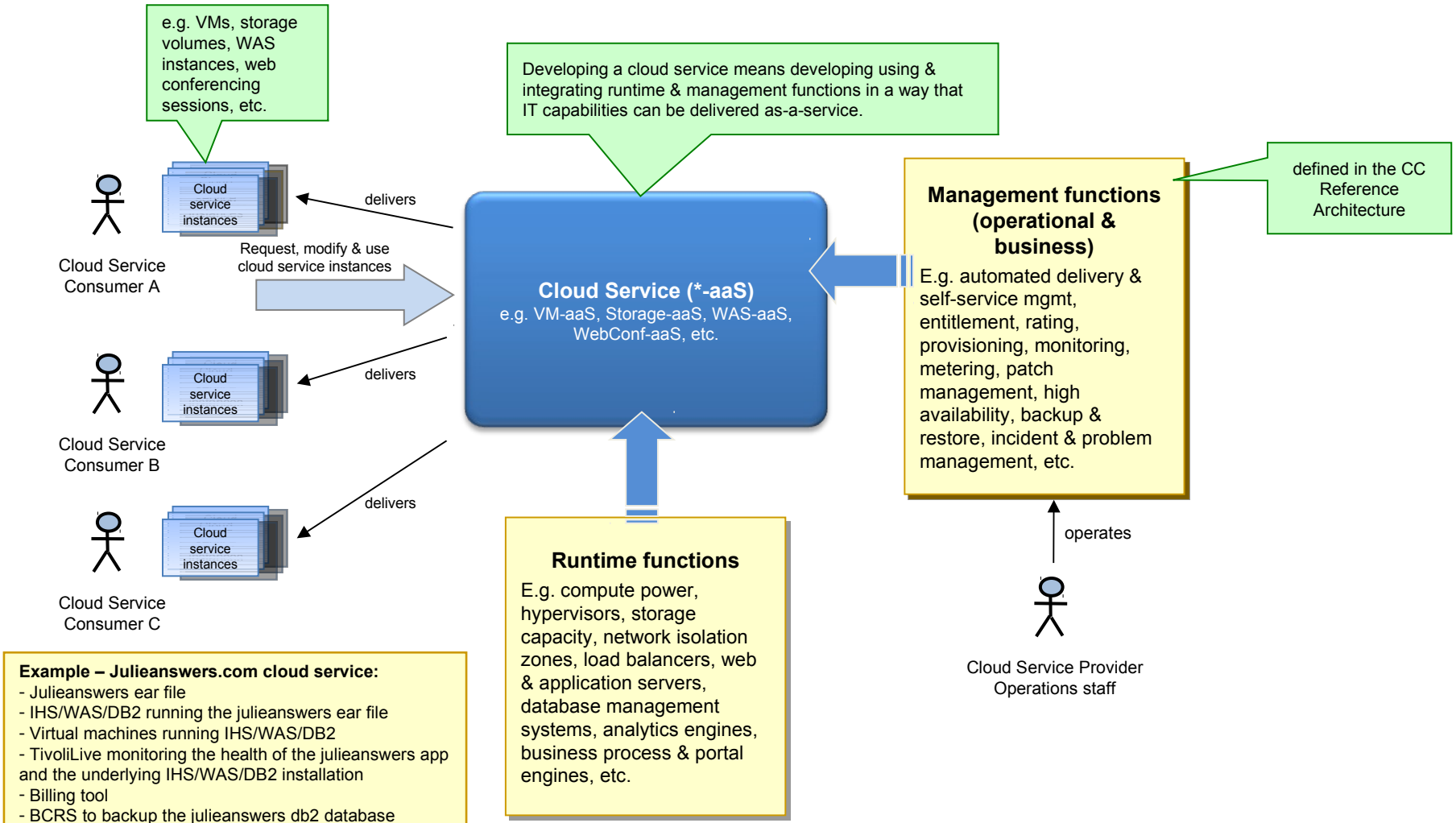


Although having different motivations, software & process legacy and requirements, there is a common approach for creating cloud services across all segments

Different scenarios for offering cloud services require
- different delivery models for management software ("build-it-yourself" vs. IBM-hosted)
- different levels of integration with existing management tools

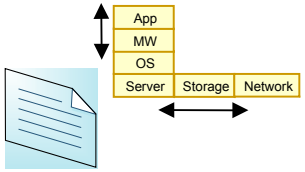
Anatomy of a Cloud service – Highlevel overview

The internal structure of a cloud service is the integration of all runtime and management functions used to deliver, run and manage instances of the respective cloud service.



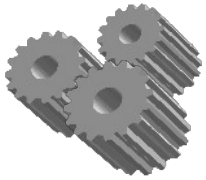
12 steps towards creating a cloud service

1. Specify cloud service description



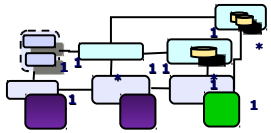
Describe function, price , SLA of cloud service, incl. management scope

2. Implement runtime functionality



Examples: Select off-the-shelf hypervisor (VMaaS), implement custom app (e.g. LotusLive)

3. Define unit of delivery & rating



Examples: VM, file system, distributed app, virtual IP address, queue, web conference, RDBMS, 3-tier business app, etc.

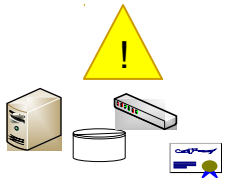
4. Implement self-service delivery & management functionality

Examples: “Create VM, add more nodes to WAS cluster, change max # of seats for LotusLive web conf

5. Implement monitoring metrics & event correlation rules

Select existing agent / implement new agent for monitoring JVM heapsize, hypervisor swap file size, # of processes, etc.

6. Implement incident, problem and asset mgmt processes

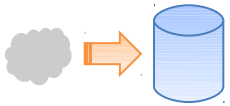


Incident, problem & asset mgmt process is specific to cloud service → customization needed

7. Implement resiliency SLA

Examples: HA for management system, delivered WAS cluster must be highly available

8. Implement backup approach

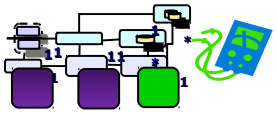


Examples: Backup all VMs, backup DB of LotusLive application

9. Implement security functions

Implement authentication, auditing, data protection, governance & audit

10. Implement cloud service specific billing metrics



Examples: CPU/hour, # of DB transactions, GB/month, # of users/webconf/hour, etc.

11. Implement rates for charging cloud service consumption

Examples: $0.11/VMhour; $0.19/MBsTransferred; $0.02/webconference; $0.05/fraudAnalysis

12. Register cloud service to service catalog

A cloud service must be registered to the service catalog to be externally accessible, entitlements need to be configured,

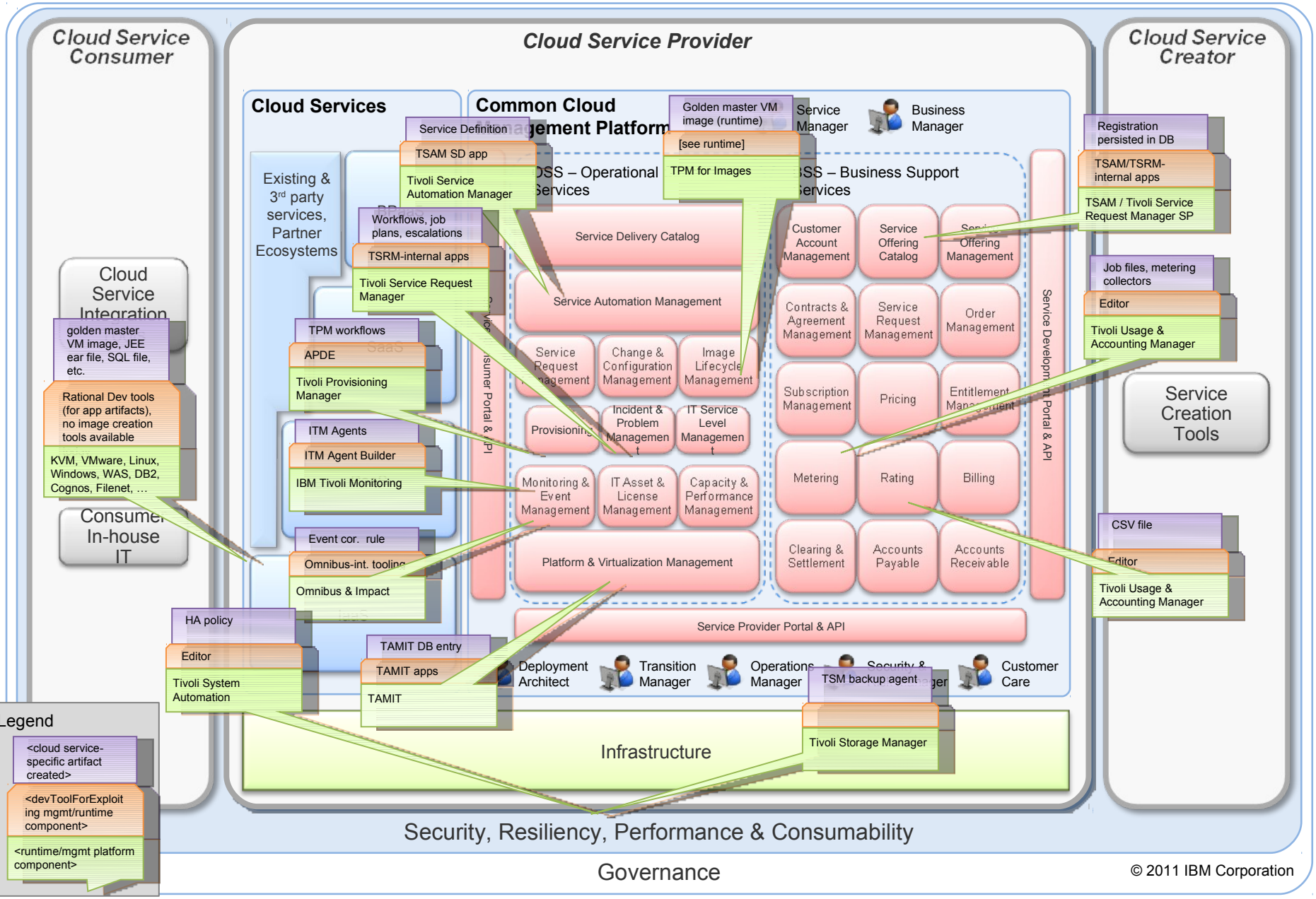
Overview: Recommended management products, to be exploited on a per cloud service basis

Step	Products & technologies to build your own cloud service
1. Specify cloud service description	n/a (cross-cutting)
2. Implement runtime functionality of the cloud service	KVM, VMware, Linux, Windows, WAS, DB2, Cognos, Filenet, …
3. Define Unit of delivery & rating	n/a (cross-cutting decision)
4. Implement self-service delivery & management functionality	TSAM
5. Implement monitoring metrics & event correlation rules	IBM Tivoli Monitoring, Omnibus, Impact
6. Implement incident, problem and asset mgmt processes	Tivoli Service Request Manager & TAMIT
7. Implement resiliency SLA	Tivoli System Automation
8. Implement backup approach	Tivoli Storage Mgr
9. Implement security functions	Tivoli Security / ISS portfolio
10. Implement cloud service specific billing metrics	TUAM
11. Implement rates for charging cloud service consumption	TUAM
12. Register cloud service to service catalog	TSAM / TSRM Service Provider Edition

Overview: Recommended management products, development tools and artifacts to be created on a per cloud service basis

Step	Building a cloud service hosted within the provider's premise		
	Products & technologies	Development/Config Tool	Cloud service specific configuration / artifact
1. Specify cloud service description	n/a (cross-cutting)	n/a	n/a
2. Implement runtime functionality of the cloud service	KVM, VMware, Linux, Windows, WAS, DB2, Cognos, Filenet, …	Rational Dev tools (for app artifacts), no image creation tools available today	VM golden master image, JEE ear file, SQL file, etc.
3. Define Unit of delivery & rating	n/a (cross-cutting decision)	n/a	n/a
4. Implement self-service delivery & management functionality	TSAM & TPM	TSAM SD app TPM ADPE	TSAM Service definition & TPM provisioning workflows
5. Implement monitoring metrics & event correlation rules	IBM Tivoli Monitoring, Omnibus, Impact	Universal Agent Builder (ITM) Omnibus-internal tooling	Monitoring agent (ITM), event correlation rule (Omnibus)
6. Implement incident, problem and asset mgmt processes	Tivoli Service Request Manager & TAMIT	TSRM-internal apps	Incident & problem management workflows, jobplans, escalations (TSRM)
7. Implement resiliency SLA	Tivoli System Automation	Text editor (no dedicated tooling)	HA policy
8. Implement backup approach	Tivoli Storage Mgr		TSM Backup agent Backup policy
9. Implement security functions	Tivoli Security & ISS Portfolio	Various	various
10. Implement cloud service specific billing metrics	TUAM	No dedicated tooling available	Metering collectors Job files
11. Implement rates for charging cloud service consumption	TUAM	UI and CSV file import	Rates persisted in TUAM database
12. Register cloud service to service catalog	TSAM / TSRM Service Provider Edition	TSAM/TSRM internal apps	Registration persisted in TSAM/TSRM internal database

*Cloud Computing Reference Architecture (CC RA) – Product & tooling mapping (on-prem products)

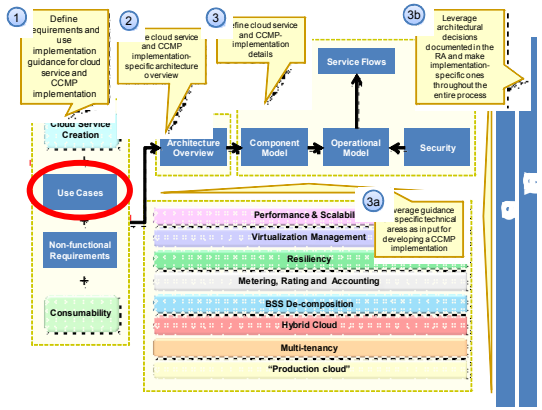


Cloud service creation – Applied patterns overview

- Walkthrough all 12 steps per applied pattern while describing implementation-specific decisions per step (incl. product selections)
  1. ISV
     - Example cloud service: “Desktop Cloud”
  1. “Enterprise”:
     1. WAS

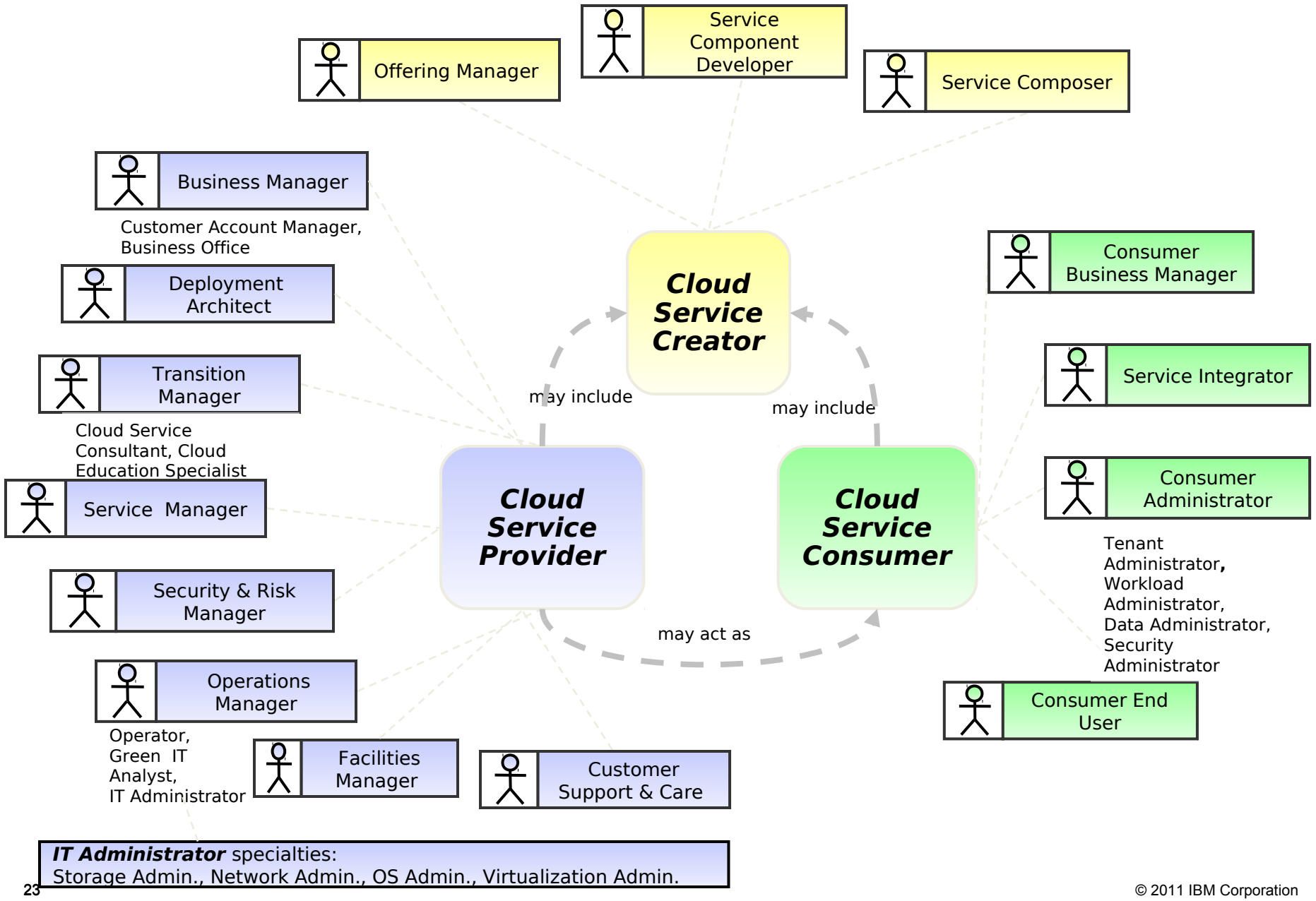
- The applied patterns should serve as a guideline for anyone who wants to deliver IT capabilities as cloud services.
- It is acknowledged that many real-world implementations won’t map exactly to the applied patterns described here – many will be in a “grey zone” in between.
  However, the applied patterns serve as good guidelines and blueprints for these custom implementations

- There will be more applied patterns available in future versions of the RA



Use Cases & Roles

Cloud Computing User Roles V2 – Applied as Actors of CC RA use cases



Detailed Role Description - Example: Service Manager

Main Goal	▪ Services meet their quality objectives ▪ Get up-to-date information on service adoption in seconds not days ▪ Have meaningful, fact based discussions on service quality with internal and external suppliers
Main Responsibility	The focus of a Service Manager is to enable the smoothest service flow possible between all systems, from business support system to operational support system services. They ensure that the running service is well aligned with business and operational objectives and targets.
Main Tasks of their day-to-day job	**Service Setup** ▪ Provision service **Service Runtime** ▪ Managing compliance with service's Service Level Agreements (SLA) ▪ Monitor third party supplier SLA ▪ Detect service faults ▪ Determine service impact (from infrastructure faults / degradation) ▪ Prioritise problems based on network impact ▪ Manage long and short term performance targets ▪ Ensure alignment of business and operational support systems
Skills	▪ Expert knowledge of the existing cloud infrastructure and good judgment of the impact of a new or changed solution on it ▪ Proficient understanding of the particular service offering and its service instances ▪ Expert automation knowledge ▪ Proficient understanding of capacity/performance issues within and across systems
Working with which other roles ?	Operations Manager; Operator; Network Administrator; Customer Account Manager; Business Manager; Green IT Analyst; Deployment Architect; Cloud Service Creator

CC RA V2.0 Use Cases

- 38 use cases, grouped into 10 packages, along the cloud service life cycle

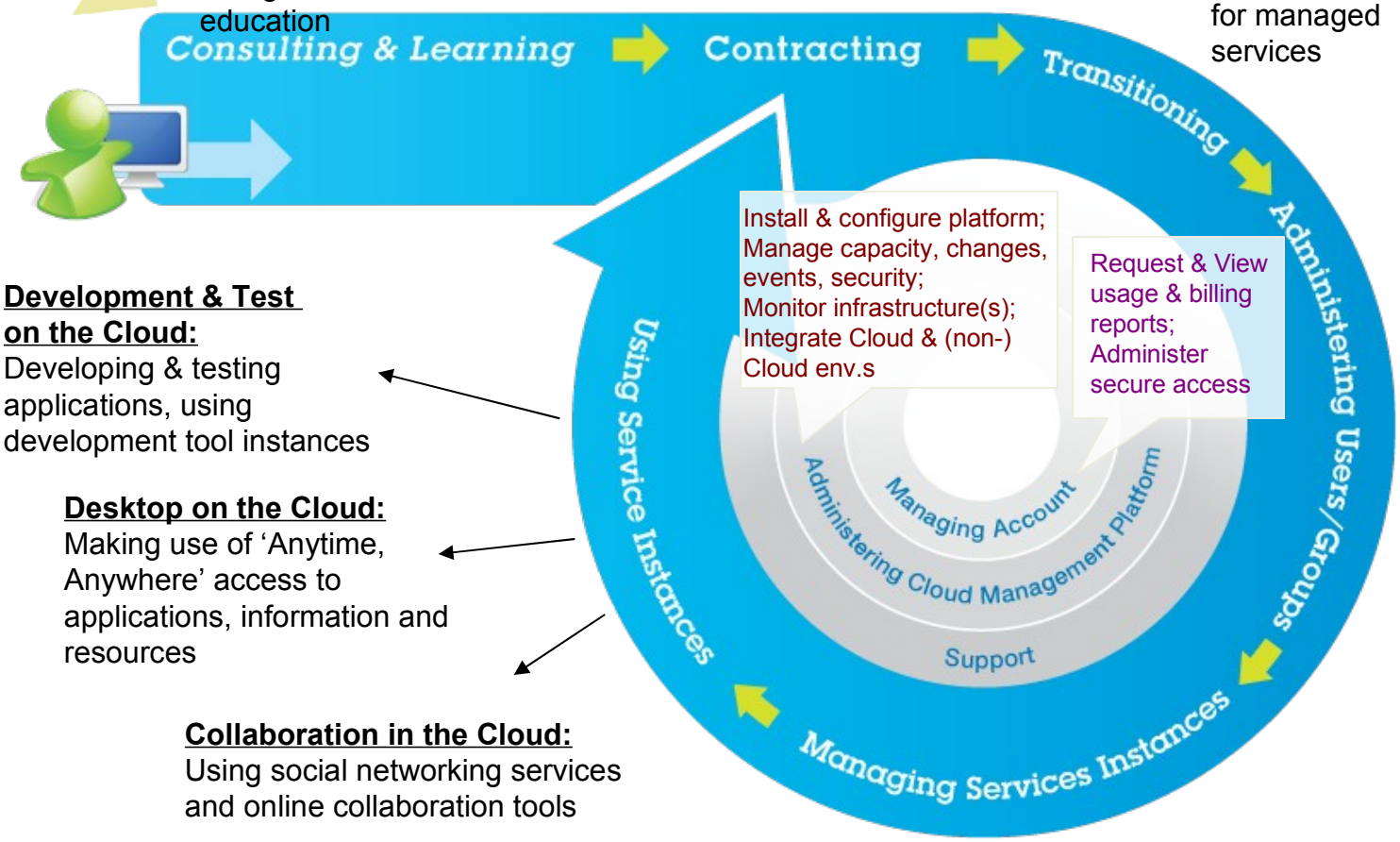
Creating & Managing Service Offerings

Defining the Scope, Creating & Managing Service Offerings, Creating & Managing Business Aspects

Consultative Analysis; Self-guided information & education

Via F2F or Web Sales channel

Enable customers for managed services



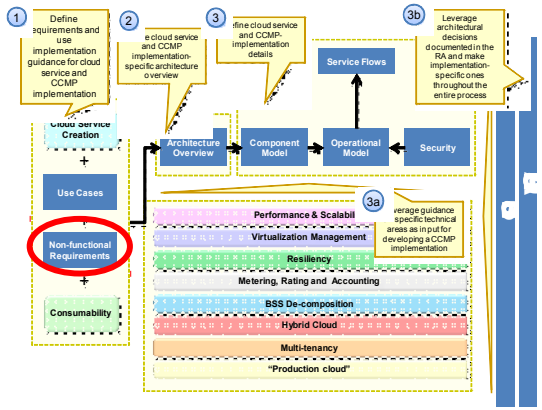
Development & Test on the Cloud:
Developing & testing applications, using development tool instances

Desktop on the Cloud:
Making use of 'Anytime, Anywhere' access to applications, information and resources

Collaboration in the Cloud:
Using social networking services and online collaboration tools

Create, modify, delete users; Create, modify, delete user groups

Request, view, update, delete service instances

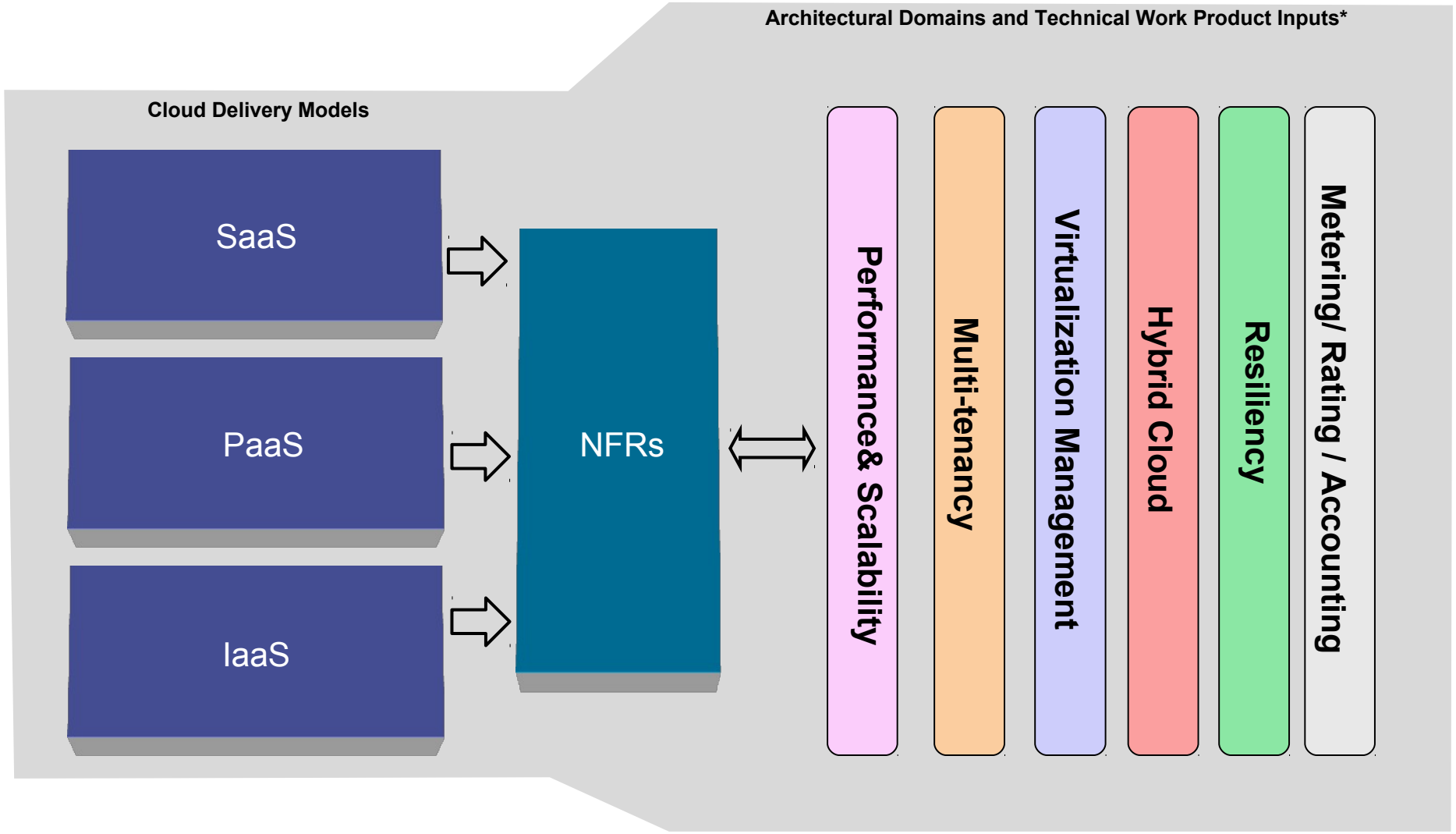


Non-functional requirements

Non-functional requirements TWP: Scope & Purpose

- Consolidate typical Cloud Computing Management Platform Non-Functional Requirements to illustrate the minimal set of non-functional requirements for building a Cloud Management Platform
- Reference architecture NFRs can help practitioners collect requirements during any project requirements gathering phase
- NFRs are key input to other TWPs including Component Model and Operation Model
- NFRs are high level in nature, as such, they should be evaluated along with the lower level CC RA Operational Model to establish the optimum Infrastructure for deploying the proposed solution
- Non-Functional requirements of a Cloud management platform are quality requirements or constraints of the platform that must be satisfied. These requirements address major operational and functional areas of the platform in order to ensure the robustness of the system.

NFR Context Relationship with other TWP and Domains

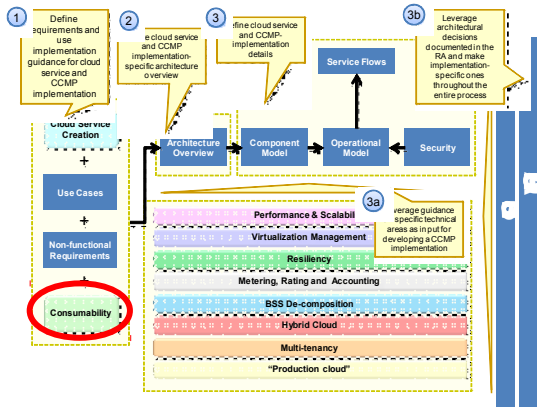


Non-Functional Requirement TWP document contains the following detailed NFRs…

In version 2 of the NFR document we added NFRs from the following areas on top of the version 1:

- Applied Pattern NFRs Added
  - IaaS - Public Cloud: +20
  - IaaS - Private Cloud: +17
  - PaaS: +15
  - SaaS – Managed Service Desk: +16

- Domain Specific NFRs Added
  - Performance and scalability: +14
  - Multi-tenancy: +16
  - Virtualization management:
    - Base virtualization: +12
    - Network virtualization: +18
  - Hybrid Cloud: +10 NFRs
  - Resiliency: +13
  - Metering / Accounting: +24



Consumability

What makes a cloud solution consumable?

Consumability is a customer-centric term that describes the total customer experience with IBM products, solutions and services. Consumable solutions:

- Reduce total cost of ownership
- Improve time to value
- Reduce support and maintenance costs

Key characteristics

On-demand self-service

Ubiquitous network access

Location independent resource pooling

Rapid elasticity

Flexible pricing models

Integration is critical

Data Integration / Migration

Process Integration

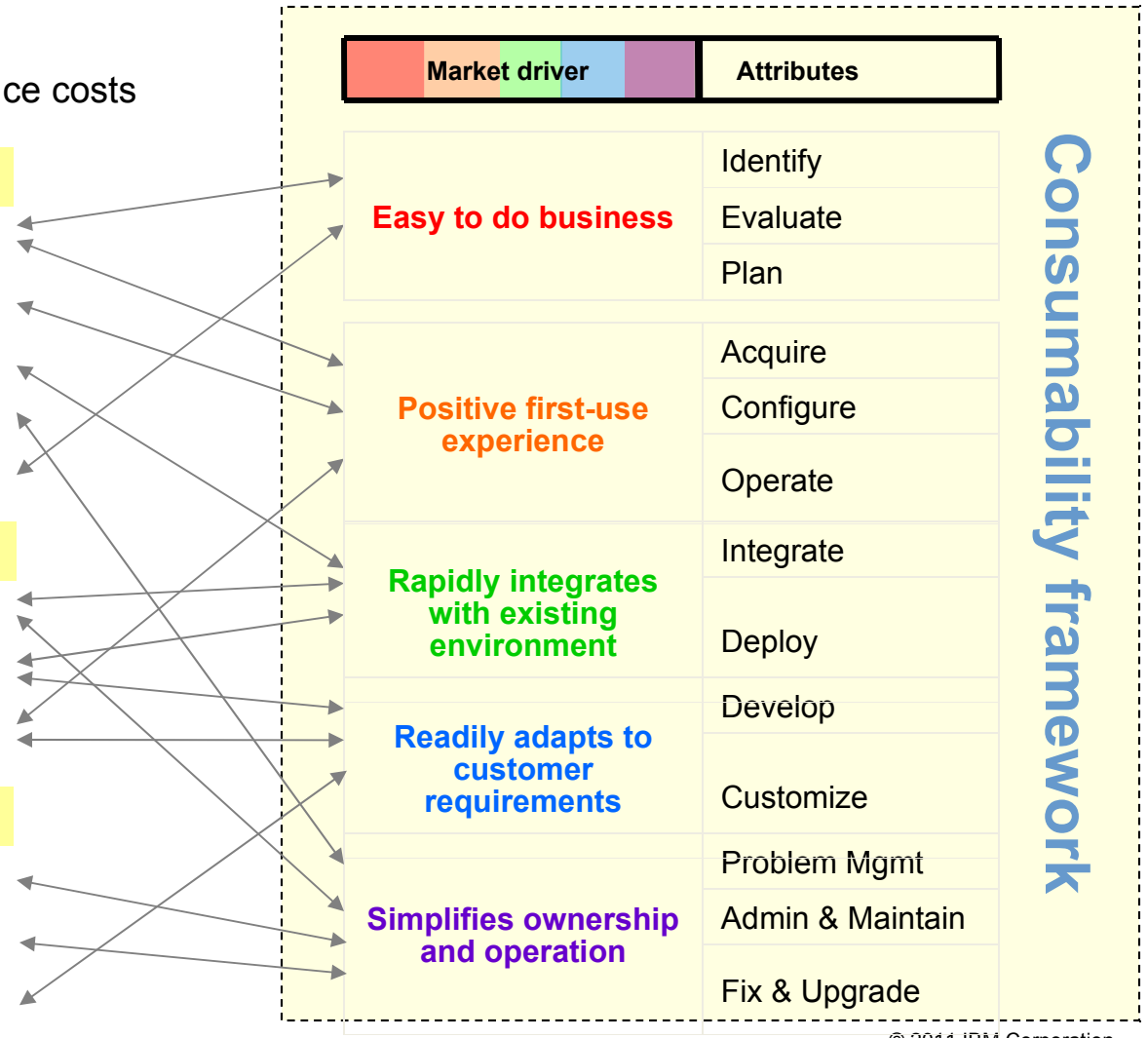
UI Mashup

Labor simplification drives efficiency

Self Service

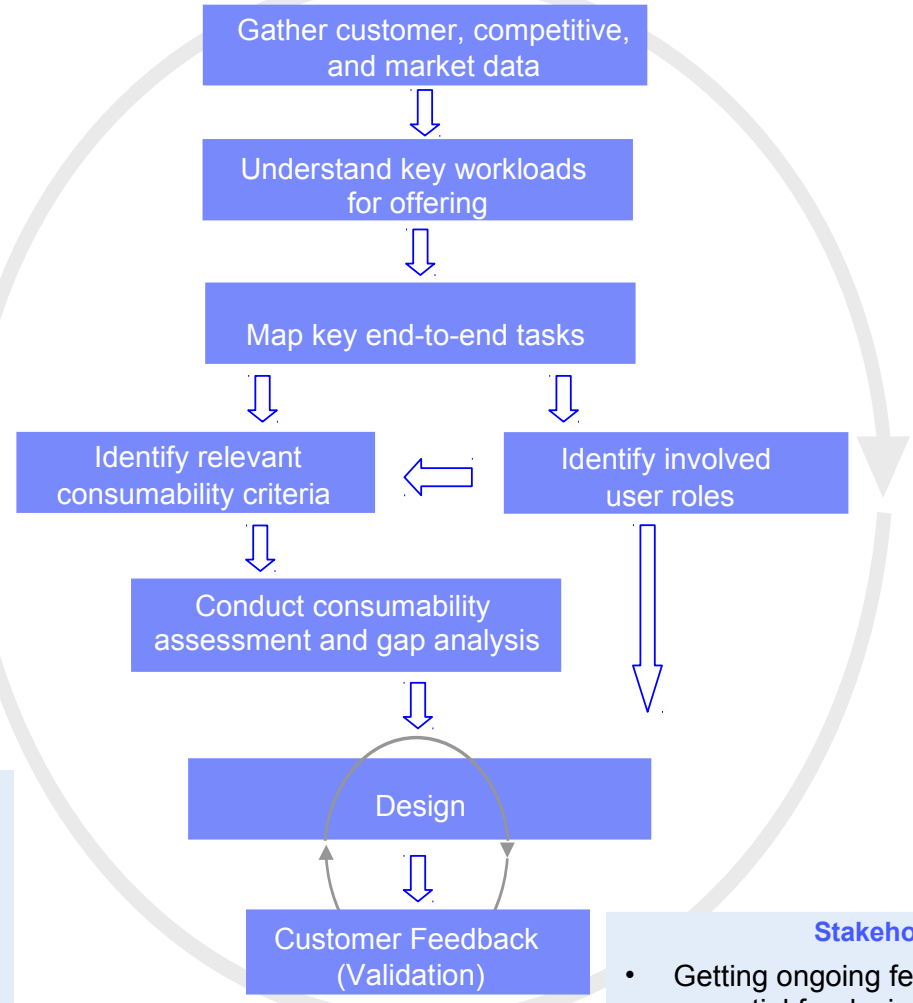
Automation of Management

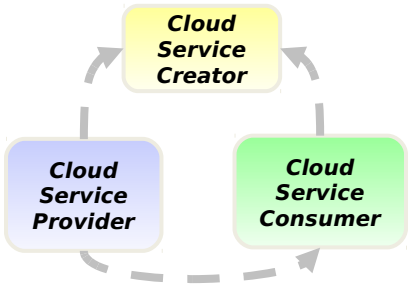
Standardization of Workloads



Designing Consumable Cloud Solutions

Consumability Assessment
Using the consumability framework that is modified specifically for cloud solutions, identify the consumability goals for a particular solution, assess the solution, and work towards closing gaps.

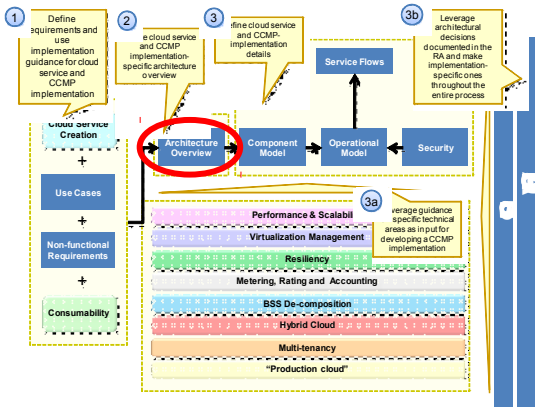




Cloud User Roles
Describing who is interacting with a cloud service at the different stages of the cloud life cycle

Stakeholder Feedback

- Getting ongoing feedback from stakeholders is essential for designing a consumable solution.
- Understanding the user roles, the next step is to identify key stakeholders whose feedback you need to gather



Architectural Principles & Overview

Cloud Computing Reference Architecture: Architectural Principles

An architectural principle is an overarching guideline or paradigm driving architectural decisions across the entire architecture process on a more granular level.

1. Design for Cloud-Scale Efficiencies ("Efficiency Principle"):

Design for cloud-scale efficiencies, and time-to-deliver/time-to-change metrics, when realizing cloud characteristics such as elasticity, self-service access, and flexible sourcing.

→ Overarching objective of Driving down costs (¢/ServiceInstanceHour) and time-to-response by orders of magnitude

2. Support Lean Service Management ("Lightweight Principle"):

Support lean and lightweight service management policies, processes, and technologies.

→ Radical exploitation of high degree of standardization in cloud environments to reduce management costs, based on an Eliminate-Standardize-Optimize approach

3. Identify and Leverage Commonalities ("Economies-of-scale Principle"):

Identify and leverage commonality in cloud service design.

→ Maximum sharing of mgmt components, infrastructure & infrastructure / platform cloud services across cloud services to reduce CapEx & OpEx and time-to-market

4. Define and Manage Cloud Services generically along their Lifecycle ("*Genericity Principle*"):

Define service templates and manage service instances generically along their lifecycle, across I/P/S/BPaaS.

→ Support I/P/S/BPaaS cloud services in a generic fashion, with a single management platform

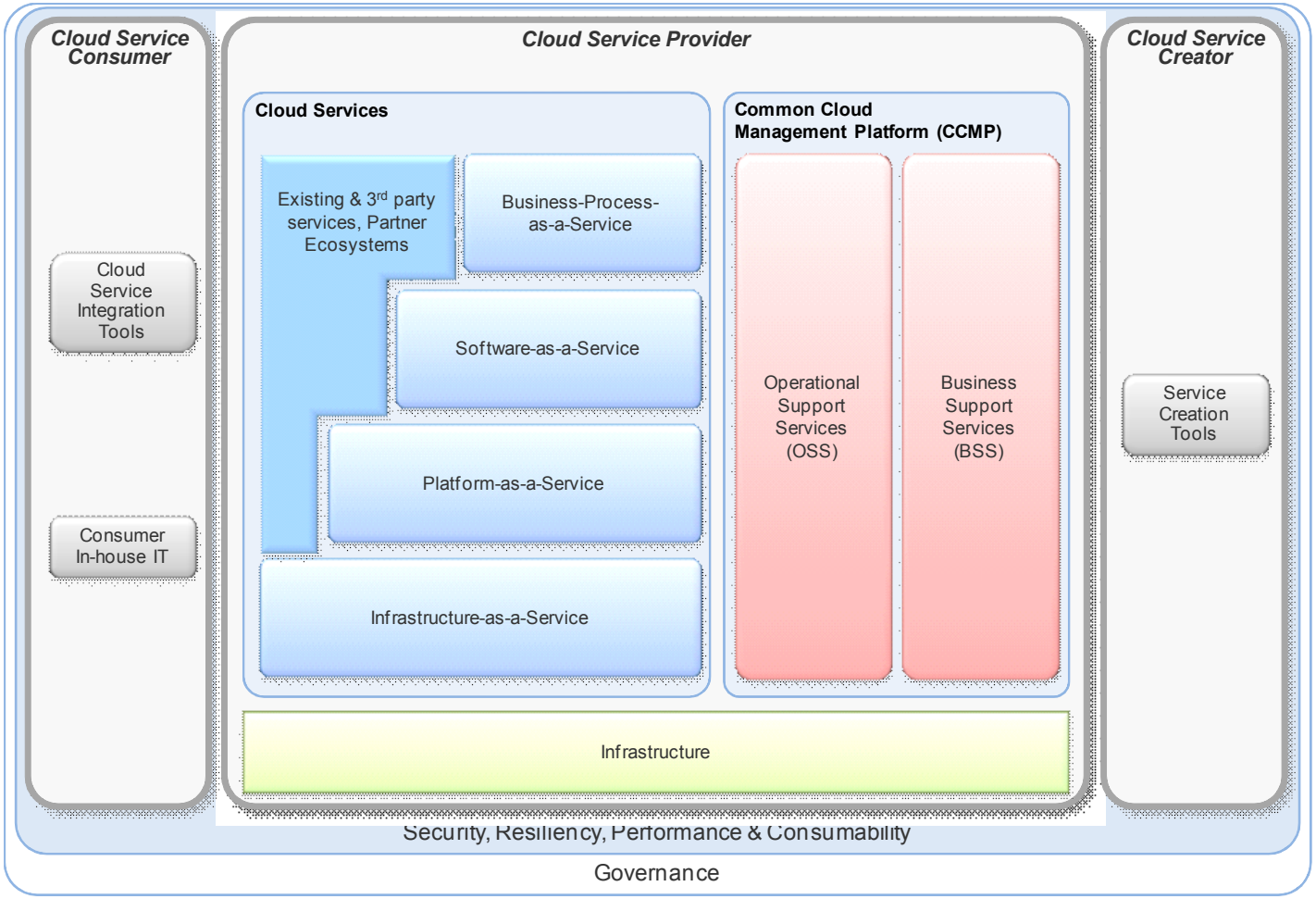
Cloud Computing Reference Architecture (CC RA) – Overview

CC RA architecture overview diagram lays defines basic elements of any cloud service environment

- Three main roles: Cloud service consumer, provider and creator whereas multiple roles can be fulfilled by the same organization/person
- Offered cloud services, required Management services (CCMP) and underlying infrastructure are defined.
- Next level of drill-down defined for each architectural element

CCMP is one element of the overall cloud computing reference architecture

- Defines operational and business support services *commonly* needed for delivering and managing any cloud service (I/P/S/BPaaS)
- Doesn't imply a single, monolithic implementation of a CCMP. CCMP is typically implemented by a well-integrated set of mgmt products



Cloud service model definitions

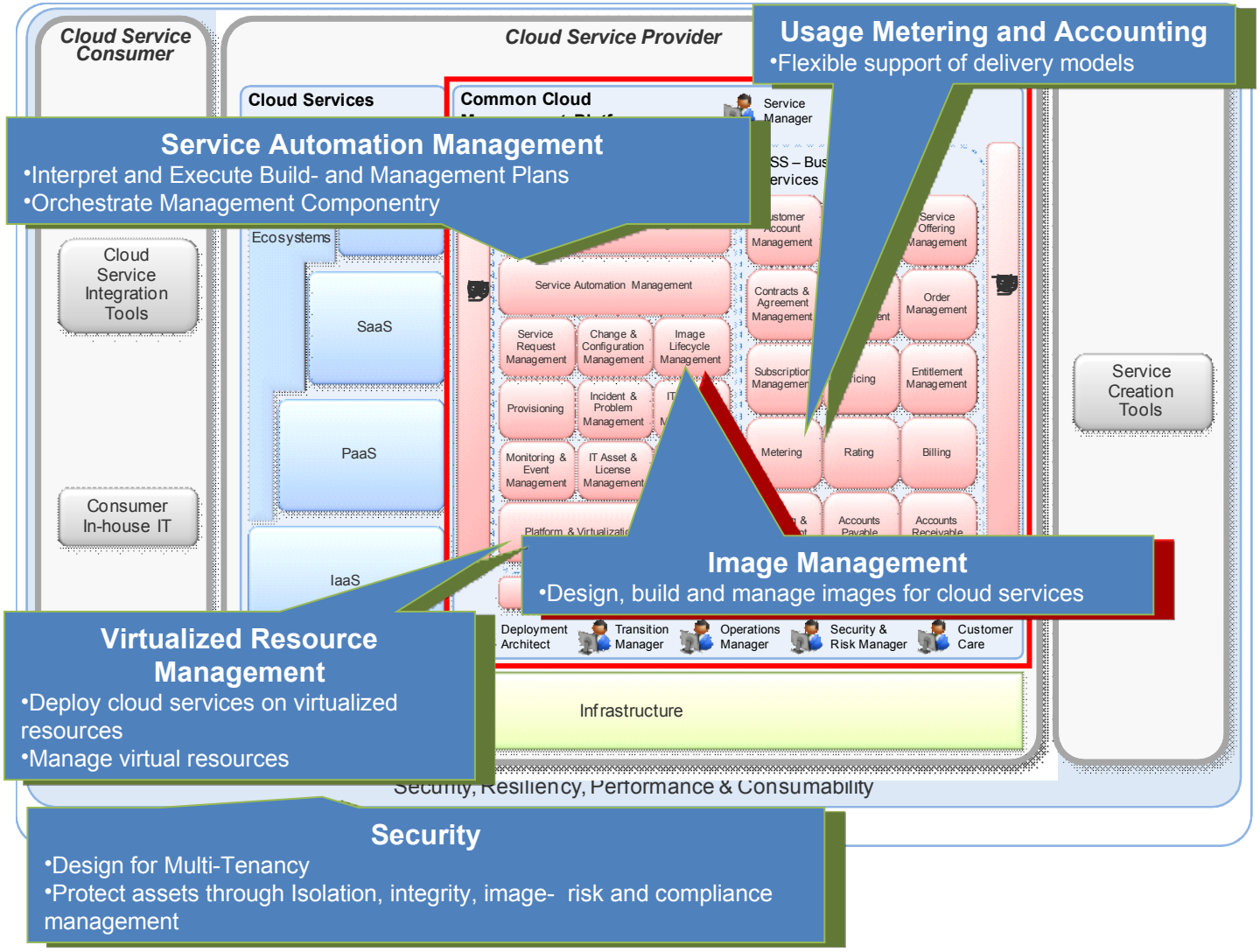
Term	Definition	Source
BPaaS	Business process services are any business process (horizontal or vertical) delivered through the Cloud service model (Multi-tenant, self-service provisioning, elastic scaling and usage metering or pricing) via the Internet with access via Web-centric interfaces and exploiting Web-oriented cloud architecture. The BPaaS provider is responsible for the related business function(s).	IBM[2]
SaaS	The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure and accessible from various client devices through a thin client interface such as a Web browser (e.g., web-based email). The consumer does not manage or control the underlying cloud infrastructure, network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.	NIST[1]
PaaS	The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created applications using programming languages and tools supported by the provider (e.g., java, python, .Net). The consumer does not manage or control the underlying cloud infrastructure, network, servers, operating systems, or storage, but the consumer has control over the deployed applications and possibly application hosting environment configurations.	NIST[1]
IaaS	The capability provided to the consumer is to rent processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly select networking components (e.g., firewalls, load balancers).	NIST[1]

Note: Across all cloud service models the definition is determined by the management scope covered by the provider. For example, in IaaS "*the consumer does not manage or control the underlying cloud infrastructure […]", in PaaS "the consumer does not manage or control the underlying cloud infrastructure, network, servers, operating systems, or storage […]", etc.*. ***So this essentially about the tasks the operations staff of the provider takes on, it is not about the virtualization technology being used.*** *For example, it's possible to use hypervisor-level virtualization to realize PaaS, SaaS or BPaaS.*

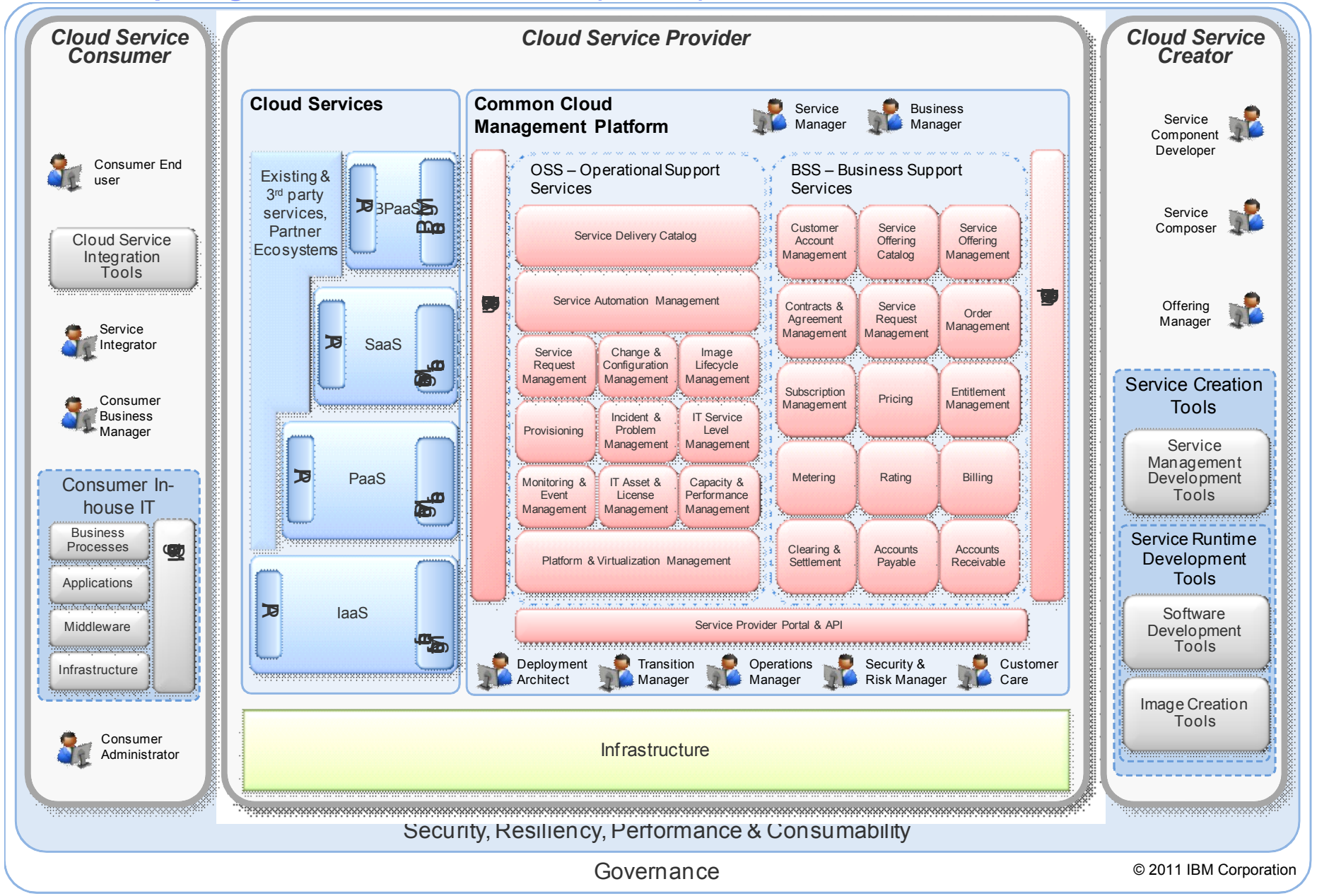
1. National Institute of Standards and Technologies; Draft NIST Working Definition of Cloud Computing, May 14, 2009
2. IBM MI and IPR definition bridge between Gartner and IDC, Aug 19, 2010

Cloud Computing Reference Architecture Overview Diagram – CCMP drill-down, highlighting some important topics

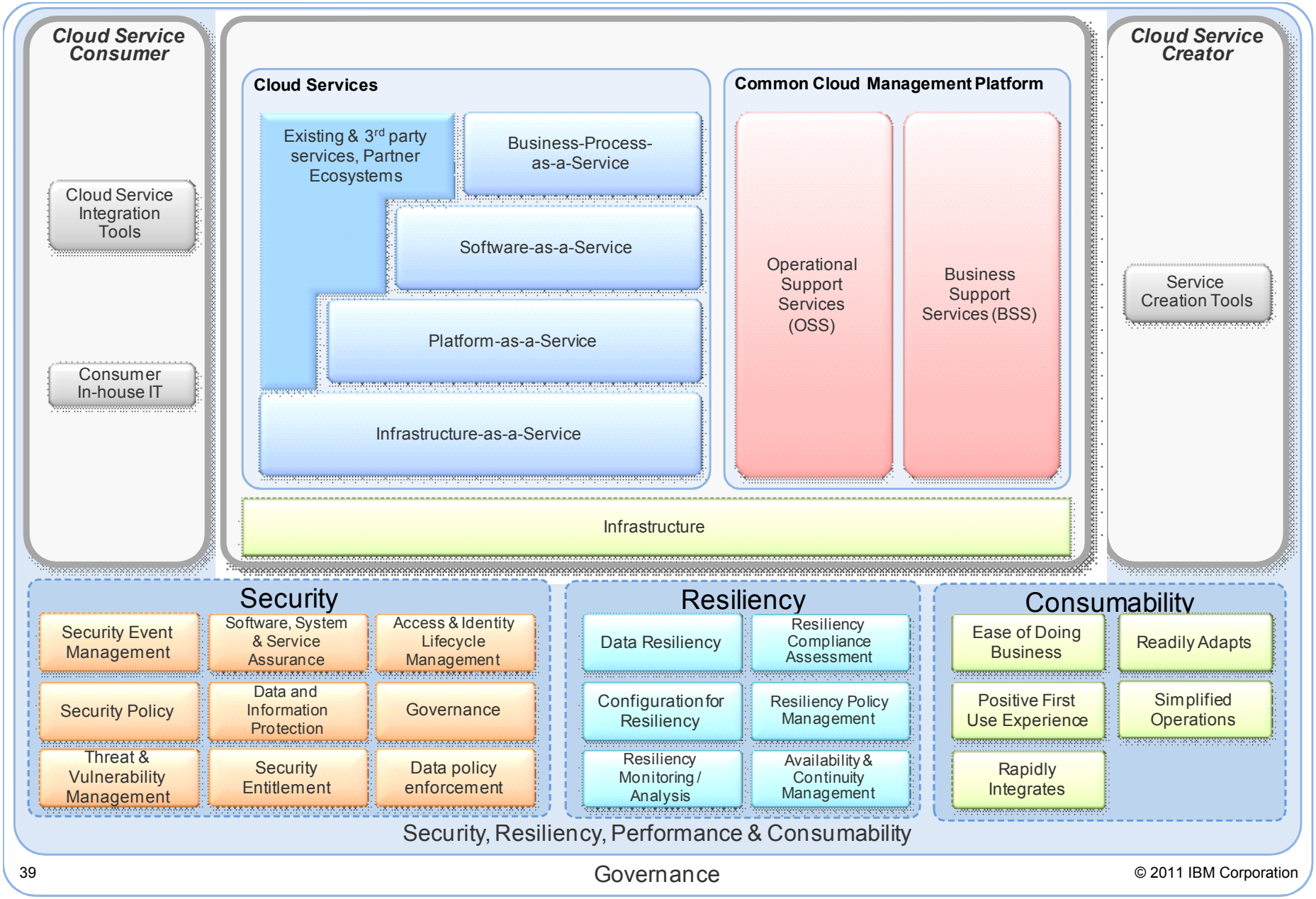
- CCMP defines all management functions commonly needed for the automated delivery & management of any cloud service
- Components are grouped in two major categories: Operational Support Services (OSS) and Business Support Services (BSS)
  - OSS: Responsible for managing the runtime components of cloud services
  - BSS: Responsible for all business / finance related aspects of cloud services
- Economies of scale can be achieved by managing multiple cloud services with the same set of mgmt components (see architectural principles)



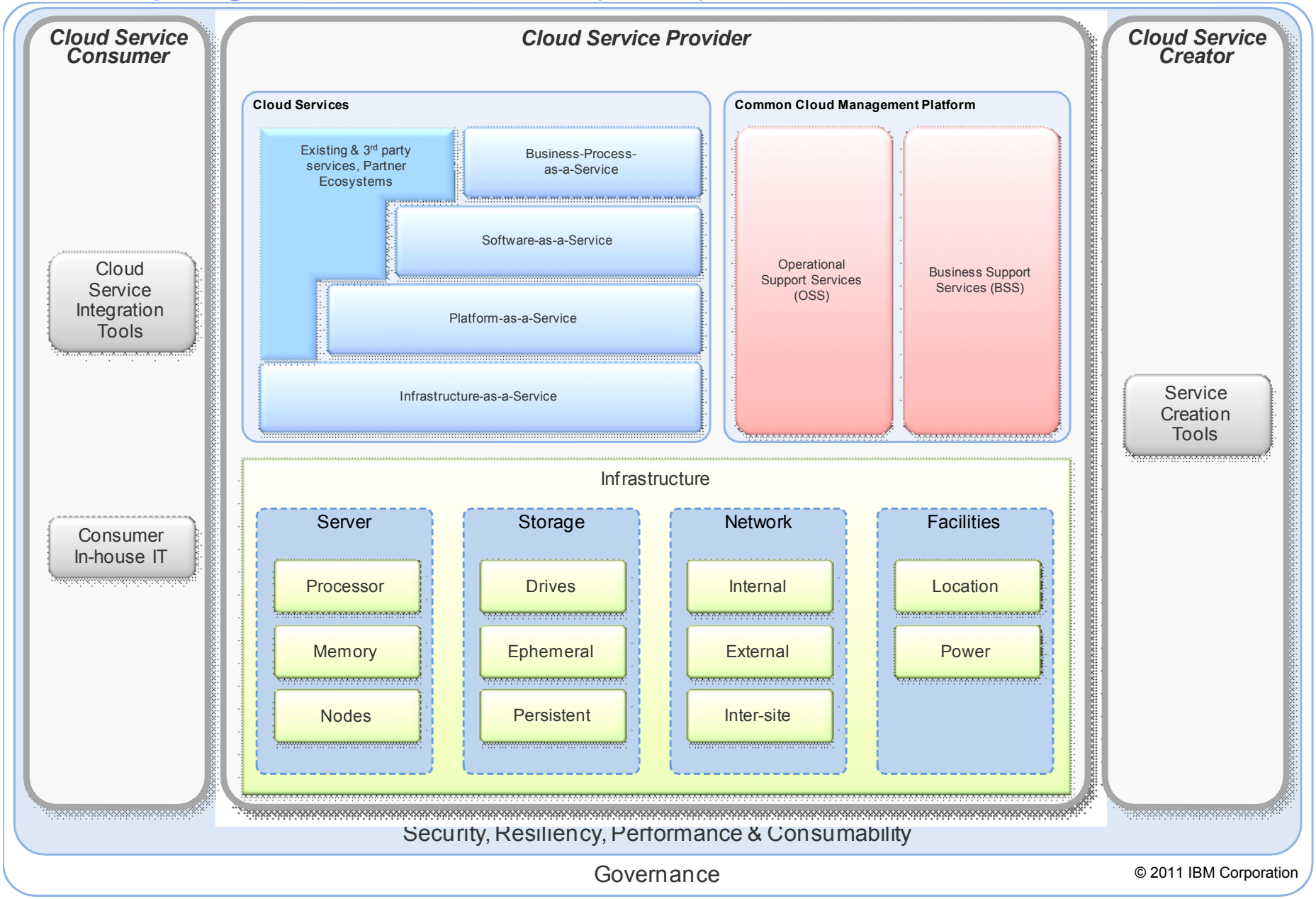
Cloud Computing Reference Architecture (CC RA) – Overall drill-down



Cloud Computing Reference Architecture (CC RA) – Security, Resiliency, Performance & Consumability drill-down

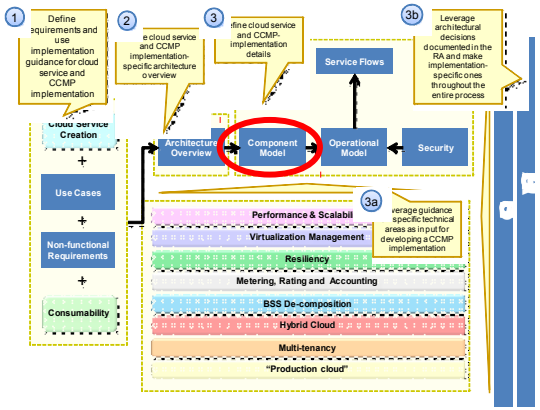


Cloud Computing Reference Architecture (CC RA) – Infrastructure drill-down



Architecture Overview: Applied patterns

- Applied Patterns are based on cloud projects where the reference architecture AOD has been used to create cloud implementations.
  Applied patterns reflect common scenarios of how a cloud could be implemented:
  1. **IBM Smart Business Development and Test on the IBM Cloud** is a sample for a "simple" (public) service provider architecture pattern. The diagram and related descriptions have been used to lead the high-level discussion and to guide decisions on scope for release 1 and following.
  2. The RA overview diagram has be used to discuss roles and responsibilities for a **Desktop cloud** project where part of the infrastructure is managed by IBM.
  3. An architecture overview diagram pattern for a **private enterprise where traditional IT is managed in parallel to a private cloud environment**, plus the consumption of externally provided public cloud services
  4. **A service provider "whitelabeling" architecture overview diagram pattern** (i.e. a service provider consuming cloud services from another cloud service provider and reselling the same offering -- with different branding and different prices)
  5. A **service provider consumption and value-add architecture overview diagram** pattern (i.e. service provider consuming 1..n cloud services from other cloud services, composing them and adding service provider specific value-add functionality on top)
  6. **Bio-Informatics Exchange**
  7. **IBM Cloud Service Provider Platform**
  8. **IBM Shared Private Production Cloud**

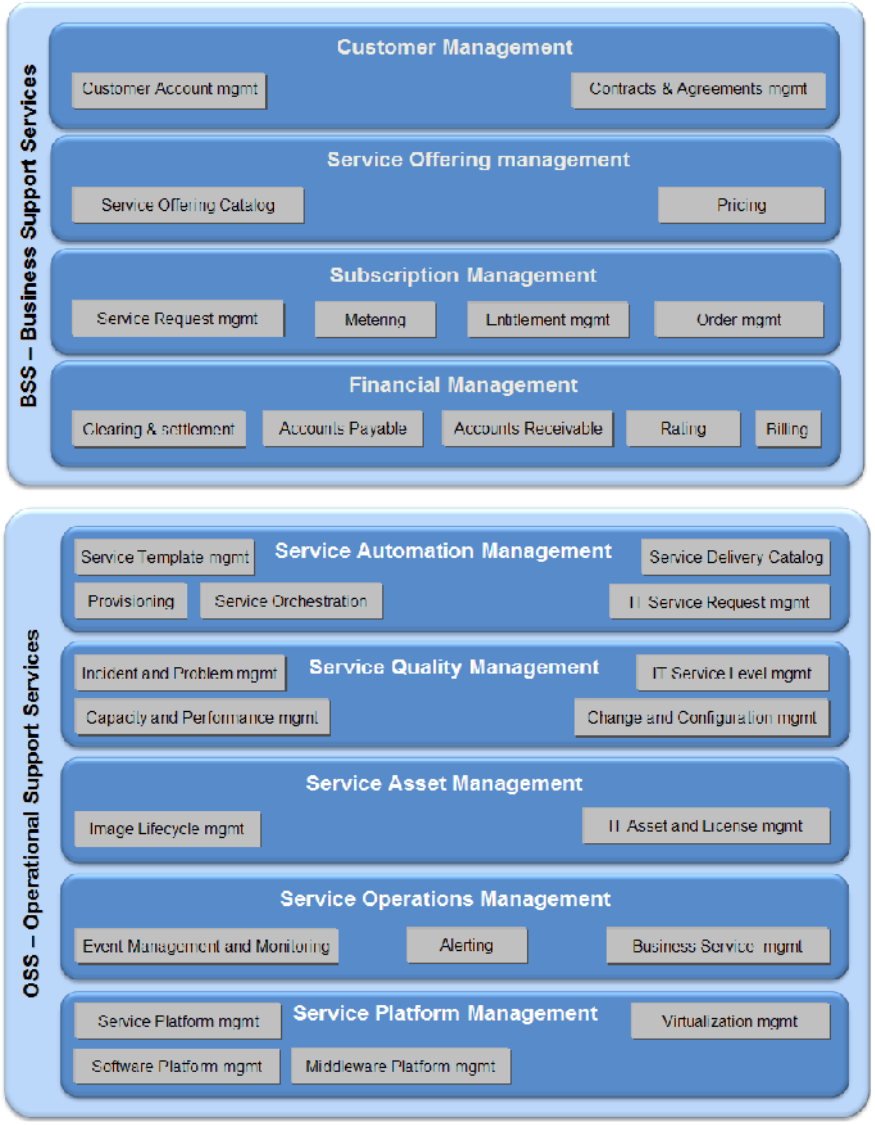


Component Model

CC RA 2.0: Component Model

Primary focus on BSS and OSS

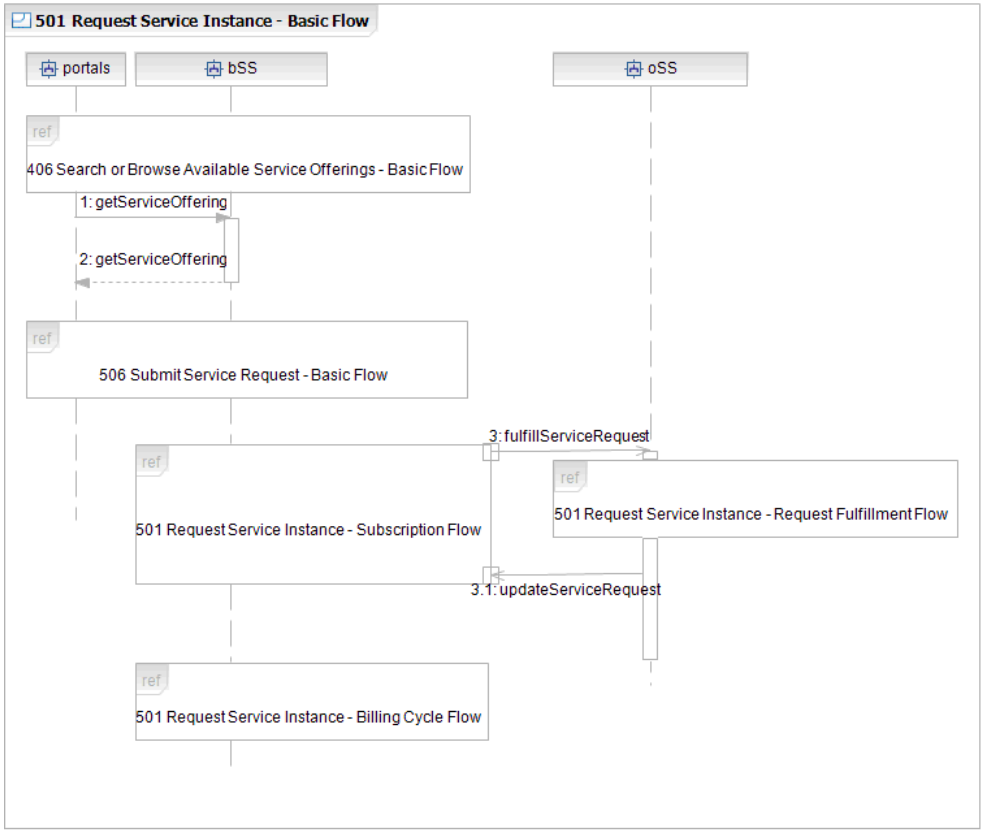
- BSS and OSS component specifications and top-level decomposition
- Use Case realizations of important use cases (e.g. Requesting Service Instances)
- Extensive UML-based modeling of BSS and OSS component relationships and dependencies
- Cross-component view on CC RA 2.0 domains
- Mapping of current IBM cloud implementations to CC RA 2.0 Component Model
- Condensed mapping of CC RA 2.0 components with IBM recommended tools



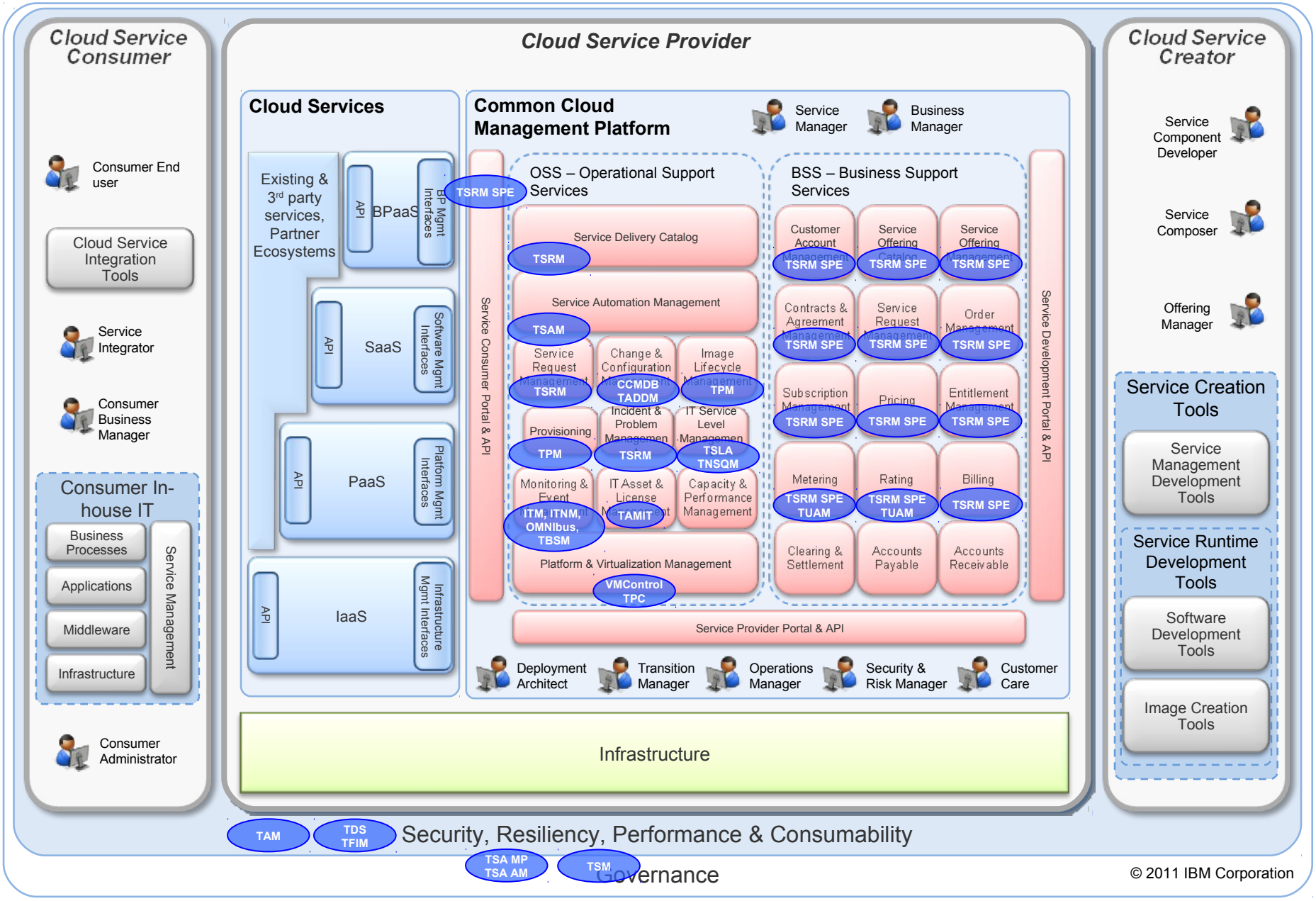
Use Case Realizations

A use-case realization describes how a particular use case is realized within the design model, in terms of collaborating objects (IRUP).

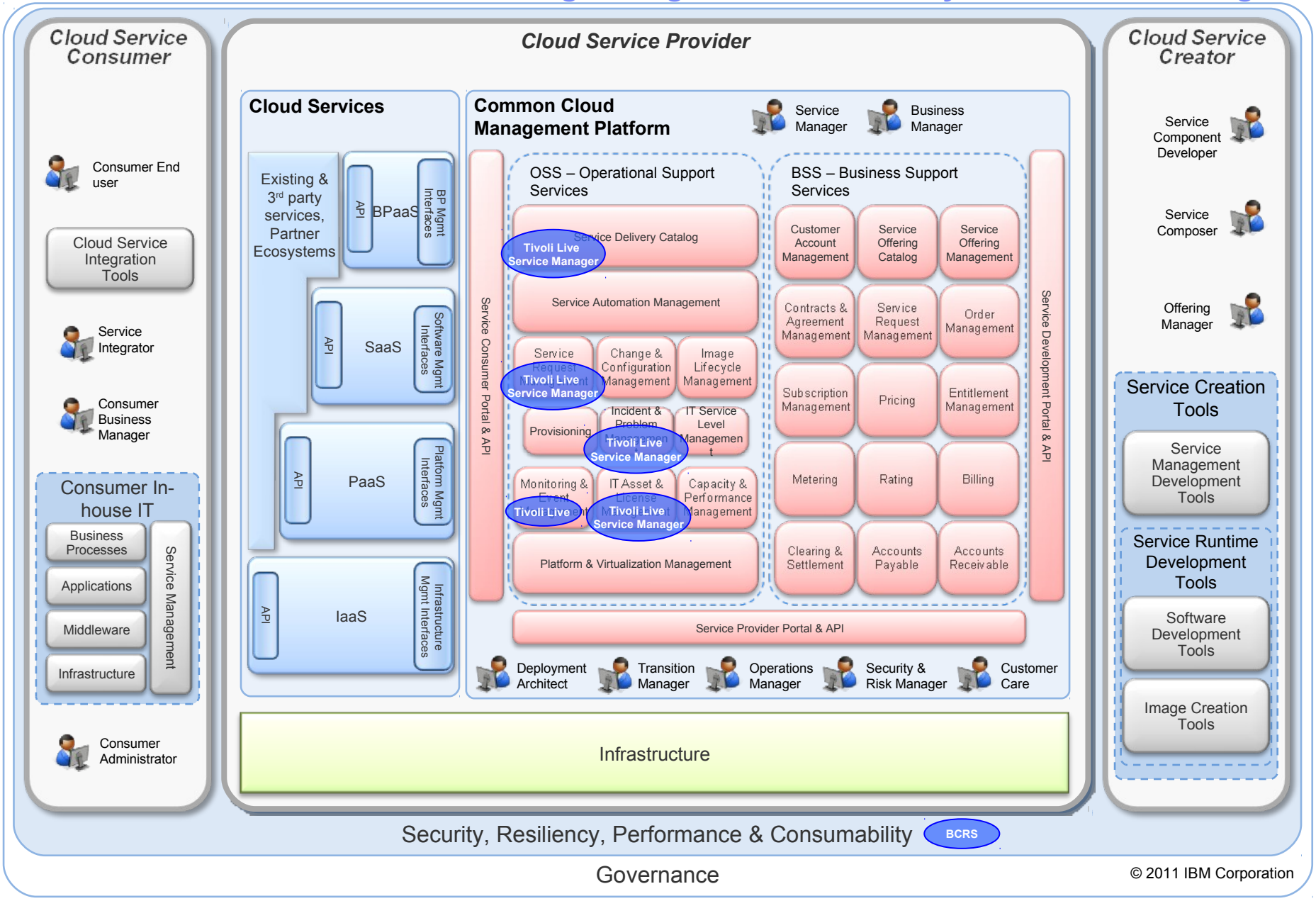
- CC RA 2.0 CM describes realization of typical use cases:
  - Creating Service Offering incl.
    - Create Service Template
    - Create Rating Scheme
  - Requesting Service Instance
- Sequence diagrams help to understand components, processes and flows involved in realization of use cases
- Detailed component diagrams are included to provide information about component interfaces, relationships and dependencies

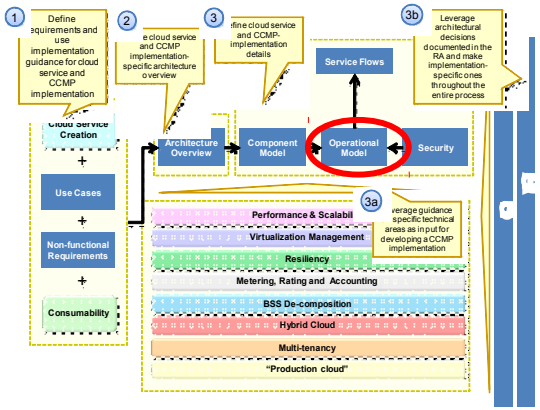


IBM recommended products for consuming management functionality as on-premise installations



IBM recommended tools for consuming management functionality as a hosted offering



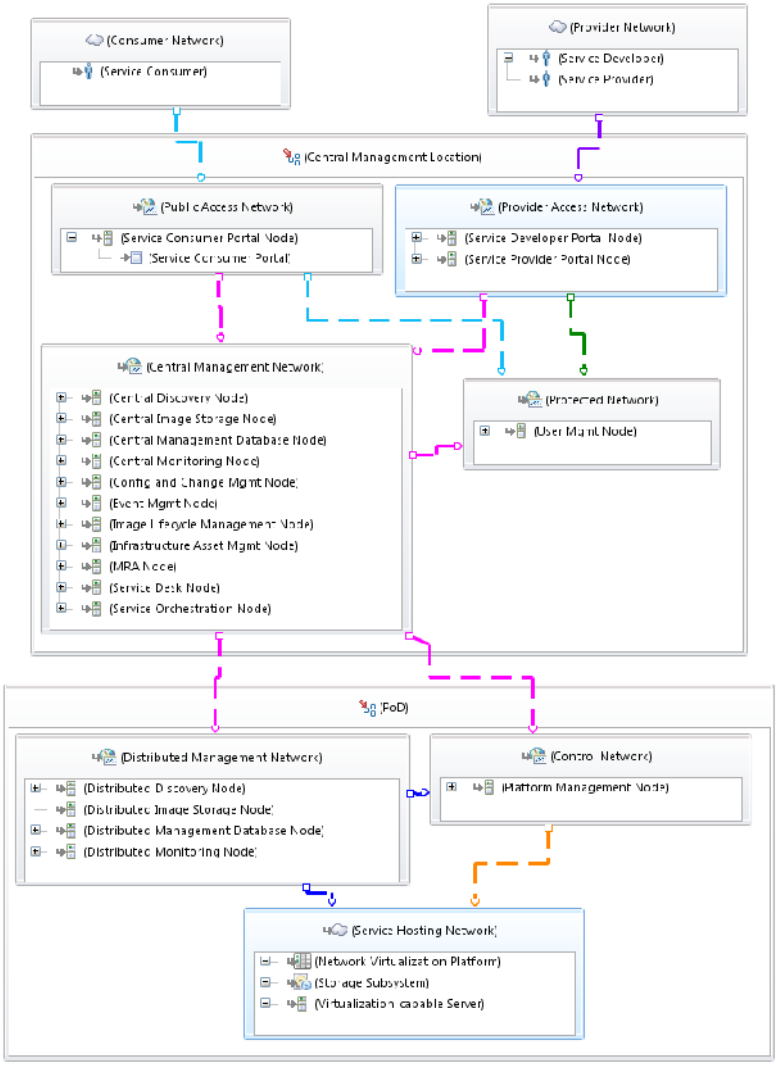


Operational Model

CC RA 2.0: Operational Model Updates

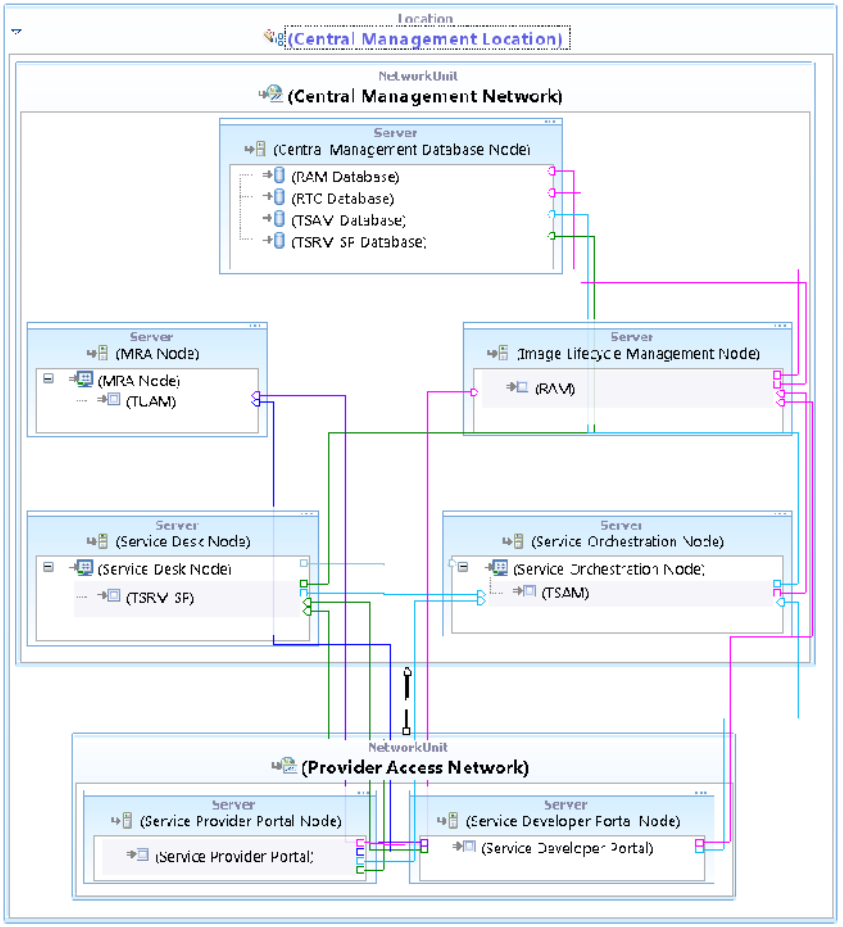
Primary focus on leveraging existing cloud implementations in IBM and other CC RA work products

- Show how components are to be realized and deployed in a cloud-specific fashion
- Harvest architectural patterns from existing cloud implementations (e.g. SBDTC)
- RSA models for each level of elaboration which can be leveraged by adopters
- System walkthroughs for major functional areas in Use Case Model and Component Model
- Implementation guidance on realizing CC RA using IBM recommended tools identified in Component Model
- Identify critical decision points and requirements used to guide the implementation



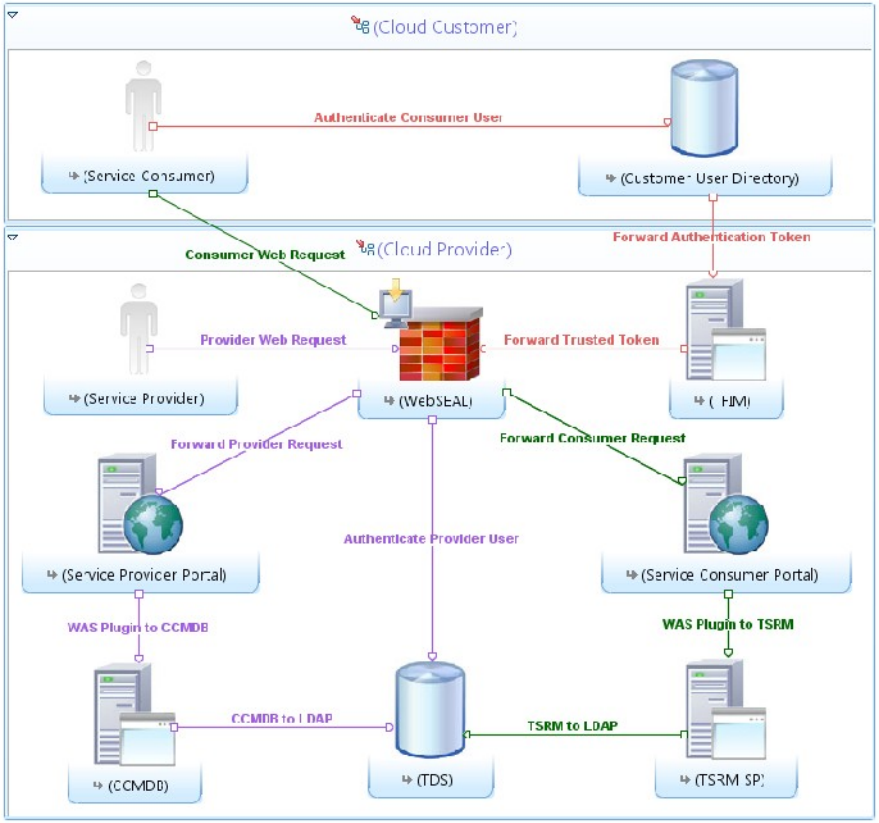
The Operational Model emphasizes how a Cloud deployment differs

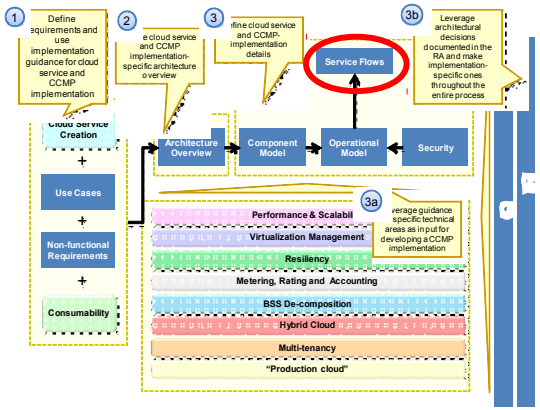
- Adopter guidance on key decision points related to Cloud
- Addresses each node and component realization specifically, noting important considerations specific to Cloud
- Walkthroughs show how the component operates in the context of the CCMP



Operational Model includes guidance for adopters to enable application to engagements

- Implementation guidance includes:
  - Applied patterns showing how CC RA can be adapted to engagement-specific requirements
  - Recommended tool selection aligned with component mappings in the Component Model
  - Identification of architectural decisions and considerations of which practitioners need to be aware





Service Flows

CC RA –Service Flows TWP: Scope & Purpose

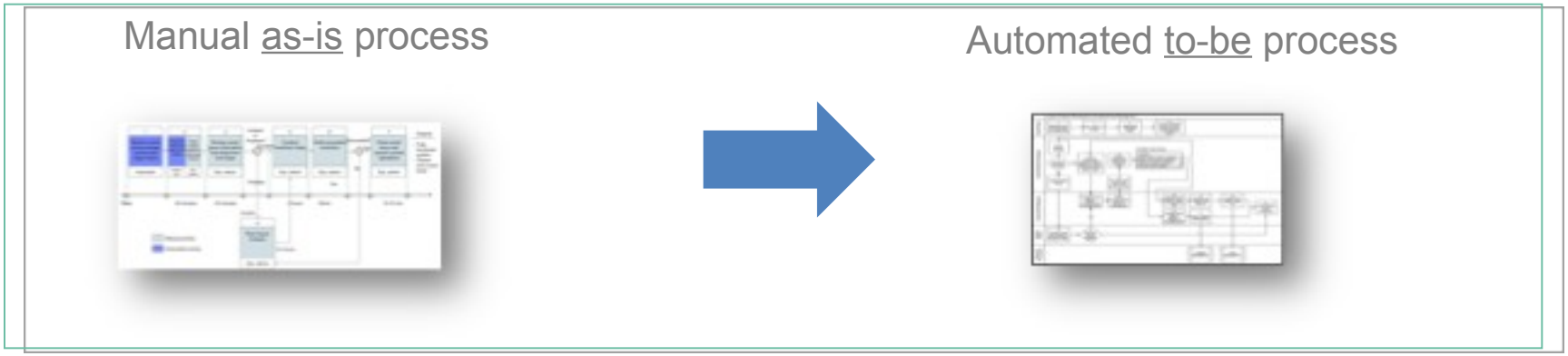
- Defines all operational processes (“service flows”) required for managing cloud services based on a CCMP deployment.
  - These service flows are focused on reducing labor costs for management to a minimum, by building on high degree of standardization present in any cloud environment.

- Service flows depend on service management components as defined in the CC RA component model – CCMP components are required for automating as many tasks as possible.

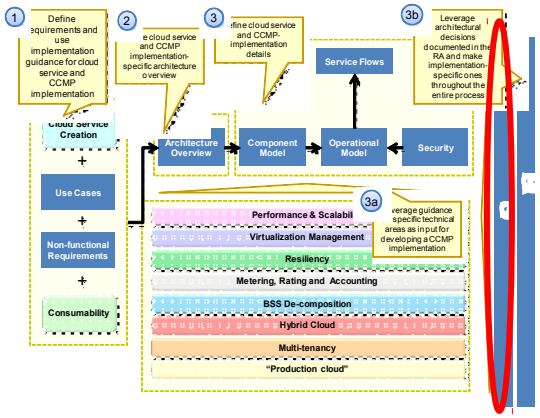
CC RA Service Flows Content: Overview

Service flows processes differences from standard IT management processes

- Leverages reduced (eliminate) scope of management, standardization, and opportunity for optimization to dramatically reduce labor costs.
- Aimed at delivering cloud management processes for cost-competitive cloud infrastructures



- Implementation of cloud-optimized service management processes
  - Configuration and Asset Mgmt
  - Patch / Provisioning / Image Mgmt
  - Incident / Problem Mgmt, Monitoring
  - Performance and Capacity Mgmt
  - Service Level Management / Metering
  - Service Request Management
  - Continuity Mgmt



Standards

Context and Scope of CC RA 2.0 Standards

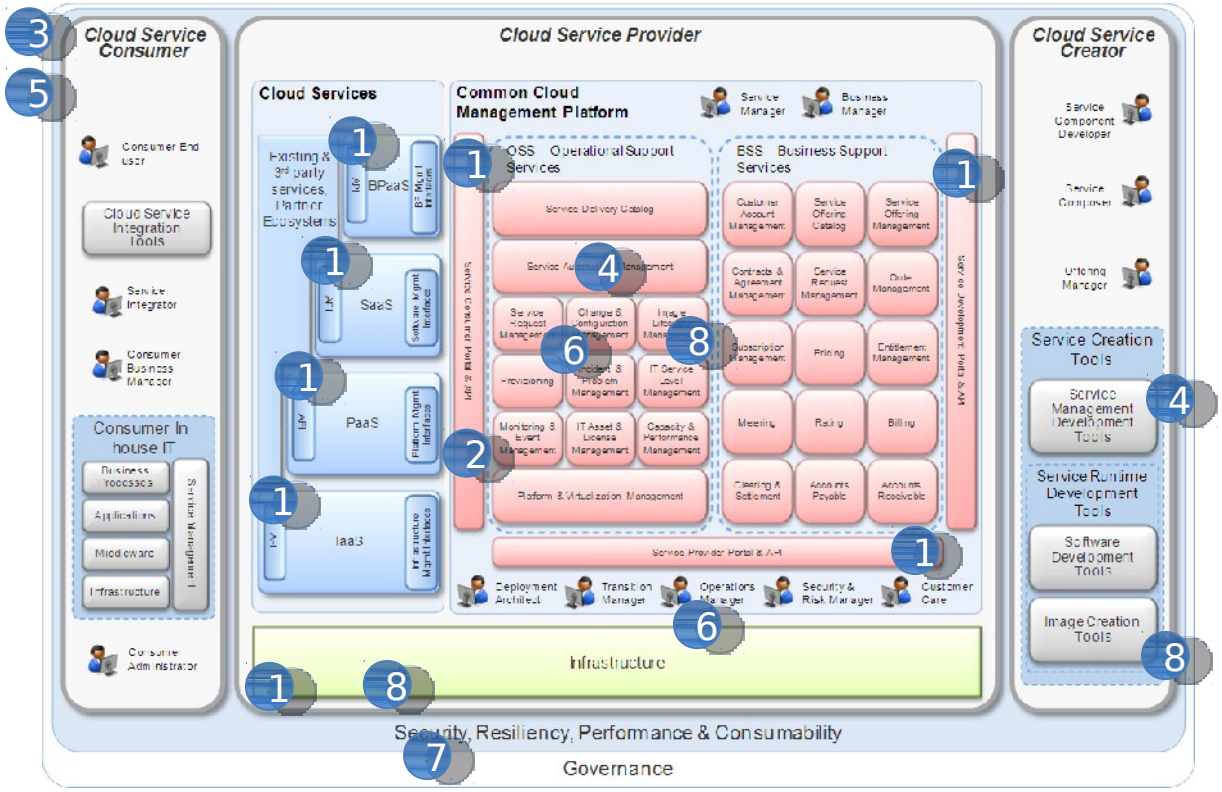
Context

By means of adhering to and driving the evolution of standards we **ease adoption** of our platform, assure **solution portability**, **avoid vendor lock-in**, and enable an **eco-system of partners** around our platform.

Scope

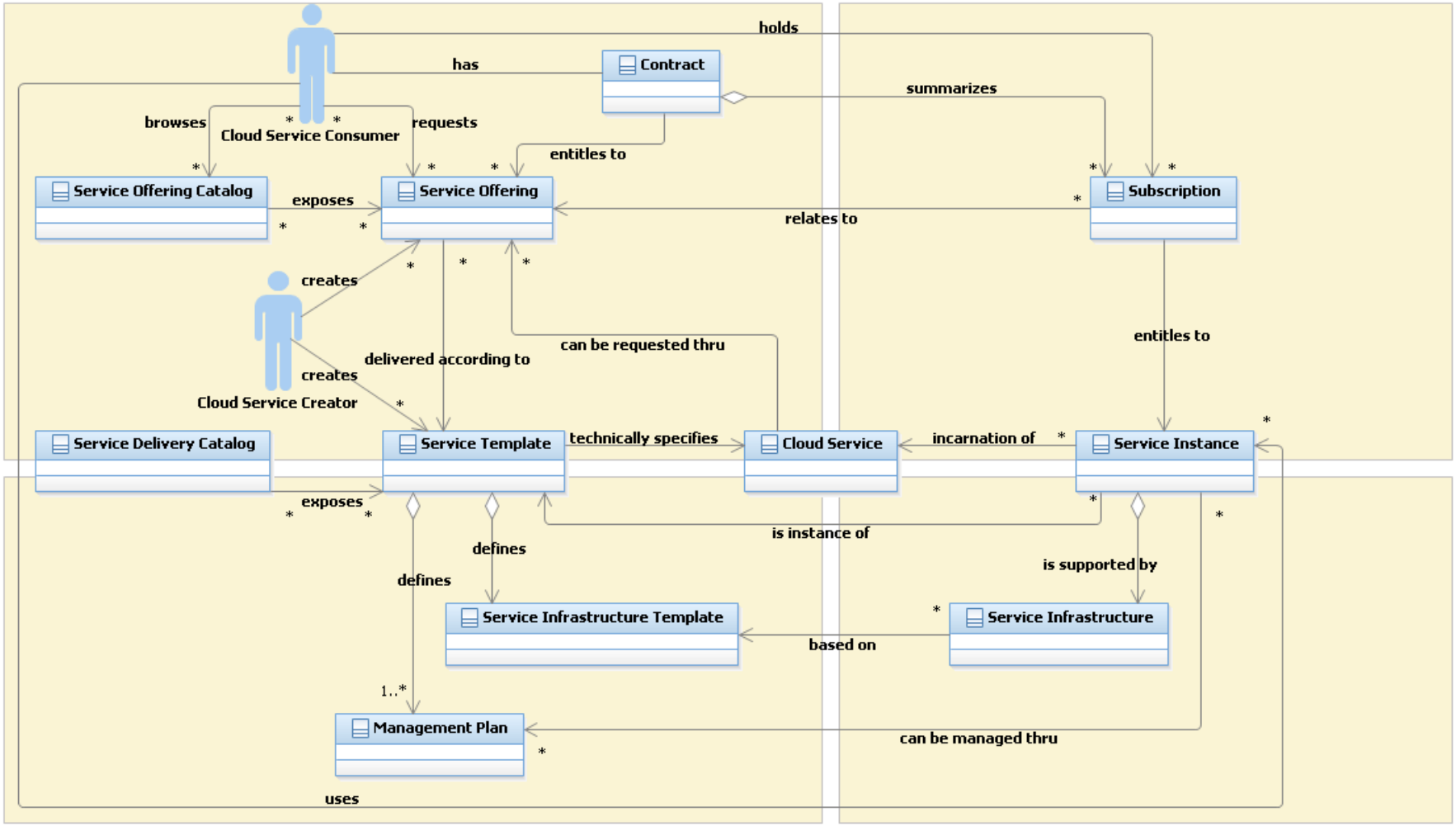
- Existing Cloud Computing and Management Standards
- Standards that are currently being defined
- Standards that need to be defined (recommended standards)

1. APIs
2. BSS UI Portal
3. Cloud Domain Model and Terminology
4. Cloud Management & Orchestration (Service
5. Consumability
6. Management Standards
7. Security
8. Virtualization (OVF, Images Mgmt., Network, Storage)



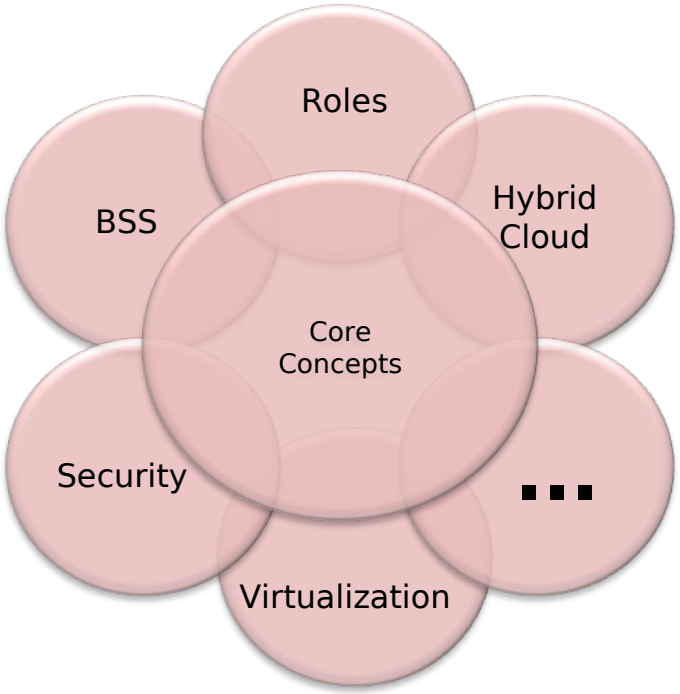
A standard can be defined as something with a pre-described specification, that is measurable, recognized as having authoritative value, and which an organization chooses to implement as a basis for 'good practice'.

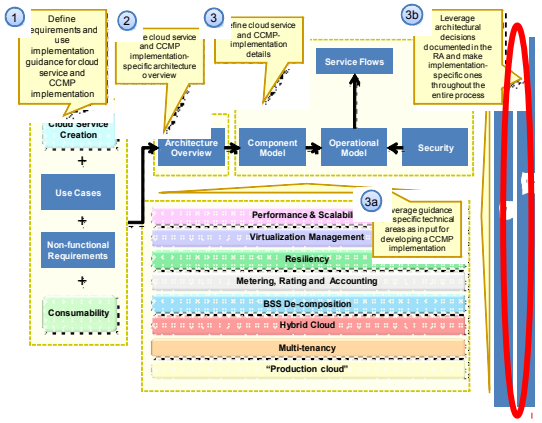
CC RA 2.0 Standards – Core Domain Model Overview



CC RA 2.0 Standards – Cloud Domain Model

- Create a central, authoritative domain model and terminology to be used consistently in cloud solutions
- Cover core concepts that are common across cloud solutions, and their touch points to specific disciplines
  - Do not dive into specific domains to avoid duplication of work
- Strong focus on (1) defining core concepts/terms and (2) defining how those concepts are related to each other
- Collect input from RA 1.0 and RA 2.0 documents, and make consistent with RA 2.0
- Make consistent with models that are defined or are being defined in IBM (e.g. CDM)





Architectural Decisions

Architectural Decisions TWP: Introduction & Purpose

- AD documents important decisions about any aspect of the architecture including the structure of the system, the provision and allocation of function, the contextual fitness of the system and adherence to standards
- Consolidate ADs from key offerings, architectural domains and TWPs to provide a single place to find important ADs that illustrate rationale and justification of architectural decisions
- Develop standard reusable artefacts that can be consumed by offering and engagement architects
- Prove a reference of documented decisions that can help practitioners to:
  - avoid unnecessary reconsideration of the same issues
  - preserve design integrity in the provision of functionality and its allocation to system components
  - ensure that the architecture is extensible and can support an evolving system

Scope Context - Capture ADs from key cloud offerings and architectural domains & TWPs

Cloud Delivery Models

SaaS
Managed Service Desk

PaaS

IaaS
(Public & private cloud)

Architectural Decisions

Architectural Domain Subset*
Technical Work Products

- Use Cases & Roles
- Architecture Overview
- Security
- Component Model
- Operational Model
- Standards
- Non Functional Requirements
- Service Flows

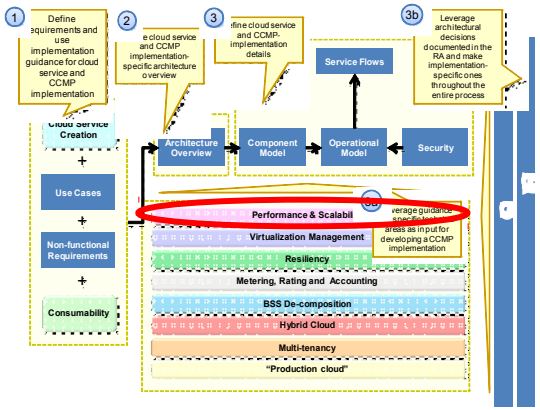
Performance & Scalability
Virtualization Management
Hybrid Cloud
Resiliency
Metering/Accounting/Billing

Architectural Decision TWP Word Document Contains the following detailed ADs…

In version 2 of the AD document we added decisions from the following areas on top of the decisions from version 1:

- Applied Pattern Architecture Decisions
  - IaaS - Public Cloud – 14 ADs
  - IaaS - Private Cloud – 18 ADs
  - SaaS – Managed Service Desk – 11 ADs

- Domain specific Ads
  - Hybrid Cloud – 7 ADs
  - Resiliency – 1 AD
  - Virtualization management – 17 ADs
  - Metering / Accounting – 4 ADs
  - Performance and scalability – 5 ADs



Security

Architectural Principles

IBM Security Framework: Business Security Reference Model

- Governance, Risk, Compliance (GRC)
- People and Identity
- Data and Information
- Application and Process
- IT Infrastructure: Network, Server, End Point
- Physical Infrastructure

Foundational Security Management

- Software, System and Service Assurance
- Identity, Access and Entitlement Management
- Data and Information Protection Management
- Threat and Vulnerability Management
- IT Service Management
- Command and Control Management
- Security Policy Management
- Risk and Compliance Assessment
- Physical Asset Management

Security Services and Infrastructure

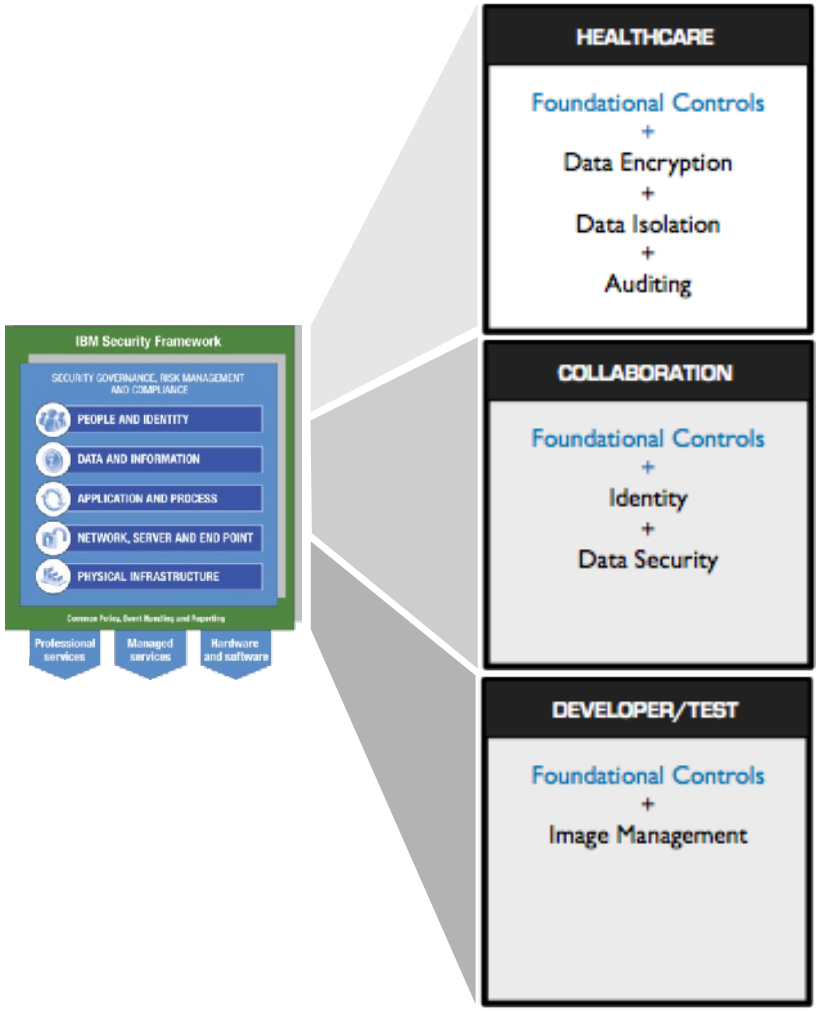
- Security Info and Event Infrastructure
- Identity, Access and Entitlement Infrastructure
- Security Policy Infrastructure
- Crypto, Key and Certificate Infrastructure
- Service Management Infrastructure
- Storage Security
- Host and End-point Security
- Application Security
- Network Security
- Physical Security

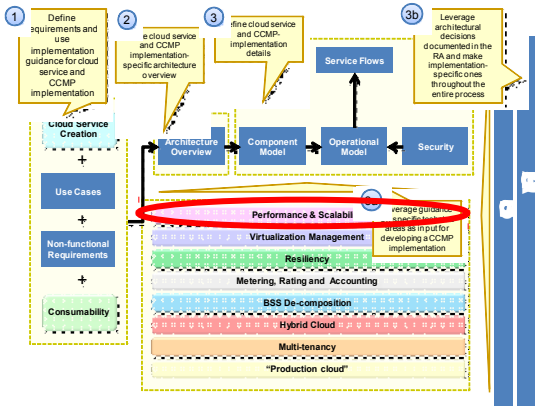
- Security Service Levels
- Code and Images
- Designs
- Policies
- Config Info and Registry
- Identities and Attributes
- Operational Context
- Events and Logs
- IT Security Knowledge
- Data Repositories and Classification

Activity/Data Driven Cloud Security

Organizations need to adopt a strategy for cloud security that considers the unique attributes of the cloud as well as the activities and data the cloud is being utilized for.

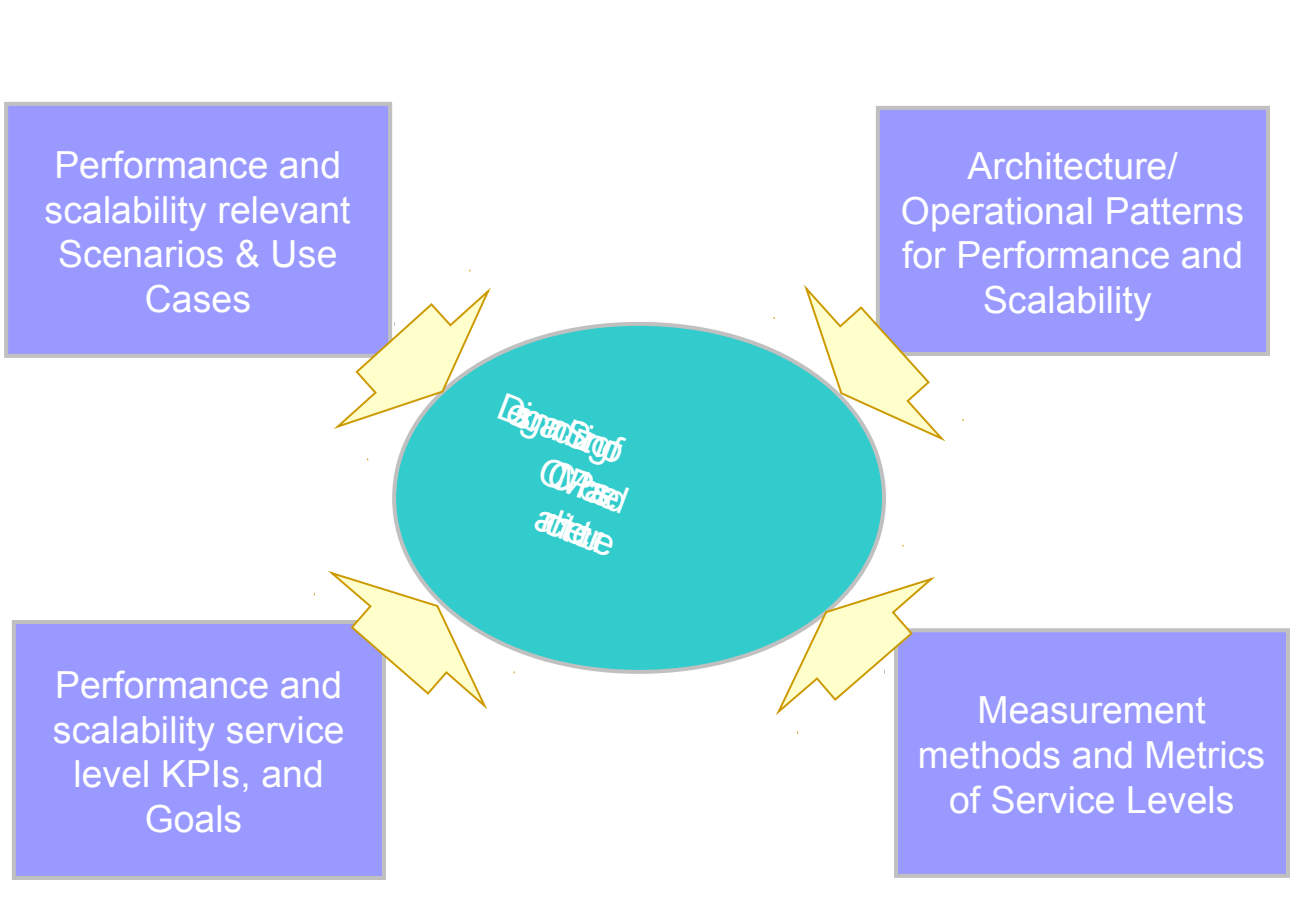
Only by combining foundational controls with activity/data specific controls can organizations meet their cloud security needs.





Performance & Scalability

Performance & Scalability WG: Scope



Roadmaps from Tivoli, Linux Technology Center (STG), and SoNAS (STG)

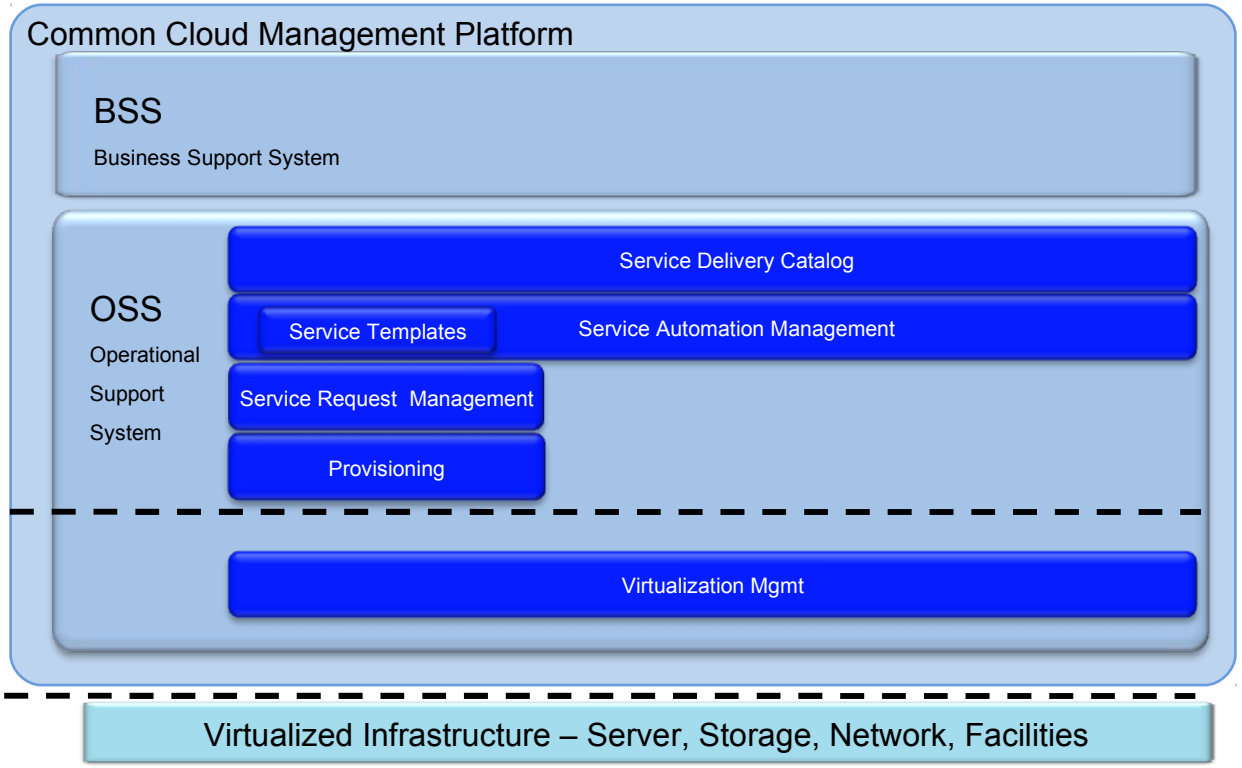
Architectural Patterns

Initial focus on these CCMP Components

- Deployment patterns for OSS components
  - Multiple Virtualization Management domains
  - Horizontal Scaling of Provisioning components
  - Horizontal Scaling of Service Automation components
  - Multiple OSS domains

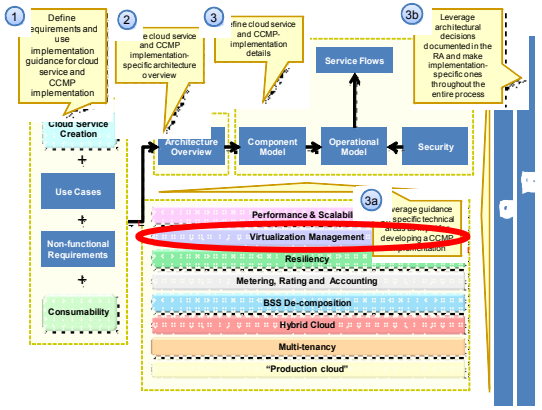
- High Scale Low Touch Virtualization Management

- Patterns and strategies for rapid provisioning
  - CoW boot disk of remote read-only images
  - Local caching of remote images and CoW root disk of local read-only image
  - Local caching of remote images and only transferring image delta
  - Use a combination of CoW, CoR, and Pre-fetching
  - Pre-create/hibernate/resume VM



Outline of the WP document

- 1 Introduction
  - 1.1 Scope of this document
  - 1.2 Supporting work products & references
  - 1.3 Legal Remark
- 2 Scenarios, Use Cases, KPIs, and Measurements
  - 2.1 Use Cases and Actors
  - 2.2 Influencing Scenarios and Factors
  - 2.3 Key Performance Indicators and Goals
  - 2.4 Measurements & metrics
- 3 Deployment patterns for CCMP OSS components
  - 3.1 Multiple Virtualization Management domains
  - 3.2 Horizontal scaling of Provisioning components
  - 3.3 Horizontal scaling of Service Automation components
  - 3.4 Multiple OSS domains
- 4 High Scale Low Touch virtualization management
  - 4.1 Architecture
  - 4.2 Functions
  - 4.3 Results
  - 4.4 Targeted scenarios and use cases
- 5 Virtual systems rapid provisioning strategies
  - 5.1 CoW boot disk of remote read-only images
  - 5.2 Local caching of remote images and CoW root disk of local read-only image
  - 5.3 Local caching of remote images and only transferring image delta
  - 5.4 Use a combination of CoW, CoR, and Pre-fetching
  - 5.5 Pre-create/hibernate/resume VM
- 6 Applied Pattern: Compute Cloud
- 7 References

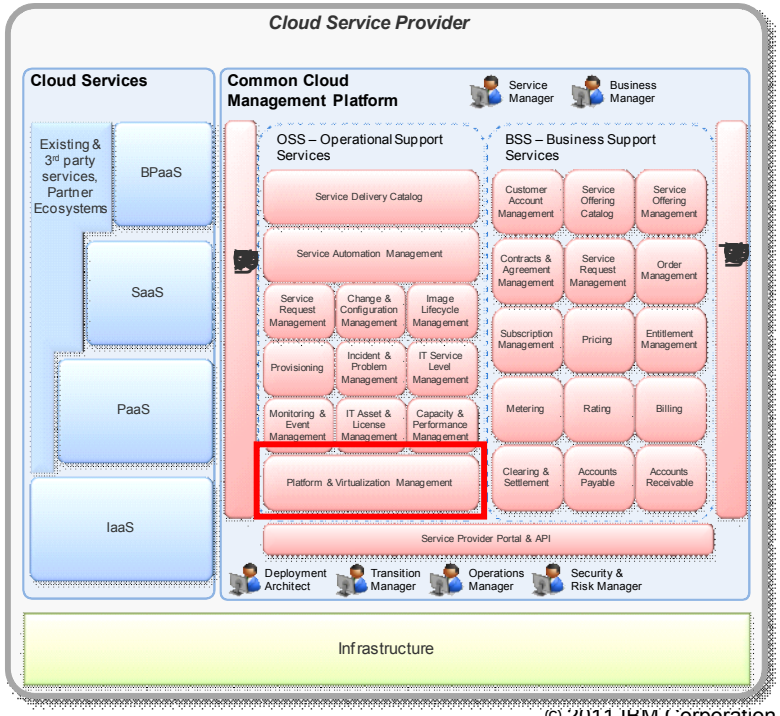


Virtualization Management – Base

Virtualization Management - Base

- Component of the OSS layer
  - provides common interface across hetergenous infrastucture & value-add functions above infrastructure (pooling, placement, mobility, notification up)
- Defines the capabilities and services for the management of virtual resources and the underlying physical resources
- Base virtualization divided into 2 main sub components
  - Platform Management
  - Virtual Resource Management
- Network and Storage Virtualization are sub teams and discussed separately

- Details what functions/subcomponents should be part of the Virtualization/Platform Management component
  - Definition of scope and functions
- What is important for implementation/options in cloud environments for these functions
  - Considerations and what options are there, pros and cons
  - Performance, scaling and architecture decisions
  - Ex: local storage vs shared storage the implications there wrt Performance, scaling, resiliency of the VMs
- What standards/recommendations are there for implementations



Virtualization and Platform Management

- **Virtualization Management**
  - Virtualization Platform Abstraction
    - Common interface to request and access virtual resources
    - Enables commonality for the higher level functions in the BSS and OSS layers
  - Virtual Resource Pooling
    - Ability to manage a set of virtualization infrastructure as a single entity or “pool”
    - Includes Compute, Network and Storage infrastructure
    - Virtualization management component handles placement/allocation across the pool and optimization of the pool.
    - Resource allocation can be based on a variety of criteria and factors such as energy, availability, licensing, etc
  - Reservation
    - Reserves virtual resources for future use
    - Provides a reservation interface
  - Image Repository
    - Library of virtual appliances which can be deployed to create workloads
  - Deployment and Undeployment
    - Creation and deletion of virtual systems and network/storage resources at the request of the provisioning layer
    - Provides for intelligent allocation across the pool, may include over commit of physical resources
  - Instance Image Management
    - Provides management of deployed instances of the images, including ability to capture images
  - Management of virtual servers
    - Provides interface to start, stop, restart virtual servers
  - Relocation
    - Provides ability to move a virtual resource from one physical machine to another
    - May happen as a result of a failure, predicted failure, or planned maintenance

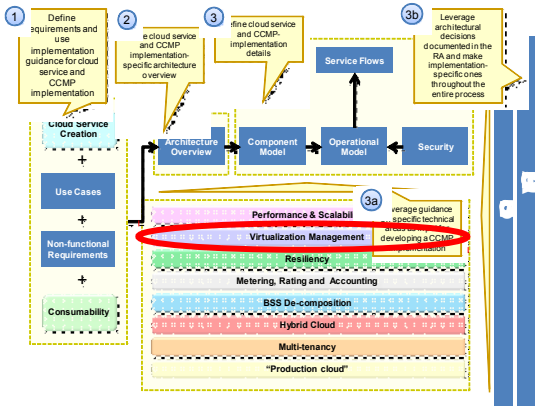
- Platform Management
  - Discovery of physical resources
    - Library or catalog of discovered resources
    - Inventory such as options, firmware, OS
  - Platform Software Maintenance
    - Monitoring, reporting, and patching of system firmware and hypervisor software.
  - Bare Metal installation/distribution
    - Install and distribute OS and hypervisor software and agents
  - Platform health monitoring and reporting
    - Monitors health of physical systems and hypervisors
    - Used for billing services and SLAs

Virtualization and Platform Management

- Standards
  - Platform management
    - DMTF CIM, SNMP
  - Virtualization management
    - DMTF OVF
    - Libvirt
    - Activation engine - VSAE

- Non-Functional Requirements
  - Mostly centers around performance and scaling which is critical in a cloud environment
  - Details Documented in the NFR document
- Input provided to other work products for performance, resiliency, component model, etc

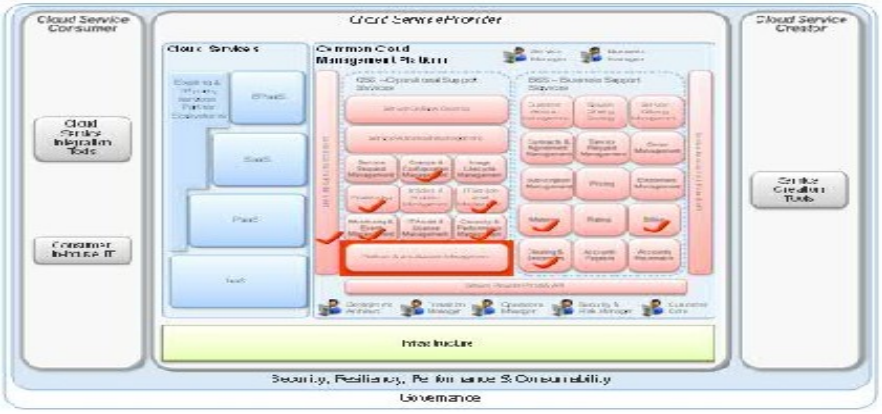
- Applied Patterns
  - Private cloud, Cloudburst, Telco scenario

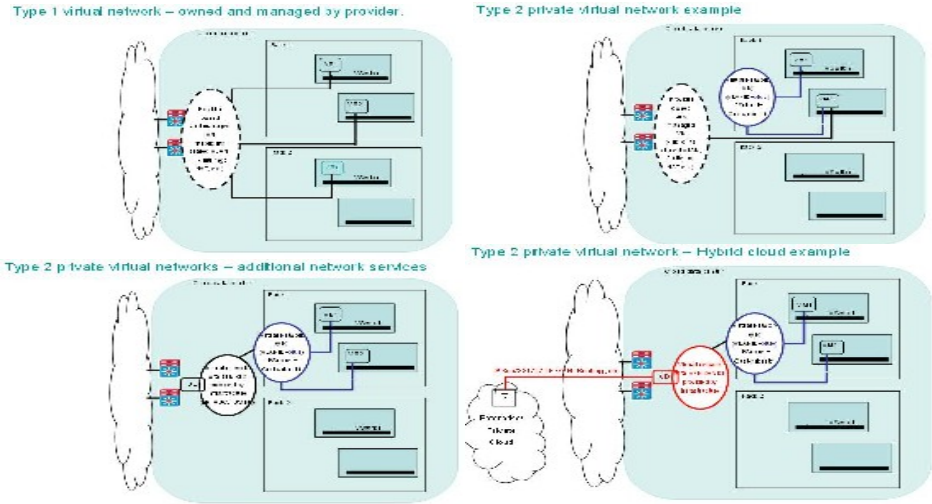


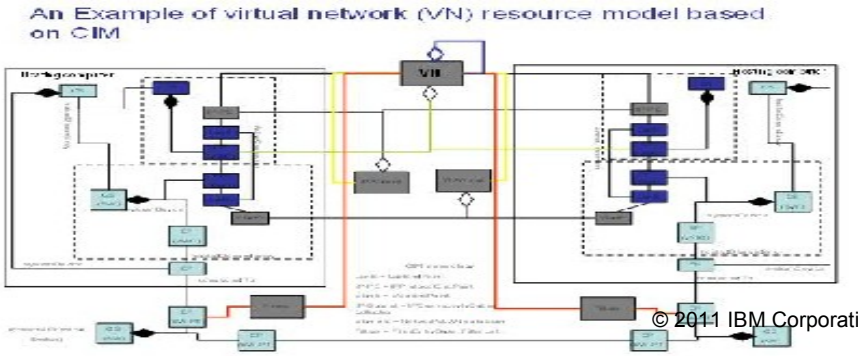
Virtualization Management – Networking

Virtualization Management – Networking Summary

- Identify BSS/OSS core components that are relevant to cloud (virtual) networking.
  - Capability extensions for virtual networking architecture and services.
- Identify a set of important cloud networking connectivity types (and patterns) to support cloud users (customers):
  - Shared cloud infrastructure network(s).
  - Private networks w/in cloud infrastructure (with variation of access over Internet).
  - Private cloud to cloud virtual network (including hybrid cloud – from customer's network to cloud).
- Identify/define resources in resource model (i.e., CIM), critical for supporting cloud networking as part of infrastructure authoritative topology DB.

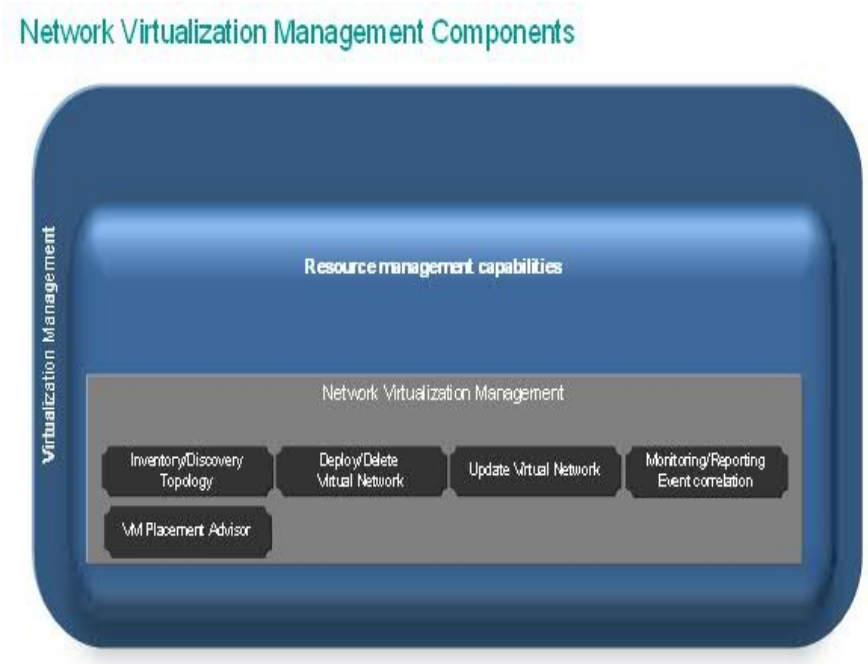


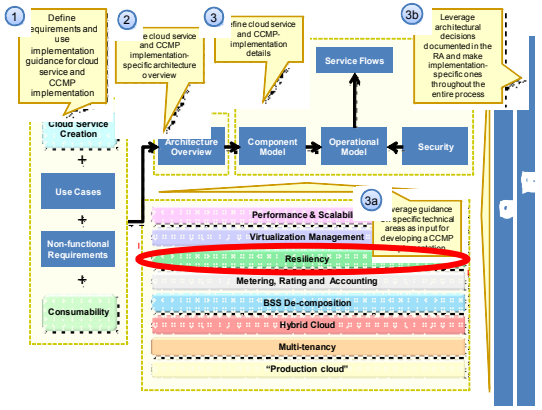




Virtualization Management – Networking TWP Summary

- Define subcomponents (and their expected capabilities) and services of Networking Virtualization Management in Virtualization and Platform Management overall component.
- Identify Architecture Decisions and Non-Functional Requirements related to networking.
- Extend cloud use cases to include networking connectivity (i.e., connecting cloud services to virtual networks and automated network provisioning).
- Relate cloud networking to on-going standard work in DMTF SVPC WG for virtual networking within OVF.
- Identify IBM products/assets that provide foundations for building out support for cloud networking and automated network provisioning.



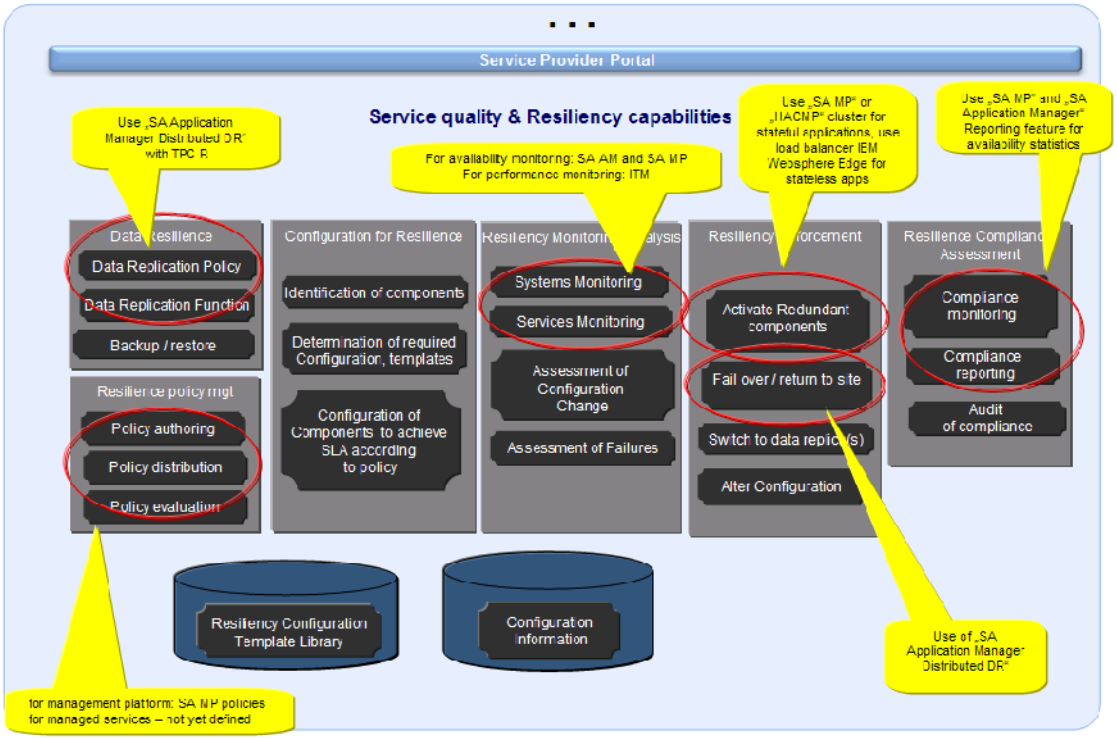


Resiliency

CC RA 2.0 Resiliency

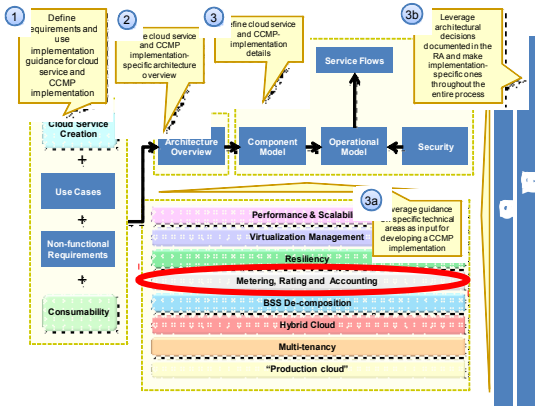
CC RA 2.0 Resiliency

- Defined cloud Resilience architecture
- Based on lessons learned from implementations
- Mapped architecture to proposed CCMP resilience implementation
- Identified required products
- NFRs enhancements



Key learning from existing cloud implementations

- OSS/BSS interdependencies and operational requirements must be considered in coordinated resilience and automation design



Metering, Rating and Accounting

Metering, Rating & Accounting: Executive Summary

- Metering, Rating and Accounting Domain addresses:
  - Different types of metered usage data (3 Key types) and the need to meter both Allocated and Activity based data
  - BSS-OSS mapping of runtime and design time data flows
  - Product specific detailed functional gap analysis with tool recommendations
  - Solution architecture along with implementation guidance

- 3 Applied Patterns from architecture as applicable to Public, Private clouds and an ISV context are explained

Terminology: Metered Usage Data Concepts

- Allocation Usage
  - The result of a consumer occupying a resource such that other consumers cannot use it
  - For example, the time period IT infrastructure topology (e.g. Servers, CPUs, Memory, Storage, Network, Database, Websphere Cluster) has been allocated to a particular cloud service.
  - More suitable as Service Usage Metric.

- Activity Usage
  - The result of activity performed by the consumer e.g. CPUSecs, Bytes transferred etc
  - More suitable as Cost Usage Metric.

- Action Usage
  - Actions initiated by the consumer that the provider may wish to charge for or track costs against
    - e.g. Backup/restore server, change virtual server configuration
  - Action may or may not involve manual steps

Cloud Delivery Requires Allocation and Activity Usage

Cloud Service Requestor #1

Service A
- ✓4 CPUs on Server 1
- ✓2 GB
- ✓60 server hrs @ $25/ server hr
- ✓240 CPU hrs @ $5/ CPU hr
- ✓120 GB hrs @ $3/ GB hr

Cloud Service Requestor #2

Service A
- ✓3 CPUs on Server 2
- ✓4 GB
- ✓50 server hrs @ 25/ server hr
- ✓150 CPU hrs @ $5/ CPU hr
- ✓200 GB hrs @ $3/ GB hr

Cloud Service Requestor #3

Service B
- ✓2 CPUs on Server 1
- ✓1 GB
- ✓100 server hrs @ $20/ server hr
- ✓200 CPU hrs @ $3/ CPU hr
- ✓100 GB hrs @ $2/ GB hr
- ✓100 DB2 Support hr @ $2/ DB2 Support hr

Allocation Usage

3,060

$2,600

Service A revenue = $5,660

Service B revenue = $3,000

Activity Usage

Cloud Service Provider

Service A Costs
- ✓Server 1 CPU usage = 3,000 CPU mins * $.40/min
- ✓Server 2 CPU usage = 2,500 CPU mins * $.60/min
- ✓Server 1 software usage = 60 hrs * $15/hr
- ✓Server 2 software usage = 40 hrs * $10/hr
- ✓Server 1 labor costs = $40
- ✓Server 2 labor costs = $60

Service A costs = $4,100 Profit

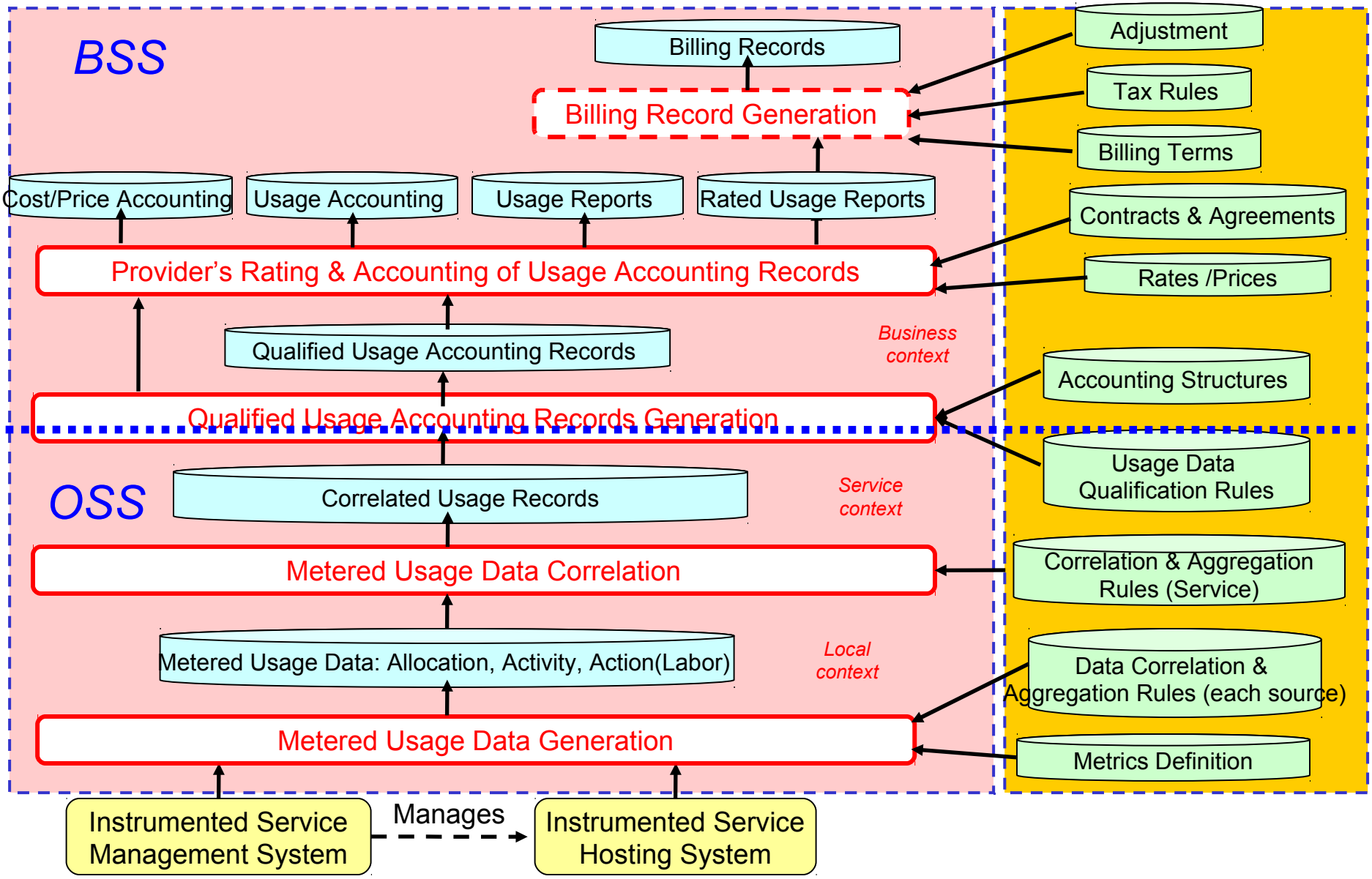
Cloud Service Provider

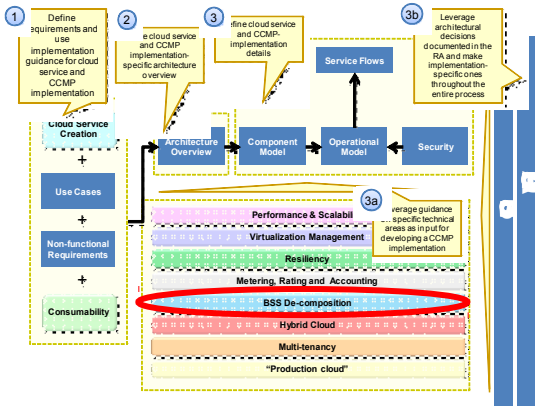
Service B Costs
- ✓Server 1 CPU usage = 5,000 CPU mins * $.40/min
- ✓Server 1 software usage = 80 hrs * $15/hr
- ✓Server 1 labor costs = $40
- ✓DB2 Help Desk = $50

Service B costs = $3,290 Loss

Cost and prices are for illustration purposes only

Metering Rating & Accounting: BSS – OSS Mapping

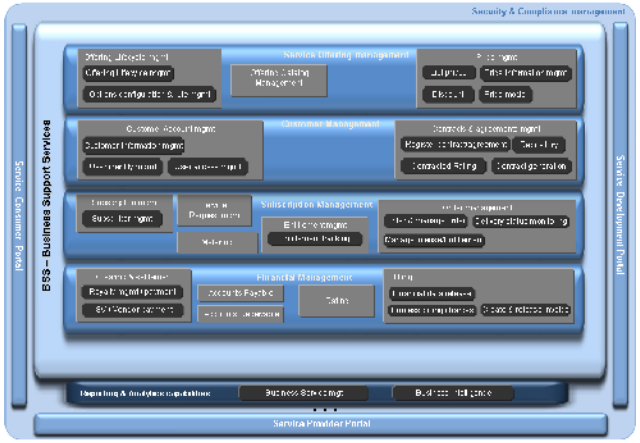




BSS Decomposition

Business Support Services Decomposition

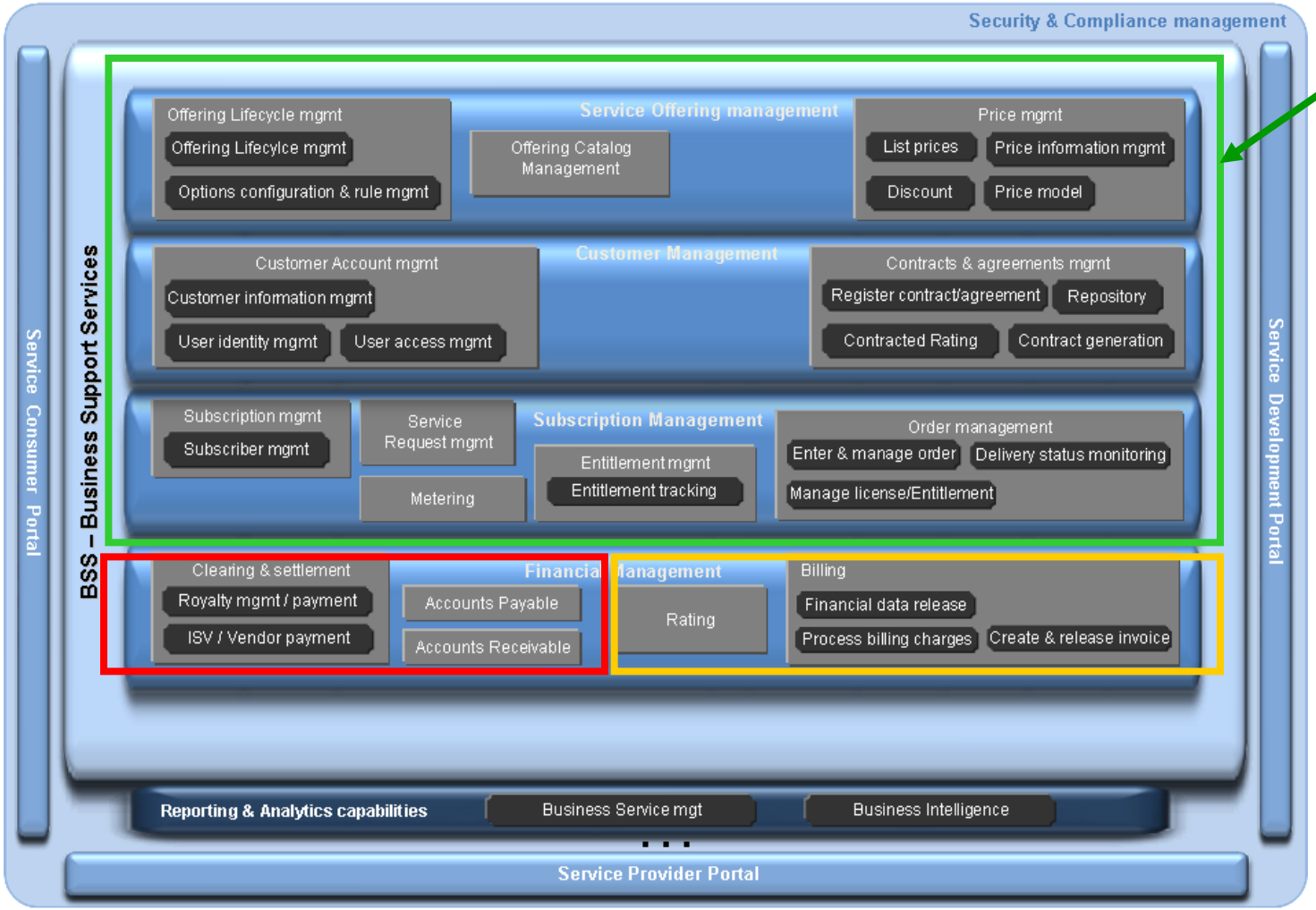
- Business Support Services (BSS) is a set of components that enables interaction of customers with the service provider. BSS works closely with Operational Support Services (OSS), which is a set of computer systems and related processes used to deliver the services. BSS and OSS functions for all IBM Cloud services are provided by the Common Cloud Management Platform (CCMP)



- BSS Decomposition RA content - 44-page doc with the following sections:
  - BSS positioning within CCMP
  - Overview of BSS functions required for all realizations
  - Public Cloud realization
  - Private Cloud realization
  - Brief discussion of hybrid cloud and product development realizations
  - Future direction – BSS as a Service for promoting re-use and modularization

CCMP RA 2.0 conceptual CM for Business Support Services

Private Cloud Realization - using Tivoli Service Request Manager – Service Provider Edition + Tivoli Usage and Accounting Manager



TSRM-SP provides all these functions

Notes on METERING

TSRM provides Allocation & Activity-based Metering.

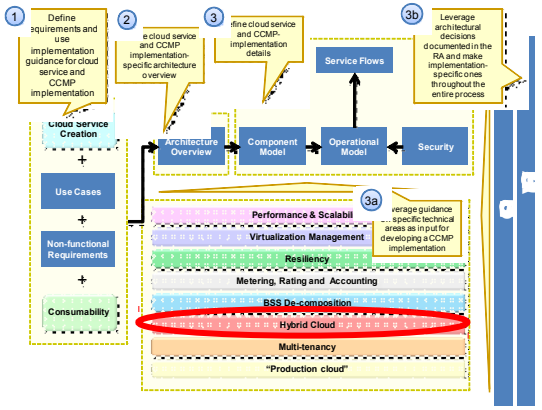
TSRM tracks labor, materials, tools, services costs.

TUAM provides Usage-based Metering

RATING & BILLING

TSRM & TUAM provide partial rating functions & bill data generation – but not direct invoicing

NO Peering and Settlement
NO accounts payable/receiving



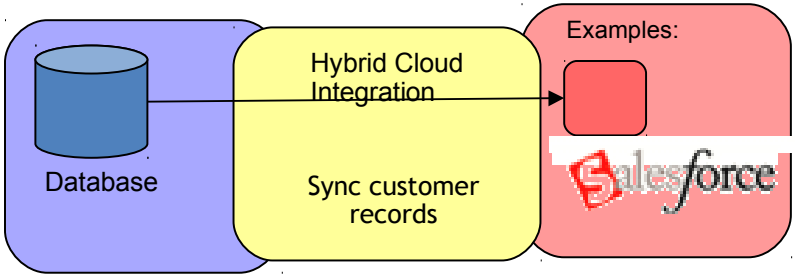
Hybrid Cloud

Hybrid Cloud Workstream: Scope and Dimensions

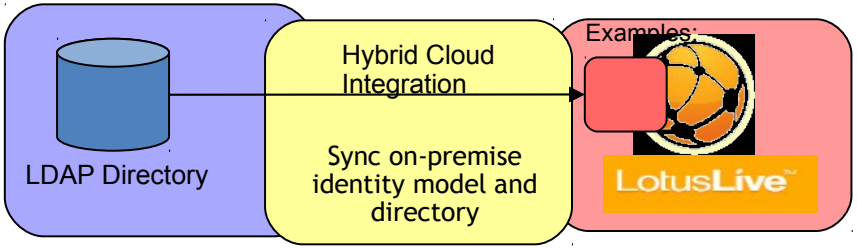
- **Scope and Purpose:**
  - **Use Cases:** Identify use cases and scenarios for hybrid cloud setup, operations, and management
  - **Patterns:** Identify solution patterns for integration of on-premise with services in public cloud(s)
  - **Life-cycle:** Identify and define workload migration and life cycle events for services in hybrid cloud
  - **Roles:** Identify roles associated with hybrid cloud operations and services
  - **Decisions:** Define architectural decisions for hybrid cloud integration framework and for hybrid cloud management services
- **Perspectives**
  - **Operating Perspective:** Seamlessly move peek workloads from on-premise to public Cloud
  - **Sourcing Perspective:** Different types of workloads to be provisioned by the most effective Cloud from the perspective of cost, functionality, availability, performance, security, etc.
  - **Management Perspective:** Unified view and capability to manage resources and information on-premise and in off-premise Clouds combined with management and integration of workloads and resources across hybrid cloud

- Dimensions:
  - Integration: How to connect on-premise services and data to off-premise counterparts… business data mapping and service integration
  - Security: How to integrate on-premise/off-premise identities, policies, auditing systems; how to ensure proper security of off-premise cloud workload; How to secure management and payload interactions
  - Monitoring: Integrate monitoring of off-premise infrastructure and applications with on-premise management system; Enable on-premise monitoring and event infrastructure to reach into clouds
  - Management: Manage Capacity in the cloud; provisioning- and de-provisioning based on Monitoring data), capacity overflow from on-premise to Cloud
  - Governance: Who can, does, should use which cloud-based services…service request management of on- and off-premise resources

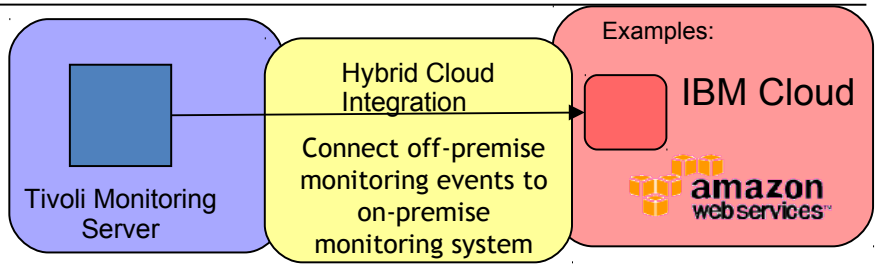
Hybrid Cloud: Use Case Scenarios



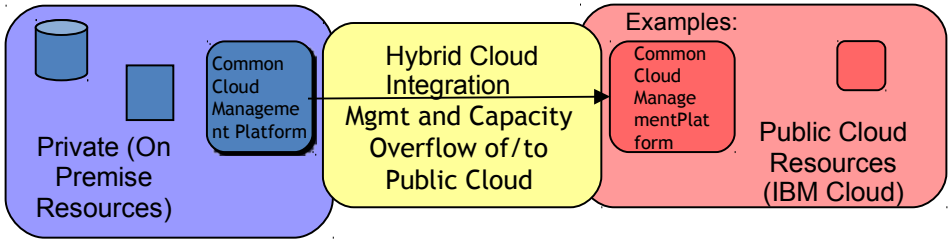
On-premise to off-premise Business Application Integration:
Example:
On-premise Database to Salesforce Cloud



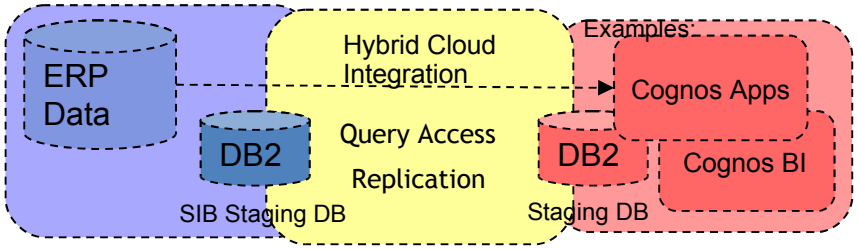
Security: Directory Integration & Identity Federation Example
Synchronize on premise ODW LDAP and LotusLive Domino directory info and facilitate SSO



Hybrid Monitoring
Example
Federate Monitoring info of Workload in IBM Public Cloud



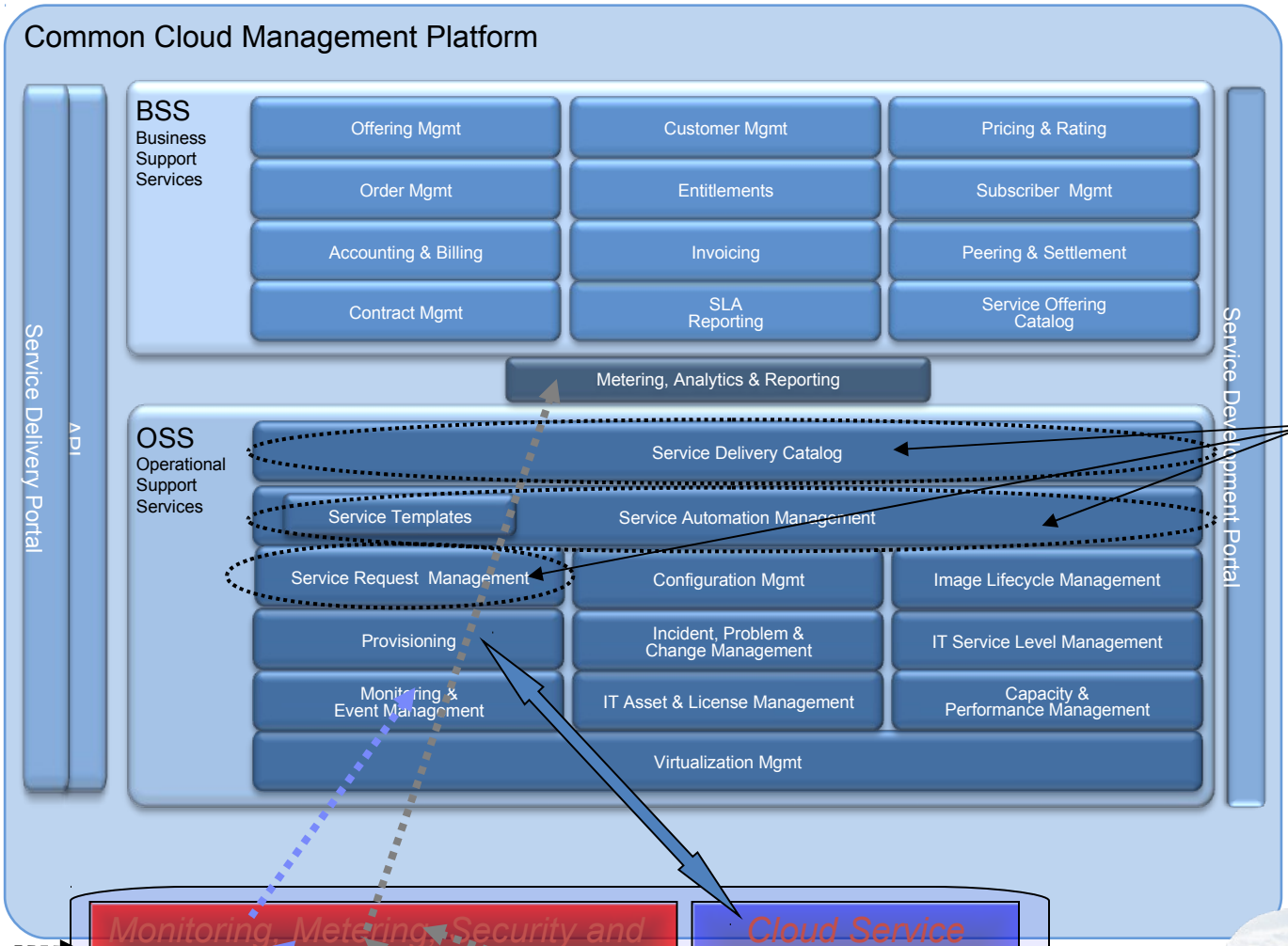
Governance & Hybrid Workload Management
Examples:
Governance for acquiring Resources from IBM Compute Cloud and/or Amazon; Workload Management and capacity overflow from CCMP based private Cloud to IBM Compute Cloud

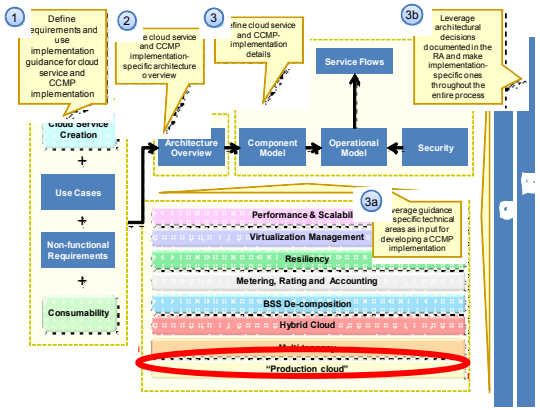


Cognos and other Analytic Applications in the Cloud:
Initial Load to Cloud; Change Data Capture Replication to & from Cloud; Query Access; Data Cleansing

* Cloud Management Platform – Hybrid Cloud Model

IBM Cloud Computing Reference Architecture





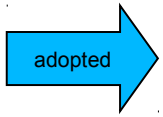
Multi-tenancy

Multi-tenancy - scope

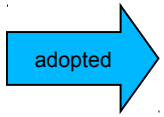
Done
Partial

- Multi-tenancy requirements and customer expectations
- Requirements mapping to CC RA component for impact and gap analysis
- Multi-tenant patterns
- Leveraging existing multi-tenancy assets and capabilities
- Specification of how secure Multi-Tenancy should be implemented across the CC RA components

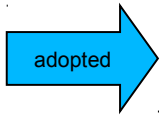
Multi-tenancy deliverables

Multi tenant NFRS 

NFR deliverable

Use Cases and extensions 

Use case deliverable

Component impact analysis 

Component deliverable

Multi-tenancy definition

Tenant definition

Multi-tenant patterns whitepaper

Multi-tenant definition

Multi-tenancy refers to the ability of services to be offered to multiple user entities (tenants) in a way so that each tenant operates as logically isolated, while, in fact, using physically shared resources.

Tenant definition

A business entity that is contractually responsible for the consumption of Cloud Service from a service provider.

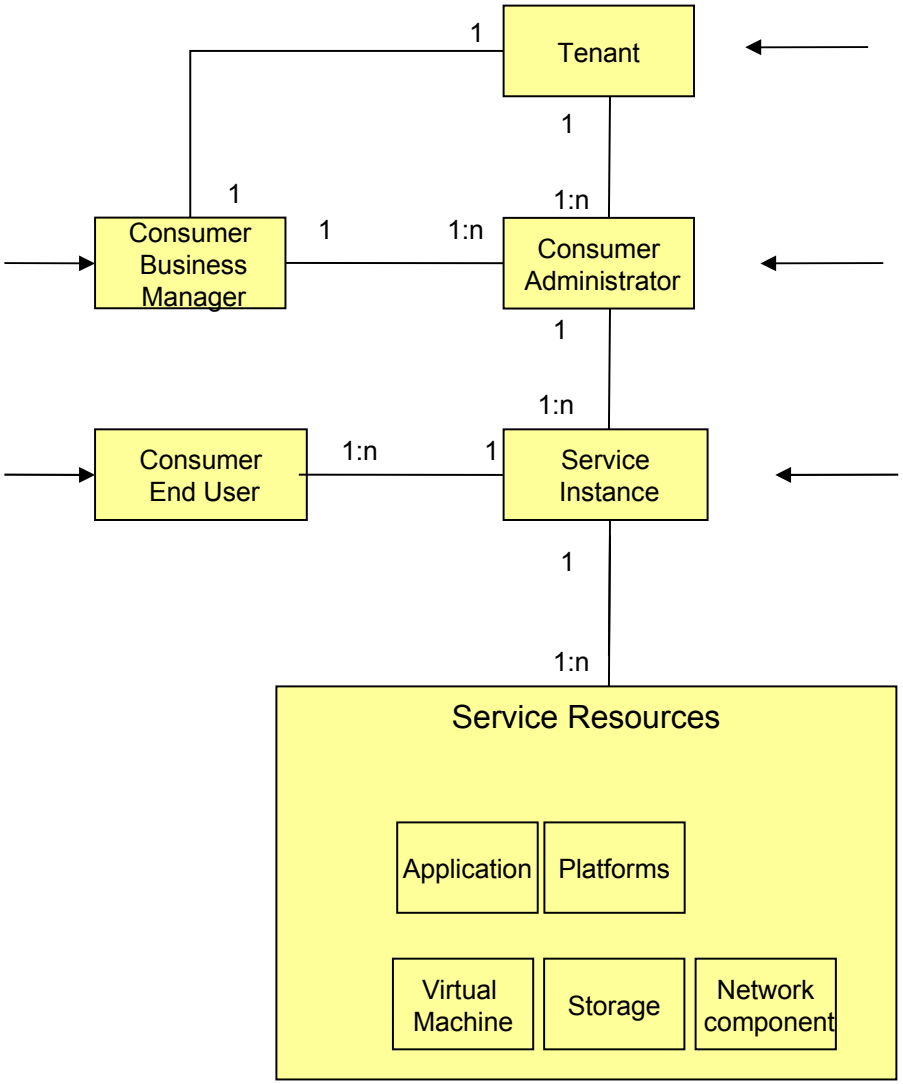
A role within the business entity that manages contracts and optionally approves commission of services from the Cloud

A role within the business entity that will request commission and decommission services from the Cloud.

The users of the Service

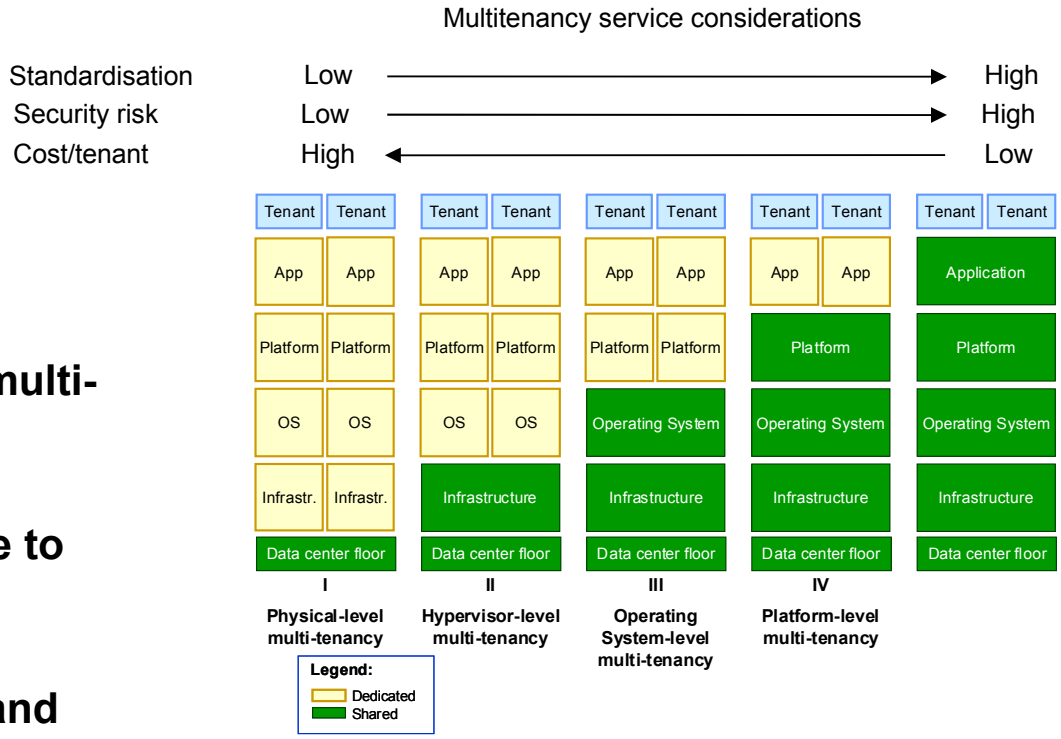
The Cloud service instance requested by the Service Owner

The resources consumed by the service



Multi-tenancy whitepaper

1. **Primitive set of patterns of multi-tenancy**
2. **Associated multi-tenant characteristics that will have to be supported in any Cloud implementation.**
3. **Multi-tenancy of managing and manage environments**
4. **Multi-tenancy needs of public and private clouds.**



Cloud Computing Reference Architecture 2.0: Overview

CC RA team / presented by Mike Buzzetti

Agenda

- Executive Summary
- Introduction & Overview
  - Historical evolution of the Reference Architecture
  - Structure of the RA 2.0
  - Target audience
  - How to apply the RA for cloud implementations
- Cloud Computing Reference Architecture Details

IBM Cloud Computing Reference Architecture

The IBM CC RA represents the aggregate experience across hundreds of cloud client engagements and the implementation of IBM-hosted clouds

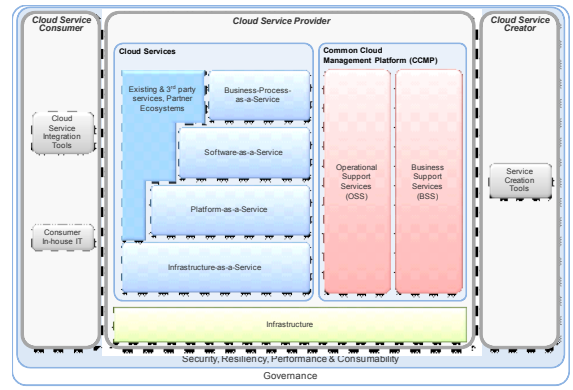
- Based on knowledge of IBM's services, software & system experiences, including IBM Research

The IBM Cloud Computing Reference Architecture (CC RA) is reflected in the design of

- IBM-hosted cloud services
- Clouds IBM implements for clients
- IBM cloud appliances
- IBM cloud service management products

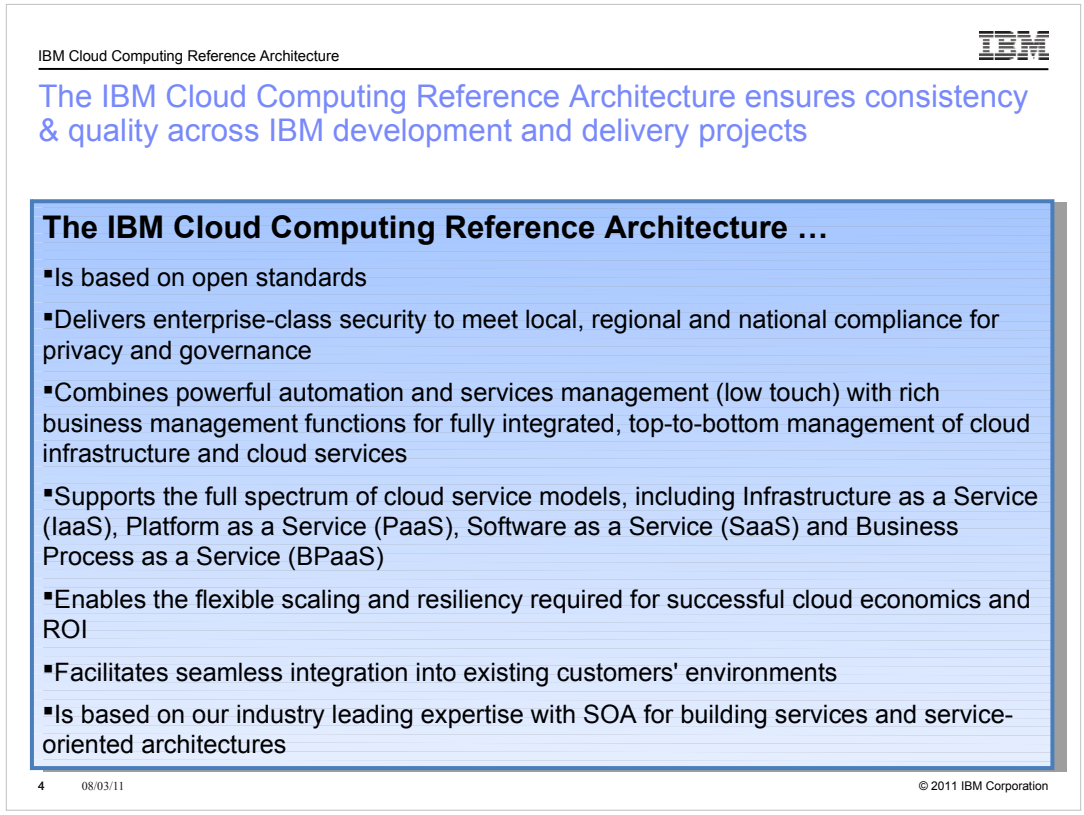
The CC RA focuses on cloud specifics such as radical cost reduction while achieving high degrees of security, reliability, scalability and control

The CC RA consists of 21 detailed documents representing best-of-industry knowledge and insight on how to architect, design and implement clouds



Mike Buzzetti notes

Existing legacy products and technology can be mapped to to the CCRA since the CCRA shows integration points between new cloud technologies and the existing ones



SOA is defined by the open group as “an architectural style that supports services orientation Service orientation is a way of thinking of services and services-based development and the outcomes of services. “

That basically means that cloud computing supports service orientation

SOA solutions usually do not have all the characteristics of cloud simultaneously.

SOA solutions usually have to have the management built from scratch

SOA service reuse is generally only within the organization that delivers the service.

Not all SOA solutions are cloud servers, since cloud requires automated deployment and management

IBM Cloud Computing Reference Architecture: Development Process

Development led by the IBM Cloud Computing Architecture Board

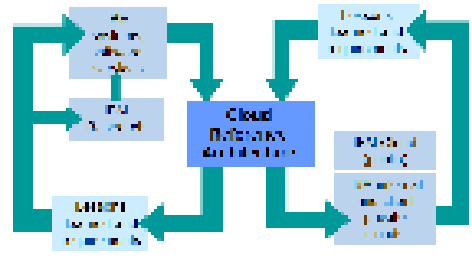
- Comprising technology leaders from IBM Research and IBM's software, systems and services organizations
- >50 of IBM's top cloud computing experts represent the core team

Derived from extensive client interaction combined with IBM's extensive capabilities and experience in building enterprise-class IT systems.

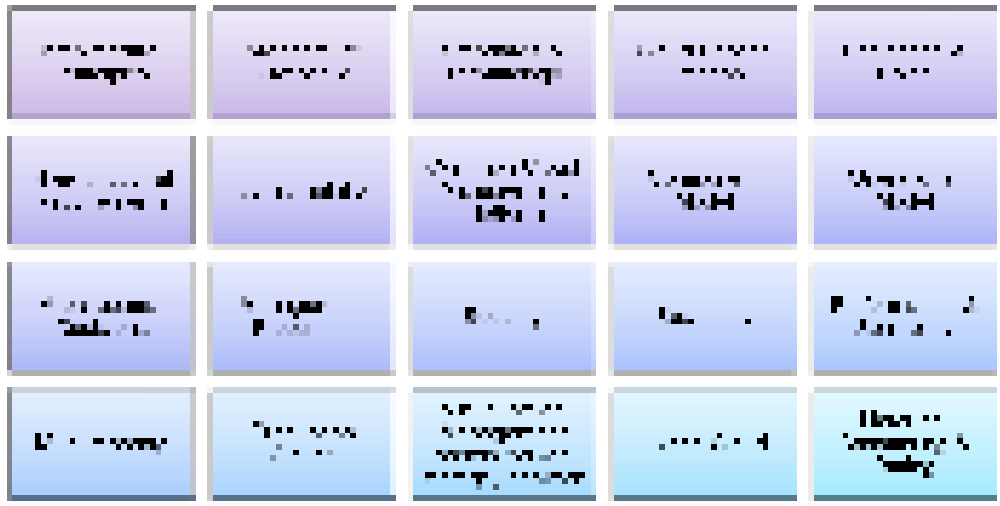
The CC RA provides specifications for

- the physical components of a cloud implementation (network, compute, storage, virtualization)
- Software components required to run management
- Operational processes
- Governance policies tailored for the environment or enterprise.

A process of continuous improvement helps ensure that the reference architecture is both responsive to changing client and security requirements and incorporates technology developments via IBM products and evolving design specifications.



The IBM Cloud Computing Reference Architecture covers are broad range of important cloud-specific topics



Structure of the IBM Cloud Computing Reference Architecture

There are two deliverables for each work product of the RA

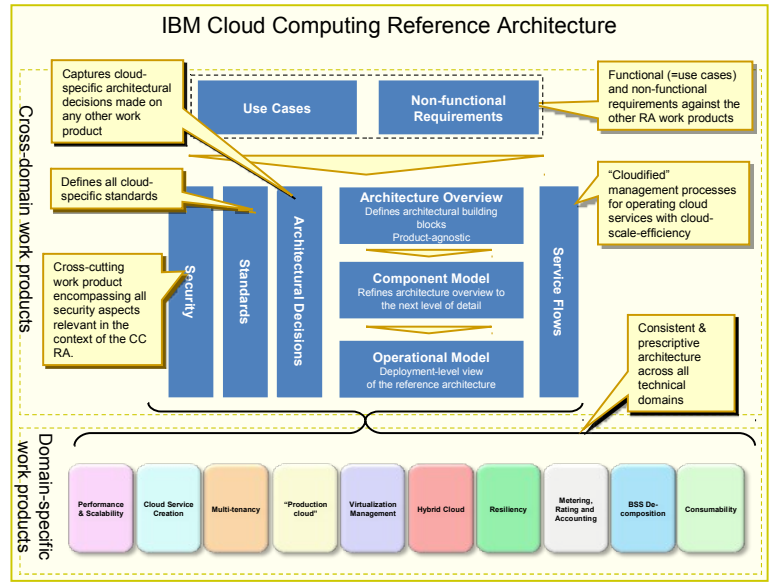
- **Document**: Detailed description of the work product, important for in-depth work
- **Presentation**: Summary of the document, for simplified consumption and quick start

The RA is defined according the Unified Method Framework (UMF)

→Simplifies field adoption since all IBM field architects attend UMF training by default

Each work product includes one or more "applied patterns"

- An applied pattern illustrates how the respective work product could be used / was used in a specific implementation



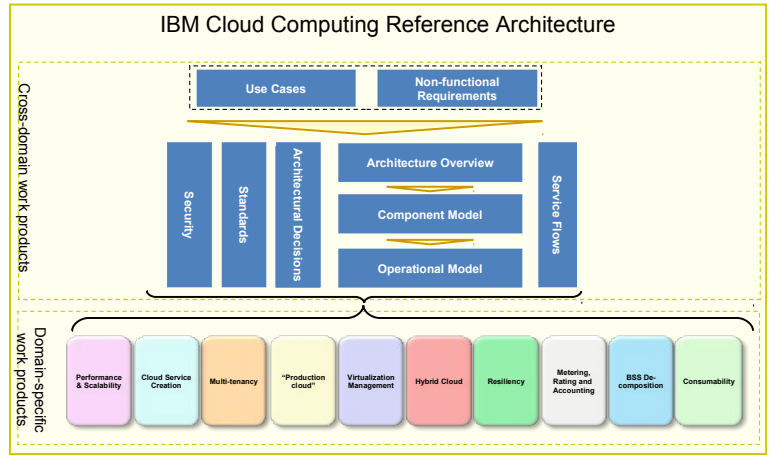
IBM Cloud Computing Reference Architecture: Domain work product details

How to implement a cloud to achieve optimal performance & scalability

Guide for creating cloud services and how to exploit management functions.

Considerations for implementing multi-tenancy in a cloud environment

Example for how to implement a cloud management system for offering production-level cloud services



Definition for how to use virtualization management across server, storage and network

Architecture & implementation considerations for implementing a hybrid cloud

Resiliency considerations for implementing a cloud

Details on metering, rating and accounting, including prescriptive guidance with respect to implementation

Detailed definition of BSS domains and how they should be implemented from a product perspective

How to optimize the end to end experiences related to cloud services

The IBM Cloud Computing Reference Architecture – Aggregating past & present experience and future requirements

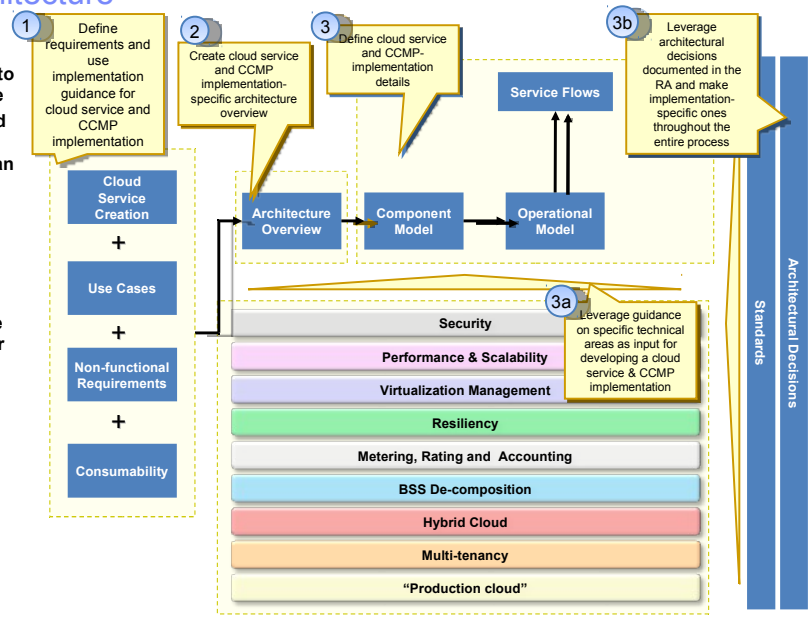
CC RA

=

Harvesting ("Learning from cloud deployments in the ***past***?")	+	Refinement ("Improve the CC RA content **presently** available")	+	Forward looking ("Which ***future*** topics are relevant for cloud computing")
• Harvest learnings & material from cloud implementations while factoring out short cuts taken by implementation team (e.g. due to constrained time lines). • Continuously ongoing harvesting process due to additional learnings during each phase		• Continuous improvement of CC RA content based on feedback from CC RA exploiters & stakeholder • Be clear about which parts of the RA are already addressed in today's IBM cloud offerings and which aspects still have to be addressed		• Today's cloud implementations are not necessarily optimal implementations (due to time & resource constraints) → Need forward-looking guidance on ideal target implementation •Examples are: • generalization of CCMP services • layered cloud services • NFRs, e.g. security, resiliency • Hybrid clouds

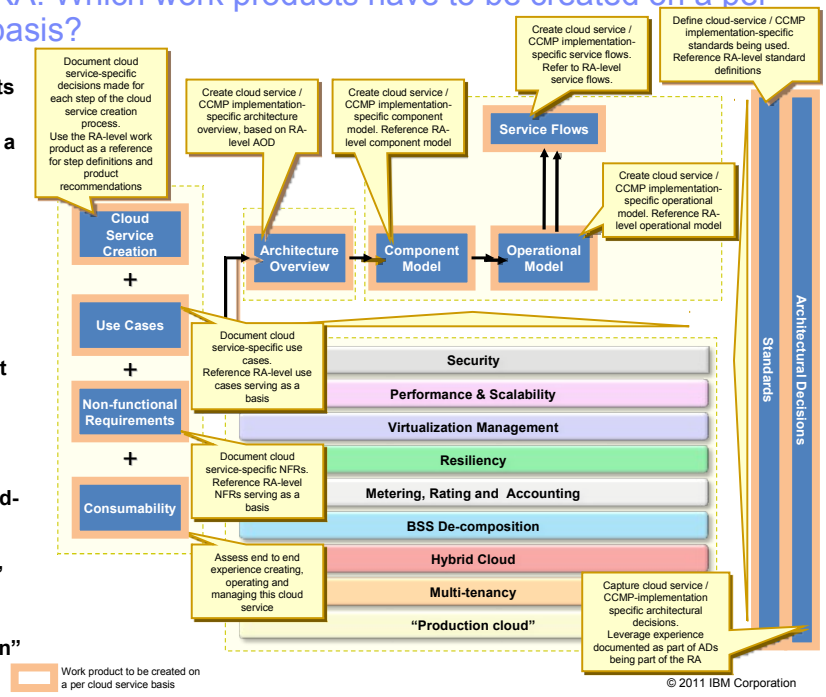
How to implement a cloud using the IBM Cloud Computing Reference Architecture

- **There is a well-defined process for using the RA to implement a cloud service**
  - **A cloud is constituted by a CCMP implementation and an arbitrary number of cloud services delivered and managed by it**
- **A dedicated presentation and document is available describing the process for using the RA to develop the management aspects for a cloud service**
- **RA work products also serve as authoritative reference on specific technical topics for education purposes**

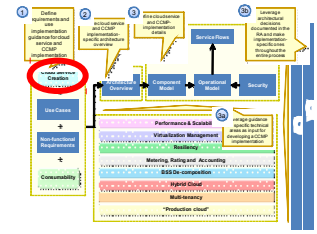


Using the CC RA: Which work products have to be created on a per cloud service basis?

- **The RA contains two types of work products**
  1. **Work products only serving as a reference documentation for specific aspects of implementing a new cloud: All "domain-specific" work products except "Cloud service creation"**
  2. **Work products serving as the basis for a cloud-specific work product: All "cross-domain" work products plus "Cloud Service Creation"**



Cloud Computing Reference Architecture Details



Cloud Service Creation

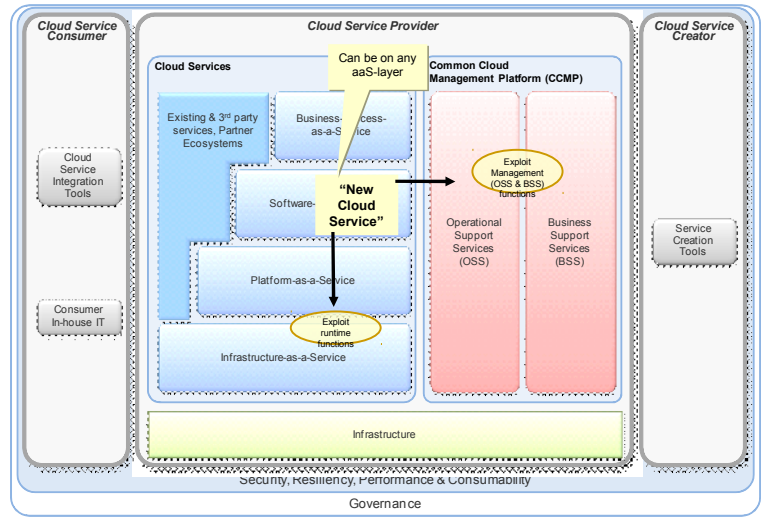
“Cloud Service Creation” – Scope and Overview

Objective:

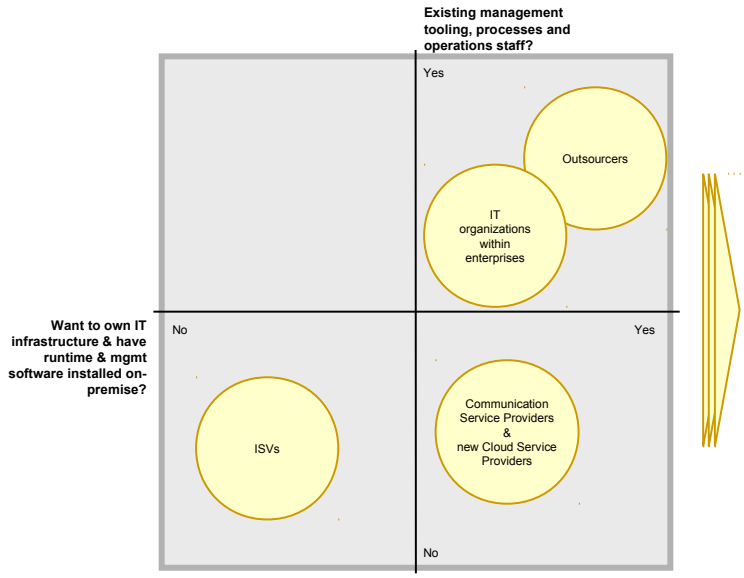
Describe all aspects around developing a new cloud service

- Scope is end-to-end process for developing a new cloud service.

  Focus is on the management aspects as they apply generically across all types of cloud services
- Provide detailed description of basic terminology relevant for creating cloud services
- Provide prescriptive product recommendations relevant for each cloud service creation step and tooling around that
- Provide illustrative examples (“applied patterns”) for how a new cloud service can be created



Developing cloud services – four major segments: Overview

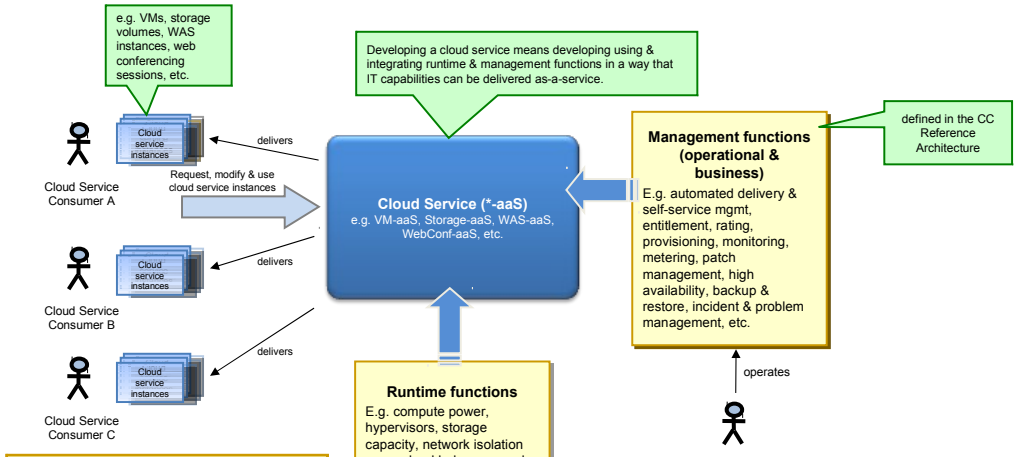


Although having different motivations, software & process legacy and requirements, there is a common approach for creating cloud services across all segments

Different scenarios for offering cloud services require
- different delivery models for management software ("build-it-yourself" vs. IBM-hosted)
- different levels of integration with existing management tools

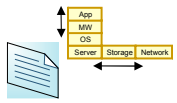
Anatomy of a Cloud service – Highlevel overview

The internal structure of a cloud service is the integration of all runtime and management functions used to deliver, run and manage instances of the respective cloud service.



12 steps towards creating a cloud service

1. Implement service definition

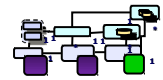


Describe function, price , SLA of cloud service, incl. management scope

2. Implement runtime functionality

Examples: Select off-the-shelf hypervisor (VMaaS), implement custom app (e.g. LotusLive)

3. Define unit of delivery & rating



Examples: VM, file system, distributed app, virtual IP address, queue, web conference, RDBMS, 3-tier business app, etc.

4. Implement self-service delivery & management functionality

Examples: "Create VM, add more nodes to WAS cluster, change max # of seats for LotusLive web conf

5. Implement monitoring metrics & event correlation rules

Select existing agent / implement new agent for monitoring JVM heapsize, hypervisor swap file size, # of processes, etc.

6. Implement incident, problem and asset mgmt processes

Incident, problem & asset mgmt process is specific to cloud service → customization needed

7. Implement resiliency SLA

Examples: HA for management system, delivered WAS cluster must be highly available

8. Implement backup approach

Examples: Backup all VMs, backup DB of LotusLive application

9. Implement security functions

Implement authentication, auditing, data protection, governance & audit

10. Implement cloud service specific billing metrics

Examples: CPU/hour, # of DB transactions, GB/month, # of users/webconf/hour, etc.

11. Implement rates for charging cloud service consumption

Examples: $0.11/VMhour; $0.19/MBsTransferred; $0.02/webconference; $0.05/fraudAnalysis

12. Register cloud service to service catalog

A cloud service must be registered to the service catalog to be externally accessible, entitlements need to be configured,

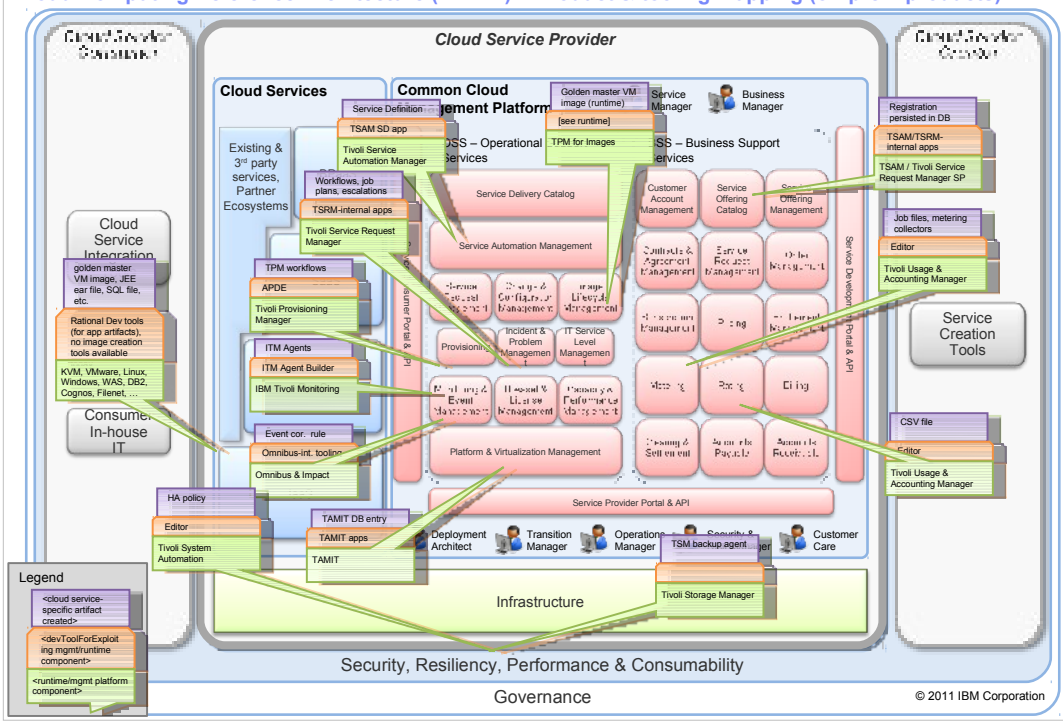
Overview: Recommended management products, to be exploited on a per cloud service basis

Step	Products & technologies to build your own cloud service
1. Specify cloud service description	n/a (cross-cutting)
2. Implement runtime functionality of the cloud service	KVM, VMware, Linux, Windows, WAS, DB2, Cognos, Filenet, …
3. Define Unit of delivery & rating	n/a (cross-cutting decision)
4. Implement self-service delivery & management functionality	TSAM
5. Implement monitoring metrics & event correlation rules	IBM Tivoli Monitoring, Omnibus, Impact
6. Implement incident, problem and asset mgmt processes	Tivoli Service Request Manager & TAMIT
7. Implement resiliency SLA	Tivoli System Automation
8. Implement backup approach	Tivoli Storage Mgr
9. Implement security functions	Tivoli Security / ISS portfolio
10. Implement cloud service specific billing metrics	TUAM
11. Implement rates for charging cloud service consumption	TUAM
12. Register cloud service to service catalog	TSAM / TSRM Service Provider Edition

Overview: Recommended management products, development tools and artifacts to be created on a per cloud service basis

Step	Building a cloud service hosted within the provider's premise		
	Products & technologies	Development/Config Tool	Cloud service specific configuration / artifact
1. Specify cloud service description	n/a (cross-cutting)	n/a	n/a
2. Implement runtime functionality of the cloud service	KVM, VMware, Linux, Windows, WAS, DB2, Cognos, Filenet, …	Rational Dev tools (for app artifacts), no image creation tools available today	VM golden master image, JEE ear file, SQL file, etc.
3. Define Unit of delivery & rating	n/a (cross-cutting decision)	n/a	n/a
4. Implement self-service delivery & management functionality	TSAM & TPM	TSAM SD app TPM ADPE	TSAM Service definition & TPM provisioning workflows
5. Implement monitoring metrics & event correlation rules	IBM Tivoli Monitoring, Omnibus, Impact	Universal Agent Builder (ITM) Omnibus-internal tooling	Monitoring agent (ITM), event correlation rule (Omnibus)
6. Implement incident, problem and asset mgmt processes	Tivoli Service Request Manager & TAMIT	TSRM-internal apps	Incident & problem management workflows, jobplans, escalations (TSRM)
7. Implement resiliency SLA	Tivoli System Automation	Text editor (no dedicated tooling)	HA policy
8. Implement backup approach	Tivoli Storage Mgr		TSM Backup agent Backup policy
9. Implement security functions	Tivoli Security & ISS Portfolio	Various	various
10. Implement cloud service specific billing metrics	TUAM	No dedicated tooling available	Metering collectors Job files
11. Implement rates for charging cloud service consumption	TUAM	UI and CSV file import	Rates persisted in TUAM database
12. Register cloud service to service catalog	TSAM / TSRM Service Provider Edition	TSAM/TSRM internal apps	Registration persisted in TSAM/TSRM internal database

*Cloud Computing Reference Architecture (CC RA) – Product & tooling mapping (on-prem products)

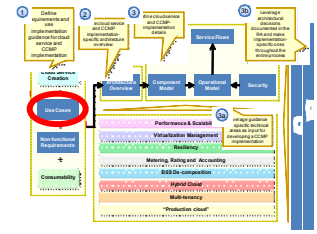


Cloud service creation – Applied patterns overview

- Walkthrough all 12 steps per applied pattern while describing implementation-specific decisions per step (incl. product selections)
  1. ISV
     - Example cloud service: “Desktop Cloud”
  1. “Enterprise”:
     1. WAS

- The applied patterns should serve as a guideline for anyone who wants to deliver IT capabilities as cloud services.
- It is acknowledged that many real-world implementations won’t map exactly to the applied patterns described here – many will be in a “grey zone” in between.
  However, the applied patterns serve as good guidelines and blueprints for these custom implementations

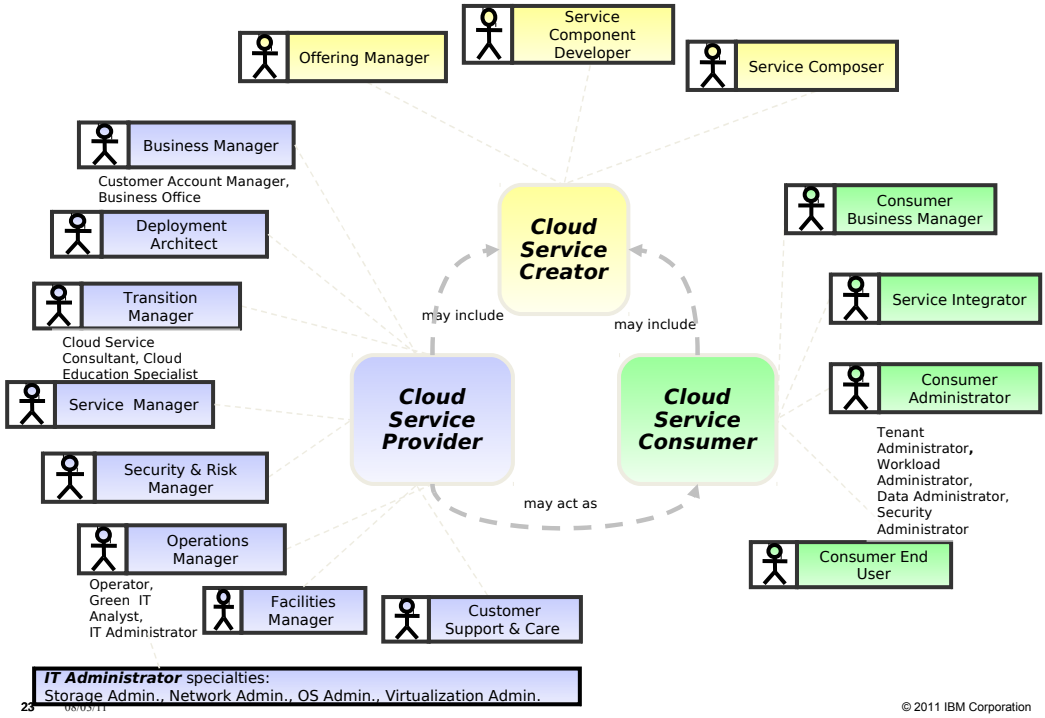
- There will be more applied patterns available in future versions of the RA



Use Cases & Roles

Cloud Computing User Roles V2 – Applied as Actors of CC RA use cases

IBM Cloud Computing Reference Architecture



Detailed Role Description - Example: Service Manager

IBM Cloud Computing Reference Architecture

Main Goal	▪ Services meet their quality objectives ▪ Get up-to-date information on service adoption in seconds not days ▪ Have meaningful, fact based discussions on service quality with internal and external suppliers
Main Responsibility	The focus of a Service Manager is to enable the smoothest service flow possible between all systems, from business support system to operational support system services. They ensure that the running service is well aligned with business and operational objectives and targets.
Main Tasks of their day-to-day job	**Service Setup** ▪ Provision service **Service Runtime** ▪ Managing compliance with service's Service Level Agreements (SLA) ▪ Monitor third party supplier SLA ▪ Detect service faults ▪ Determine service impact (from infrastructure faults / degradation) ▪ Prioritise problems based on network impact ▪ Manage long and short term performance targets ▪ Ensure alignment of business and operational support systems
Skills	▪ Expert knowledge of the existing cloud infrastructure and good judgment of the impact of a new or changed solution on it ▪ Proficient understanding of the particular service offering and its service instances ▪ Expert automation knowledge ▪ Proficient understanding of capacity/performance issues within and across systems
Working with which other roles ?	Operations Manager; Operator; Network Administrator; Customer Account Manager; Business Manager; Green IT Analyst; Deployment Architect; Cloud Service Creator

CC RA V2.0 Use Cases

- 38 use cases, grouped into 10 packages, along the cloud service life cycle

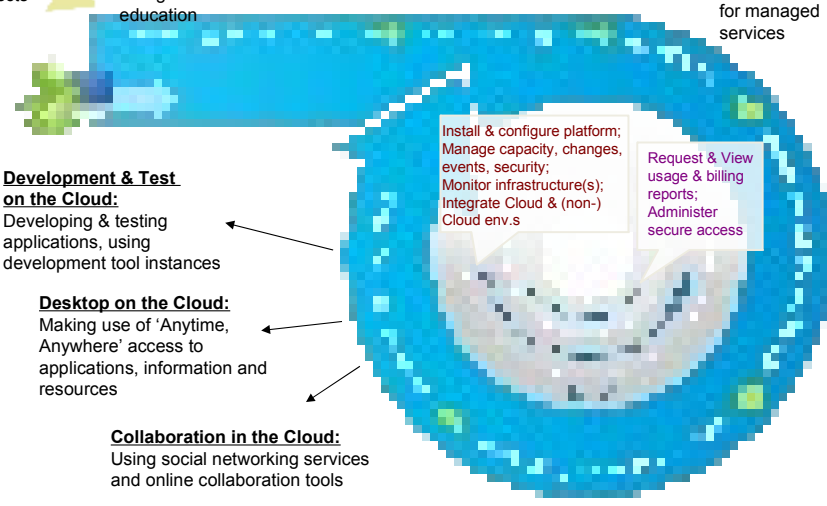
Creating & Managing Service Offerings

Defining the Scope, Creating & Managing Service Offerings, Creating & Managing Business Aspects

Consultative Analysis; Self-guided information & education

Via F2F or Web Sales channel

Enable customers for managed services



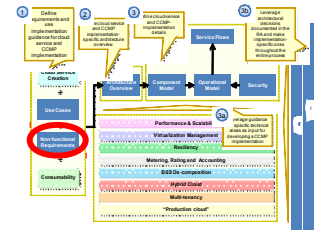
Development & Test on the Cloud:
Developing & testing applications, using development tool instances

Desktop on the Cloud:
Making use of 'Anytime, Anywhere' access to applications, information and resources

Collaboration in the Cloud:
Using social networking services and online collaboration tools

Create, modify, delete users; Create, modify, delete user groups

Request, view, update, delete service instances

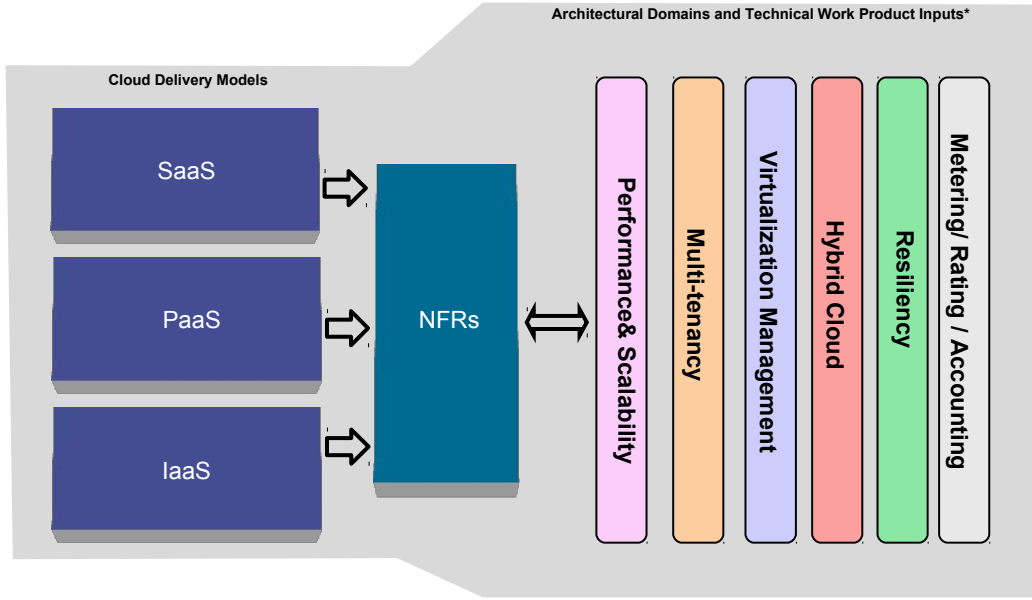


Non-functional requirements

Non-functional requirements TWP: Scope & Purpose

- Consolidate typical Cloud Computing Management Platform Non-Functional Requirements to illustrate the minimal set of non-functional requirements for building a Cloud Management Platform
- Reference architecture NFRs can help practitioners collect requirements during any project requirements gathering phase
- NFRs are key input to other TWPs including Component Model and Operation Model
- NFRs are high level in nature, as such, they should be evaluated along with the lower level CC RA Operational Model to establish the optimum Infrastructure for deploying the proposed solution
- Non-Functional requirements of a Cloud management platform are quality requirements or constraints of the platform that must be satisfied. These requirements address major operational and functional areas of the platform in order to ensure the robustness of the system.

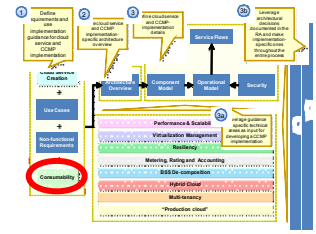
NFR Context Relationship with other TWP and Domains



Non-Functional Requirement TWP document contains the following detailed NFRs…

In version 2 of the NFR document we added NFRs from the following areas on top of the version 1:

- Applied Pattern NFRs Added
  - IaaS - Public Cloud: +20
  - IaaS - Private Cloud: +17
  - PaaS: +15
  - SaaS – Managed Service Desk: +16
- Domain Specific NFRs Added
  - Performance and scalability: +14
  - Multi-tenancy: +16
  - Virtualization management:
    - Base virtualization: +12
    - Network virtualization: +18
  - Hybrid Cloud: +10 NFRs
  - Resiliency: +13
  - Metering / Accounting: +24

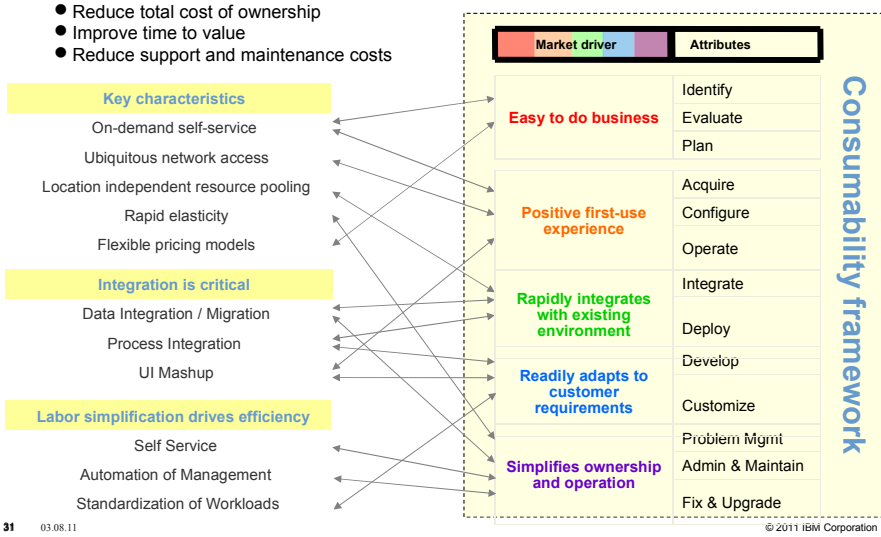


Consumability

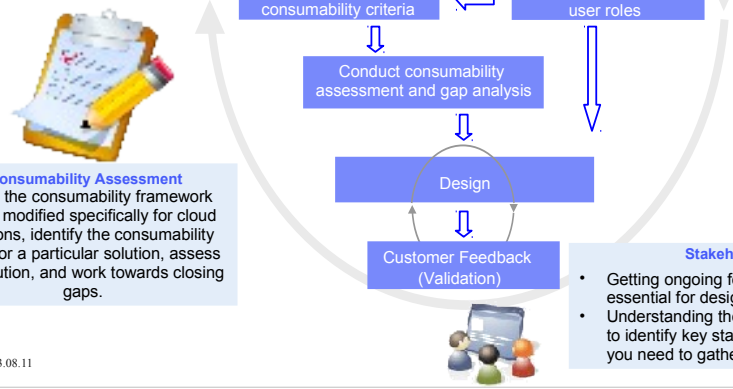
What makes a cloud solution consumable?

Consumability is a customer-centric term that describes the total customer experience with IBM products, solutions and services. Consumable solutions:

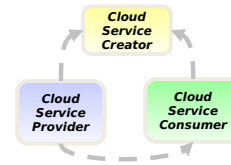
- Reduce total cost of ownership
- Improve time to value
- Reduce support and maintenance costs



Designing Consumable Cloud Solutions



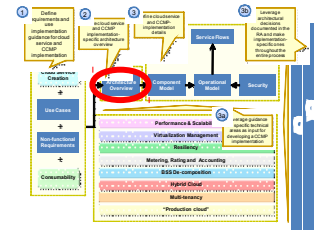
Consumability Assessment
Using the consumability framework that is modified specifically for cloud solutions, identify the consumability goals for a particular solution, assess the solution, and work towards closing gaps.



Cloud User Roles
Describing who is interacting with a cloud service at the different stages of the cloud life cycle

Stakeholder Feedback

- Getting ongoing feedback from stakeholders is essential for designing a consumable solution.
- Understanding the user roles, the next step is to identify key stakeholders whose feedback you need to gather



Architectural Principles & Overview

Cloud Computing Reference Architecture: Architectural Principles

An architectural principle is an overarching guideline or paradigm driving architectural decisions across the entire architecture process on a more granular level.

1. Design for Cloud-Scale Efficiencies ("Efficiency Principle"):

Design for cloud-scale efficiencies, and time-to-deliver/time-to-change metrics, when realizing cloud characteristics such as elasticity, self-service access, and flexible sourcing.

→ Overarching objective of Driving down costs (¢/ServiceInstanceHour) and time-to-response by orders of magnitude

2. Support Lean Service Management ("Lightweight Principle"):

Support lean and lightweight service management policies, processes, and technologies.

→ Radical exploitation of high degree of standardization in cloud environments to reduce management costs, based on an Eliminate-Standardize-Optimize approach

3. Identify and Leverage Commonalities ("Economies-of-scale Principle"):

Identify and leverage commonality in cloud service design.

→ Maximum sharing of mgmt components, infrastructure & infrastructure / platform cloud services across cloud services to reduce CapEx & OpEx and time-to-market

4. Define and Manage Cloud Services generically along their Lifecycle ("*Genericity Principle*"):

Define service templates and manage service instances generically along their lifecycle, across I/P/S/BPaaS.

→ Support I/P/S/BPaaS cloud services in a generic fashion, with a single management platform

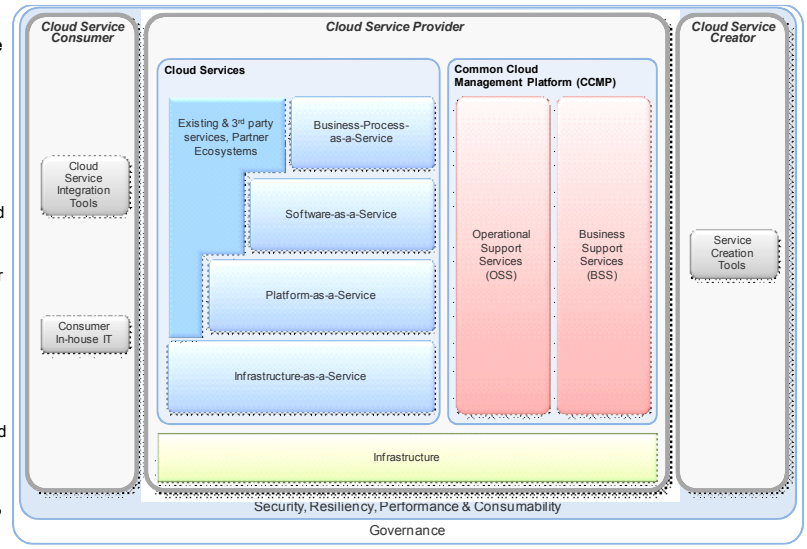
Cloud Computing Reference Architecture (CC RA) – Overview

CC RA architecture overview diagram lays defines basic elements of any cloud service environment

- Three main roles: Cloud service consumer, provider and creator whereas multiple roles can be fulfilled by the same organization/person
- Offered cloud services, required Management services (CCMP) and underlying infrastructure are defined.
- Next level of drill-down defined for each architectural element

CCMP is one element of the overall cloud computing reference architecture

- Defines operational and business support services *commonly* needed for delivering and managing any cloud service (I/P/S/BPaaS)
- Doesn't imply a single, monolithic implementation of a CCMP. CCMP is typically implemented by a well-integrated set of mgmt products



Cloud service model definitions

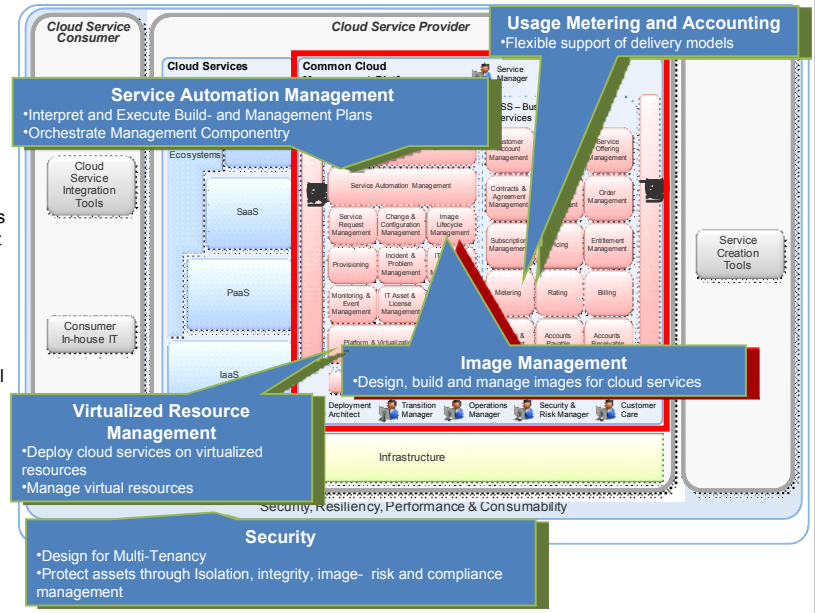
Term	Definition	Source
BPaaS	Business process services are any business process (horizontal or vertical) delivered through the Cloud service model (Multi-tenant, self-service provisioning, elastic scaling and usage metering or pricing) via the Internet with access via Web-centric interfaces and exploiting Web-oriented cloud architecture. The BPaaS provider is responsible for the related business function(s).	IBM[2]
SaaS	The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure and accessible from various client devices through a thin client interface such as a Web browser (e.g., web-based email). The consumer does not manage or control the underlying cloud infrastructure, network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.	NIST[1]
PaaS	The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created applications using programming languages and tools supported by the provider (e.g., java, python, .Net). The consumer does not manage or control the underlying cloud infrastructure, network, servers, operating systems, or storage, but the consumer has control over the deployed applications and possibly application hosting environment configurations.	NIST[1]
IaaS	The capability provided to the consumer is to rent processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly select networking components (e.g., firewalls, load balancers).	NIST[1]

Note: Across all cloud service models the definition is determined by the management scope covered by the provider. For example, in IaaS "*the consumer does not manage or control the underlying cloud infrastructure […]", in PaaS "the consumer does not manage or control the underlying cloud infrastructure, network, servers, operating systems, or storage […]", etc..* ***So this essentially about the tasks the operations staff of the provider takes on, it is not about the virtualization technology being used.*** *For example, it's possible to use hypervisor-level virtualization to realize PaaS, SaaS or BPaaS.*

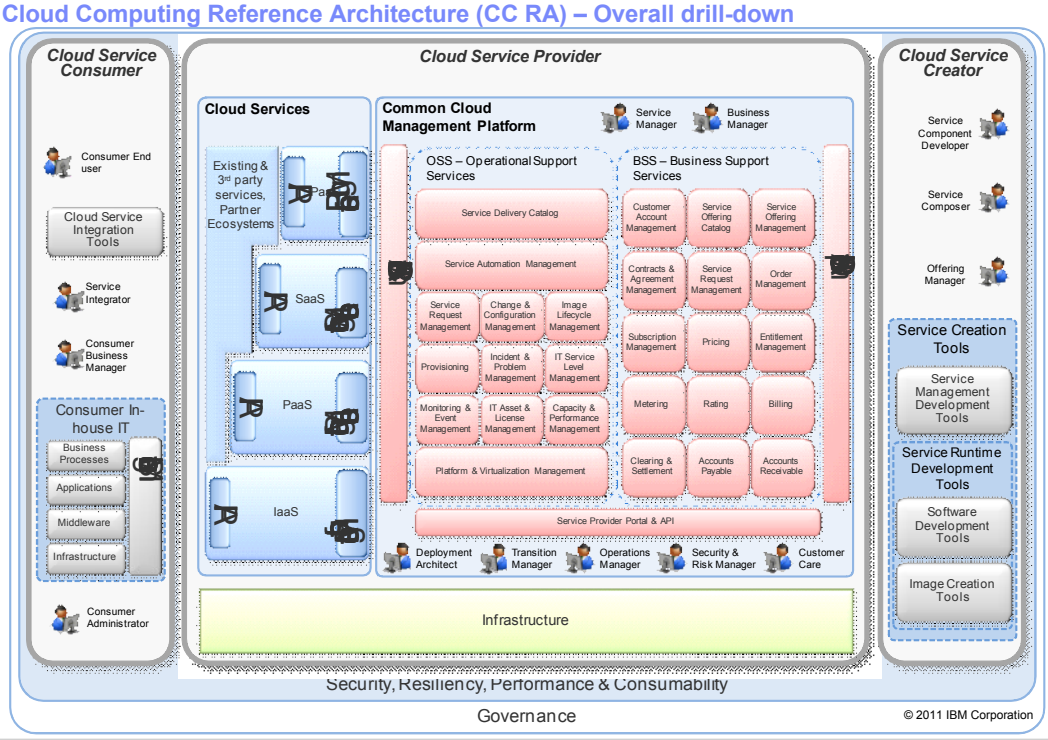
1. National Institute of Standards and Technologies; Draft NIST Working Definition of Cloud Computing, May 14, 2009
2. IBM MI and IPR definition bridge between Gartner and IDC, Aug 19, 2010

Cloud Computing Reference Architecture Overview Diagram – CCMP drill-down, highlighting some important topics

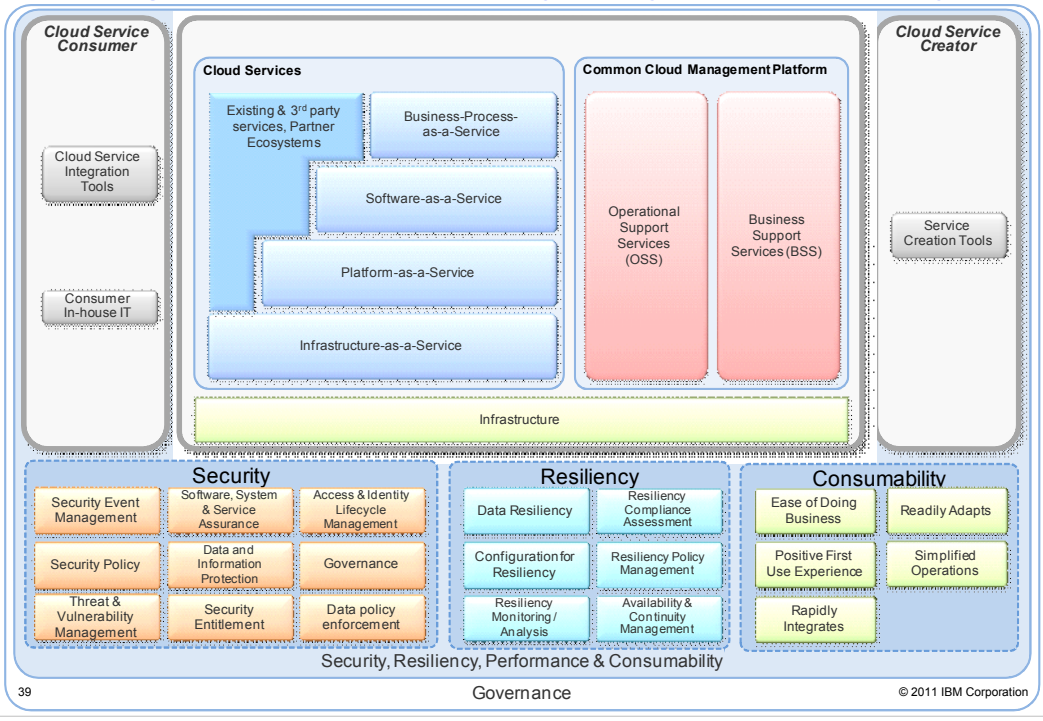
- CCMP defines all management functions commonly needed for the automated delivery & management of any cloud service
- Components are grouped in two major categories: Operational Support Services (OSS) and Business Support Services (BSS)
  - OSS: Responsible for managing the runtime components of cloud services
  - BSS: Responsible for all business / finance related aspects of cloud services
- Economies of scale can be achieved by managing multiple cloud services with the same set of mgmt components (see architectural principles)



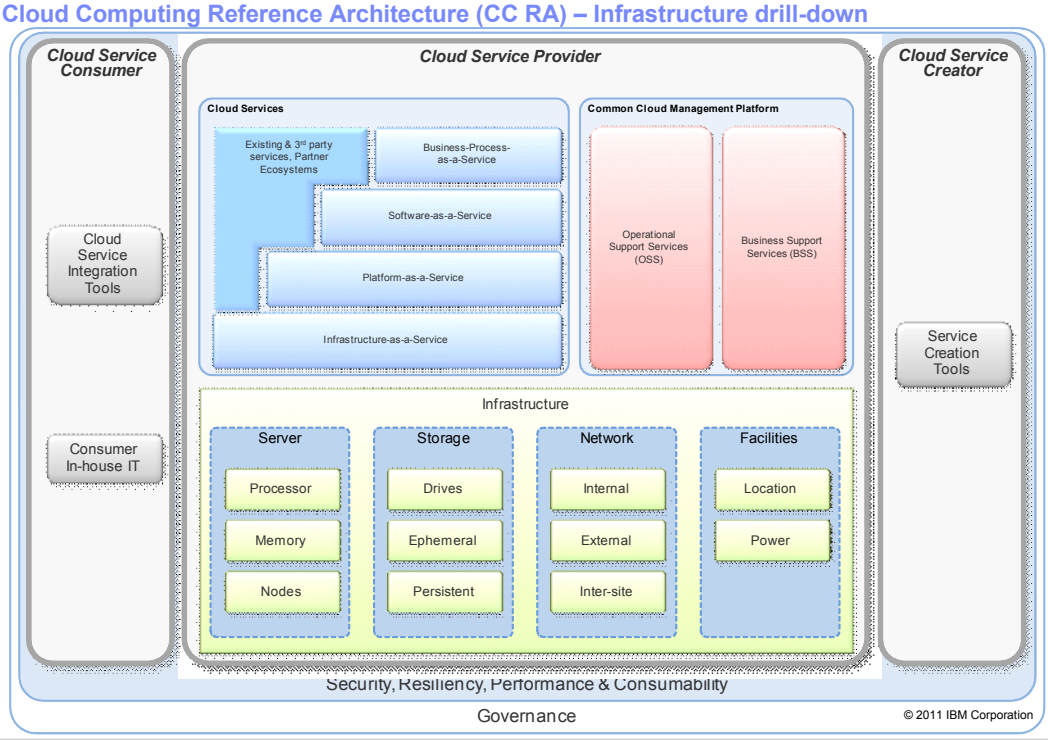
Cloud Computing Reference Architecture (CC RA) – Overall drill-down



Cloud Computing Reference Architecture (CC RA) – Security, Resiliency, Performance & Consumability drill-down

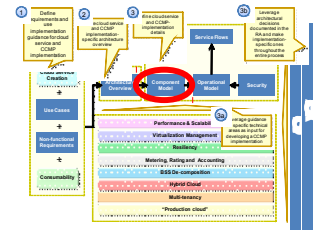


Cloud Computing Reference Architecture (CC RA) – Infrastructure drill-down



Architecture Overview: Applied patterns

- Applied Patterns are based on cloud projects where the reference architecture AOD has been used to create cloud implementations.
  Applied patterns reflect common scenarios of how a cloud could be implemented:
  1. **IBM Smart Business Development and Test on the IBM Cloud** is a sample for a "simple" (public) service provider architecture pattern. The diagram and related descriptions have been used to lead the high-level discussion and to guide decisions on scope for release 1 and following.
  2. The RA overview diagram has be used to discuss roles and responsibilities for a **Desktop cloud** project where part of the infrastructure is managed by IBM.
  3. An architecture overview diagram pattern for a **private enterprise where traditional IT is managed in parallel to a private cloud environment**, plus the consumption of externally provided public cloud services
  4. **A service provider "whitelabeling" architecture overview diagram pattern** (i.e. a service provider consuming cloud services from another cloud service provider and reselling the same offering -- with different branding and different prices)
  5. A **service provider consumption and value-add architecture overview diagram** pattern (i.e. service provider consuming 1..n cloud services from other cloud services, composing them and adding service provider specific value-add functionality on top)
  6. **Bio-Informatics Exchange**
  7. **IBM Cloud Service Provider Platform**
  8. **IBM Shared Private Production Cloud**

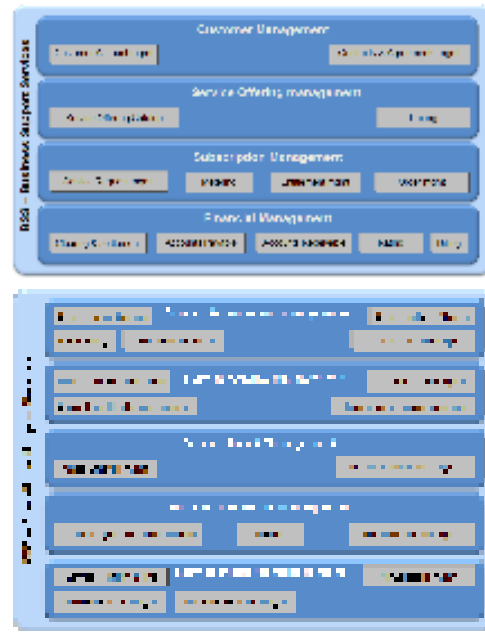


Component Model

CC RA 2.0: Component Model

Primary focus on BSS and OSS

- BSS and OSS component specifications and top-level decomposition
- Use Case realizations of important use cases (e.g. Requesting Service Instances)
- Extensive UML-based modeling of BSS and OSS component relationships and dependencies
- Cross-component view on CC RA 2.0 domains
- Mapping of current IBM cloud implementations to CC RA 2.0 Component Model
- Condensed mapping of CC RA 2.0 components with IBM recommended tools

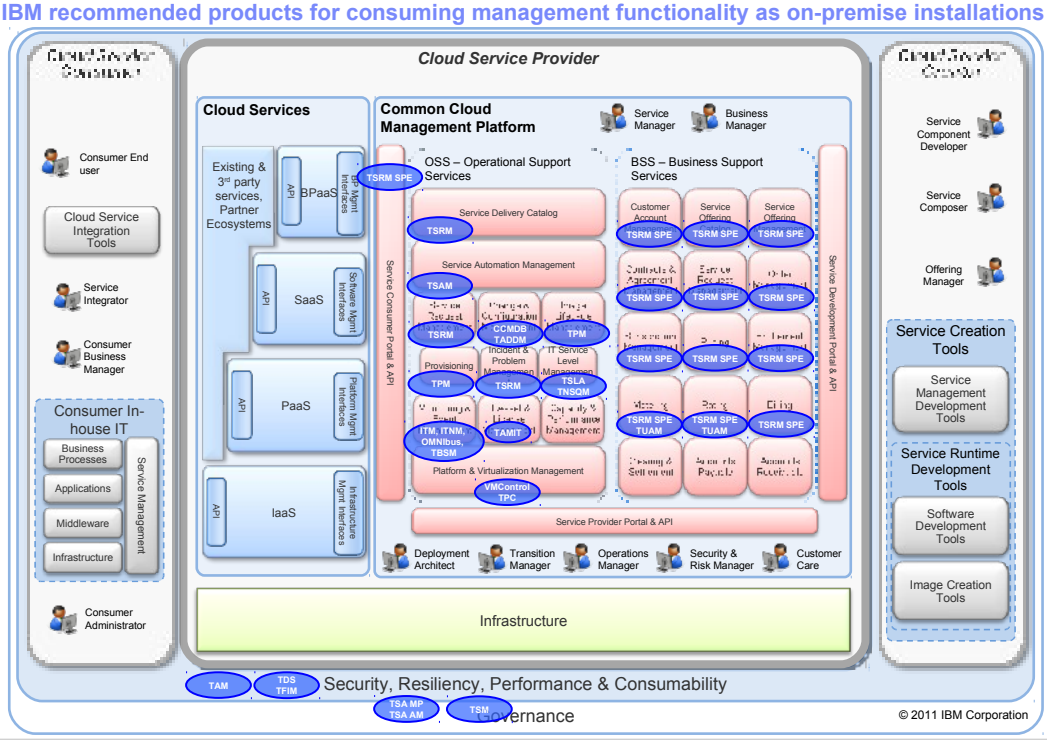


Use Case Realizations

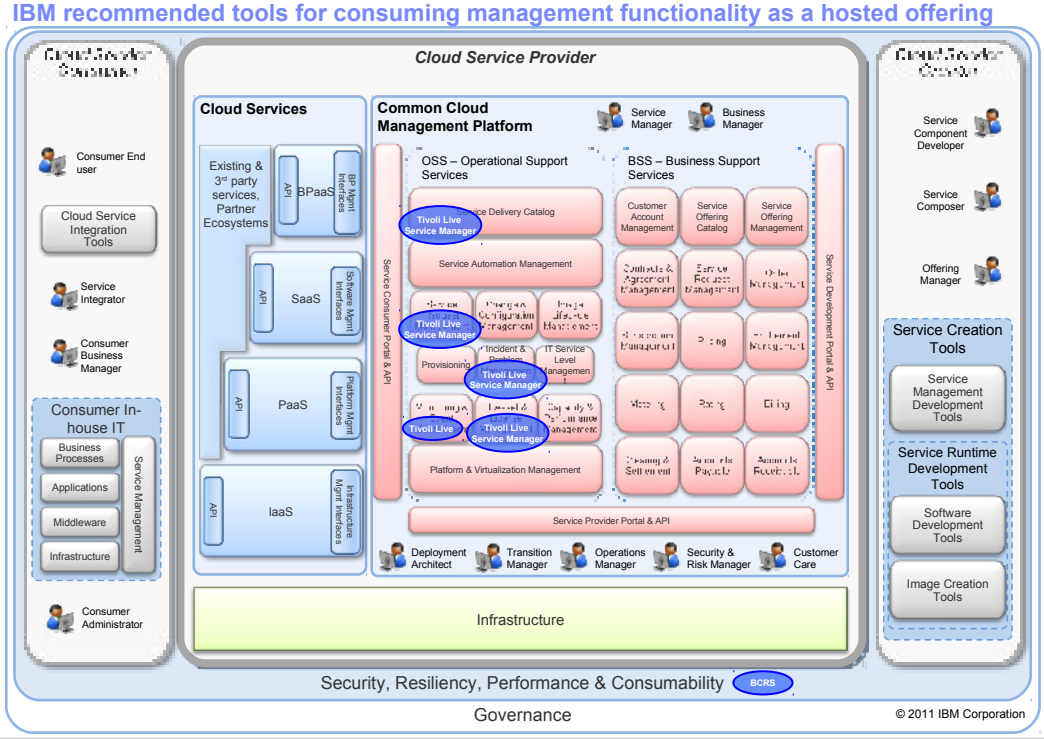
A use-case realization describes how a particular use case is realized within the design model, in terms of collaborating objects (IRUP).

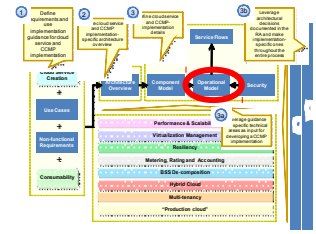
- CC RA 2.0 CM describes realization of typical use cases:
  - Creating Service Offering incl.
    - Create Service Template
    - Create Rating Scheme
  - Requesting Service Instance
- Sequence diagrams help to understand components, processes and flows involved in realization of use cases
- Detailed component diagrams are included to provide information about component interfaces, relationships and dependencies

IBM recommended products for consuming management functionality as on-premise installations



IBM recommended tools for consuming management functionality as a hosted offering



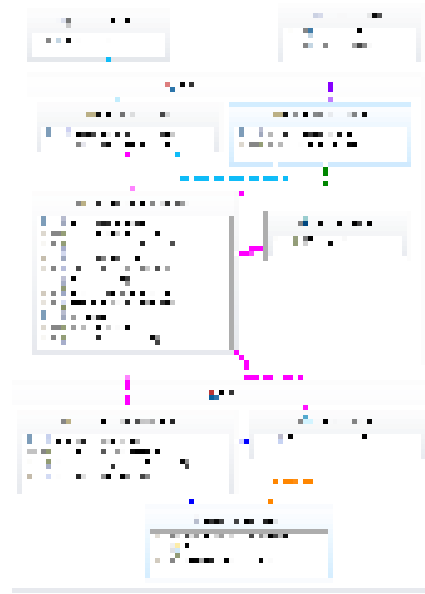


Operational Model

CC RA 2.0: Operational Model Updates

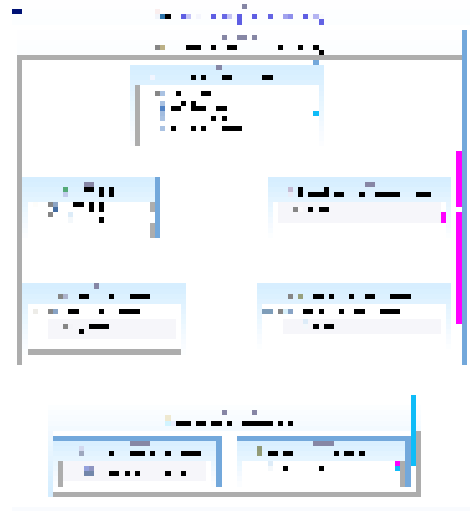
Primary focus on leveraging existing cloud implementations in IBM and other CC RA work products

- Show how components are to be realized and deployed in a cloud-specific fashion
- Harvest architectural patterns from existing cloud implementations (e.g. SBDTC)
- RSA models for each level of elaboration which can be leveraged by adopters
- System walkthroughs for major functional areas in Use Case Model and Component Model
- Implementation guidance on realizing CC RA using IBM recommended tools identified in Component Model
- Identify critical decision points and requirements used to guide the implementation



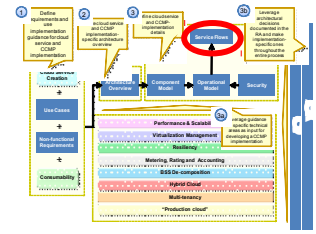
The Operational Model emphasizes how a Cloud deployment differs

- Adopter guidance on key decision points related to Cloud
- Addresses each node and component realization specifically, noting important considerations specific to Cloud
- Walkthroughs show how the component operates in the context of the CCMP



Operational Model includes guidance for adopters to enable application to engagements

- Implementation guidance includes:
  - Applied patterns showing how CC RA can be adapted to engagement-specific requirements
  - Recommended tool selection aligned with component mappings in the Component Model
  - Identification of architectural decisions and considerations of which practitioners need to be aware



Service Flows

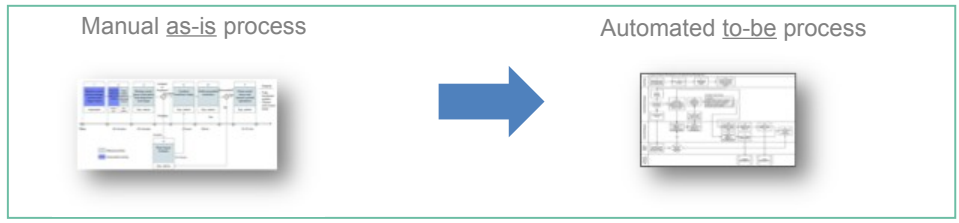
CC RA –Service Flows TWP: Scope & Purpose

- Defines all operational processes (“service flows”) required for managing cloud services based on a CCMP deployment.
  - These service flows are focused on reducing labor costs for management to a minimum, by building on high degree of standardization present in any cloud environment.
- Service flows depend on service management components as defined in the CC RA component model – CCMP components are required for automating as many tasks as possible.

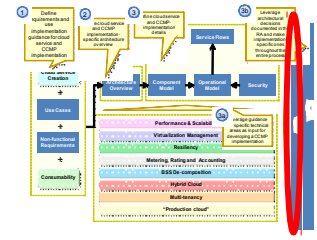
CC RA Service Flows Content: Overview

Service flows processes differences from standard IT management processes

- Leverages reduced (eliminate) scope of management, standardization, and opportunity for optimization to dramatically reduce labor costs.
- Aimed at delivering cloud management processes for cost-competitive cloud infrastructures



- Implementation of cloud-optimized service management processes
  - Configuration and Asset Mgmt
  - Patch / Provisioning / Image Mgmt
  - Incident / Problem Mgmt, Monitoring
  - Performance and Capacity Mgmt
  - Service Level Management / Metering
  - Service Request Management
  - Continuity Mgmt



Standards

Context and Scope of CC RA 2.0 Standards

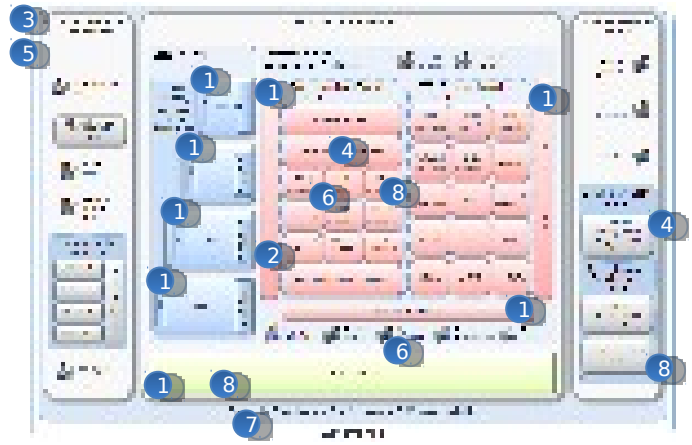
Context

By means of adhering to and driving the evolution of standards we **ease adoption** of our platform, assure **solution portability**, **avoid vendor lock-in**, and enable an **eco-system of partners** around our platform.

Scope

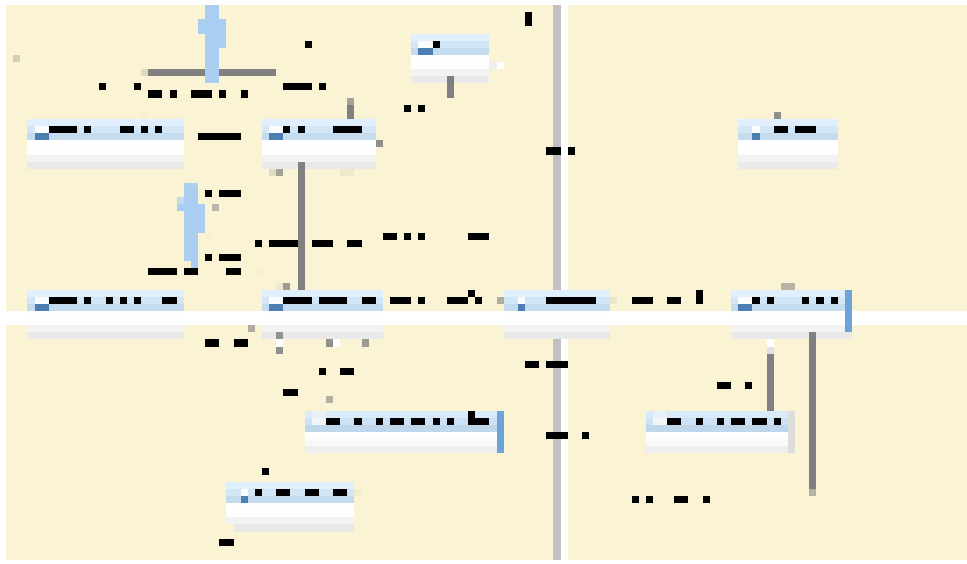
- Existing Cloud Computing and Management Standards
- Standards that are currently being defined
- Standards that need to be defined (recommended standards)

1. APIs
2. BSS UI Portal
3. Cloud Domain Model and Terminology
4. Cloud Management & Orchestration (Service
5. Consumability
6. Management Standards
7. Security
8. Virtualization (OVF, Images Mgmt., Network, Storage)



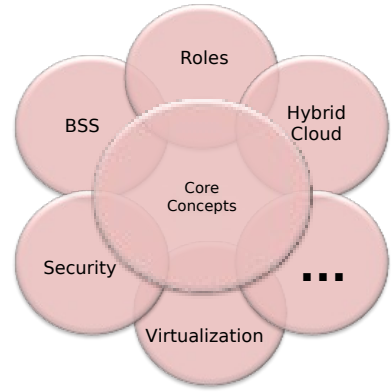
A standard can be defined as something with a pre-described specification, that is measurable, recognized as having authoritative value, and which an organization chooses to implement as a basis for 'good practice'.

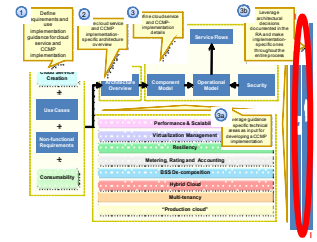
CC RA 2.0 Standards – Core Domain Model Overview



CC RA 2.0 Standards – Cloud Domain Model

- Create a central, authoritative domain model and terminology to be used consistently in cloud solutions
- Cover core concepts that are common across cloud solutions, and their touch points to specific disciplines
  - Do not dive into specific domains to avoid duplication of work
- Strong focus on (1) defining core concepts/terms and (2) defining how those concepts are related to each other
- Collect input from RA 1.0 and RA 2.0 documents, and make consistent with RA 2.0
- Make consistent with models that are defined or are being defined in IBM (e.g. CDM)



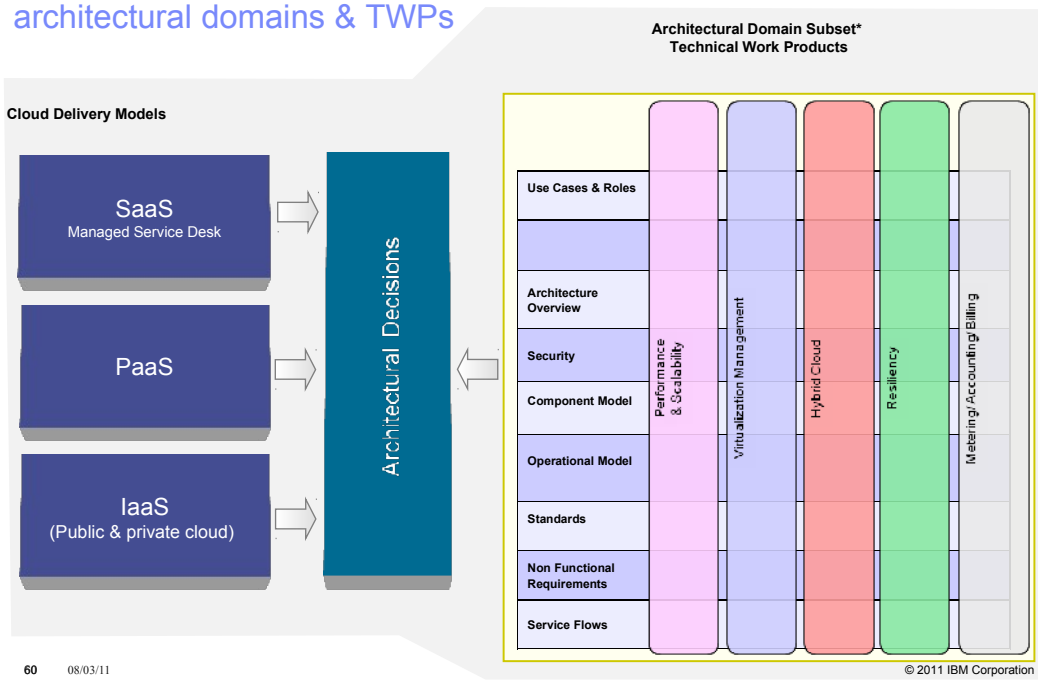


Architectural Decisions

Architectural Decisions TWP: Introduction & Purpose

- AD documents important decisions about any aspect of the architecture including the structure of the system, the provision and allocation of function, the contextual fitness of the system and adherence to standards
- Consolidate ADs from key offerings, architectural domains and TWPs to provide a single place to find important ADs that illustrate rationale and justification of architectural decisions
- Develop standard reusable artefacts that can be consumed by offering and engagement architects
- Prove a reference of documented decisions that can help practitioners to:
  - avoid unnecessary reconsideration of the same issues
  - preserve design integrity in the provision of functionality and its allocation to system components
  - ensure that the architecture is extensible and can support an evolving system

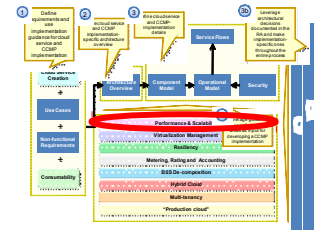
Scope Context - Capture ADs from key cloud offerings and architectural domains & TWPs



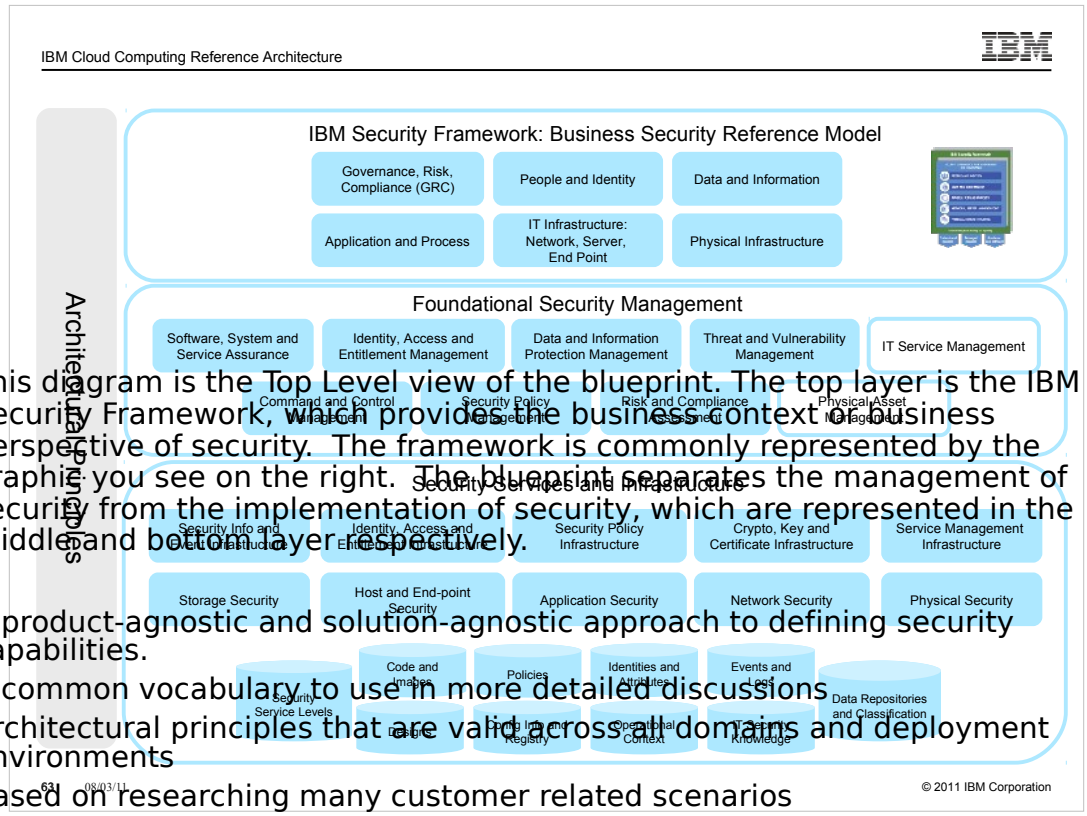
Architectural Decision TWP Word Document Contains the following detailed ADs…

In version 2 of the AD document we added decisions from the following areas on top of the decisions from version 1:

- Applied Pattern Architecture Decisions
  - IaaS - Public Cloud – 14 ADs
  - IaaS - Private Cloud – 18 ADs
  - SaaS – Managed Service Desk – 11 ADs
- Domain specific Ads
  - Hybrid Cloud – 7 ADs
  - Resiliency – 1 AD
  - Virtualization management – 17 ADs
  - Metering / Accounting – 4 ADs
  - Performance and scalability – 5 ADs



Security



This diagram is the Top Level view of the blueprint. The top layer is the IBM Security Framework, which provides the business context or business perspective of security. The framework is commonly represented by the graphic you see on the right. The blueprint separates the management of security from the implementation of security, which are represented in the middle and bottom layer respectively.

A product-agnostic and solution-agnostic approach to defining security capabilities.

A common vocabulary to use in more detailed discussions

Architectural principles that are valid across all domains and deployment environments

Based on researching many customer related scenarios

A roadmap to assist in designing and deploying security solutions

The security management layer represents the capabilities needed to translate the business view of security concerns into policies, operational procedures, and technical controls that can be deployed into the IT landscape and the organization. The Services and Infrastructures layer represents the security capabilities needed to enforce policies and their integration points into the IT infrastructure.

By separating security management from security implementation, the IT organization can focus on getting the policy and needed controls correctly defined and can better monitor and assess how completely and effectively the policies are being enforced.

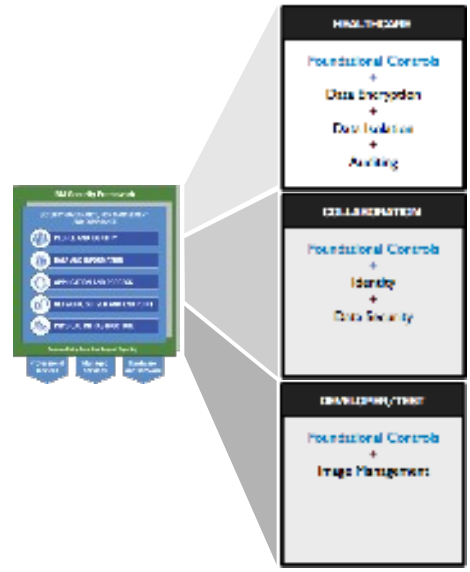
Architecture Principles in the Blueprint

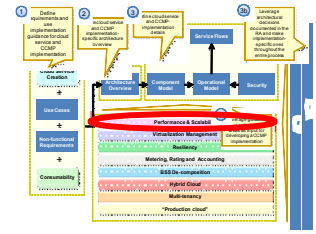
1. Openness
2. Security by default
3. Design for accountability
4. Design for regulations
5. Design for privacy
6. Design for extensibility
7. Design for sharing
8. Design for consumability
9. Multiple levels of protection
10. Separation of management, enforcement and accountability
11. Security is model-driven
12. Security-critical resources must be aware of their security context
13. Consistency in approaches, mechanisms and software components

Activity/Data Driven Cloud Security

Organizations need to adopt a strategy for cloud security that considers the unique attributes of the cloud as well as the activities and data the cloud is being utilized for.

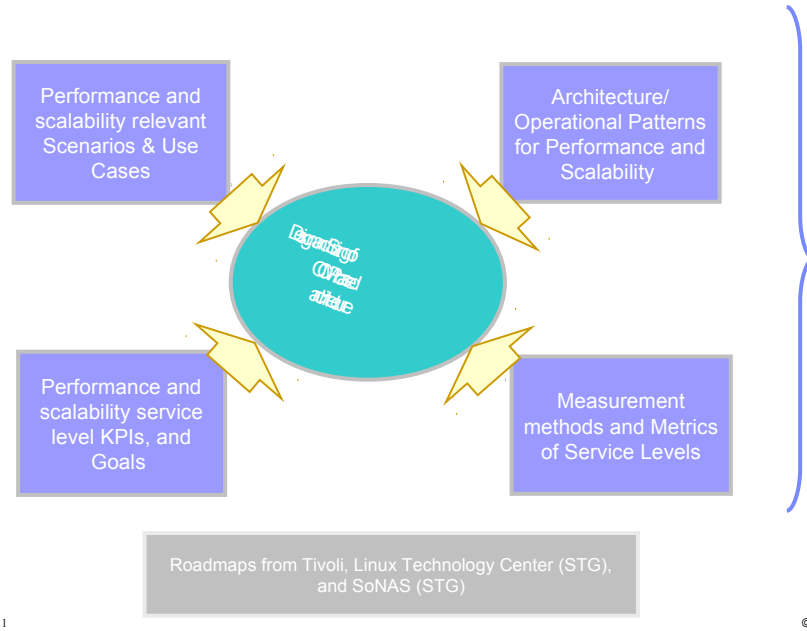
Only by combining foundational controls with activity/data specific controls can organizations meet their cloud security needs.





Performance & Scalability

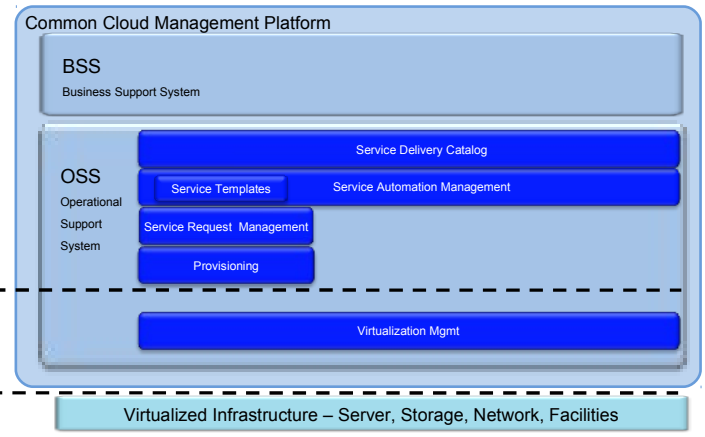
Performance & Scalability WG: Scope



Architectural Patterns

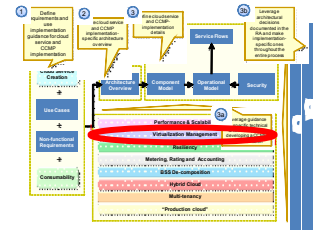
Initial focus on these CCMP Components

- Deployment patterns for OSS components
  - Multiple Virtualization Management domains
  - Horizontal Scaling of Provisioning components
  - Horizontal Scaling of Service Automation components
  - Multiple OSS domains
- High Scale Low Touch Virtualization Management
- Patterns and strategies for rapid provisioning
  - CoW boot disk of remote read-only images
  - Local caching of remote images and CoW root disk of local read-only image
  - Local caching of remote images and only transferring image delta
  - Use a combination of CoW, CoR, and Pre-fetching
  - Pre-create/hibernate/resume VM



Outline of the WP document

- 1 Introduction
  - 1.1 Scope of this document
  - 1.2 Supporting work products & references
  - 1.3 Legal Remark
- 2 Scenarios, Use Cases, KPIs, and Measurements
  - 2.1 Use Cases and Actors
  - 2.2 Influencing Scenarios and Factors
  - 2.3 Key Performance Indicators and Goals
  - 2.4 Measurements & metrics
- 3 Deployment patterns for CCMP OSS components
  - 3.1 Multiple Virtualization Management domains
  - 3.2 Horizontal scaling of Provisioning components
  - 3.3 Horizontal scaling of Service Automation components
  - 3.4 Multiple OSS domains
- 4 High Scale Low Touch virtualization management
  - 4.1 Architecture
  - 4.2 Functions
  - 4.3 Results
  - 4.4 Targeted scenarios and use cases
- 5 Virtual systems rapid provisioning strategies
  - 5.1 CoW boot disk of remote read-only images
  - 5.2 Local caching of remote images and CoW root disk of local read-only image
  - 5.3 Local caching of remote images and only transferring image delta
  - 5.4 Use a combination of CoW, CoR, and Pre-fetching
  - 5.5 Pre-create/hibernate/resume VM
- 6 Applied Pattern: Compute Cloud
- 7 References

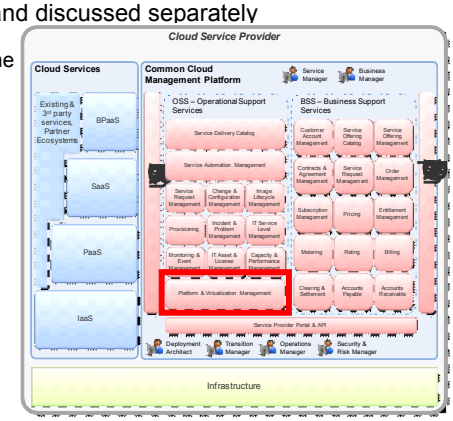


Virtualization Management – Base

Virtualization Management - Base

- Component of the OSS layer
  - provides common interface across hetergenous infrastucture & value-add functions above infrastructure (pooling, placement, mobility, notification up)
- Defines the capabilities and services for the management of virtual resources and the underlying physical resources
- Base virtualization divided into 2 main sub components
  - Platform Management
  - Virtual Resource Management
- Network and Storage Virtualization are sub teams and discussed separately

- Details what functions/subcomponents should be part of the Virtualization/Platform Management component
  - Definition of scope and functions
- What is important for implementation/options in cloud environments for these functions
  - Considerations and what options are there, pros and cons
  - Performance, scaling and architecture decisions
  - Ex: local storage vs shared storage the implications there wrt Performance, scaling, resiliency of the VMs
- What standards/recommendations are there for implementations



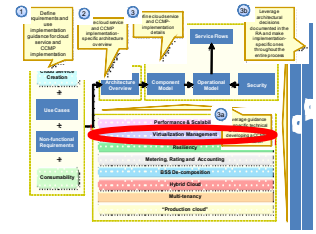
Virtualization and Platform Management

- **Virtualization Management**
  - Virtualization Platform Abstraction
    - Common interface to request and access virtual resources
    - Enables commonality for the higher level functions in the BSS and OSS layers
  - Virtual Resource Pooling
    - Ability to manage a set of virtualization infrastructure as a single entity or "pool"
    - Includes Compute, Network and Storage infrastructure
    - Virtualization management component handles placement/allocation across the pool and optimization of the pool.
    - Resource allocation can be based on a variety of criteria and factors such as energy, availability, licensing, etc
  - Reservation
    - Reserves virtual resources for future use
    - Provides a reservation interface
  - Image Repository
    - Library of virtual appliances which can be deployed to create workloads
  - Deployment and Undeployment
    - Creation and deletion of virtual systems and network/storage resources at the request of the provisioning layer
    - Provides for intelligent allocation across the pool, may include over commit of physical resources
  - Instance Image Management
    - Provides management of deployed instances of the images, including ability to capture images
  - Management of virtual servers
    - Provides interface to start, stop, restart virtual servers
  - Relocation
    - Provides ability to move a virtual resource from one physical machine to another
    - May happen as a result of a failure, predicted failure, or planned maintenance

- Platform Management
  - Discovery of physical resources
    - Library or catalog of discovered resources
    - Inventory such as options, firmware, OS
  - Platform Software Maintenance
    - Monitoring, reporting, and patching of system firmware and hypervisor software.
  - Bare Metal installation/distribution
    - Install and distribute OS and hypervisor software and agents
  - Platform health monitoring and reporting
    - Monitors health of physical systems and hypervisors
    - Used for billing services and SLAs

Virtualization and Platform Management

- Standards
  - Platform management
    - DMTF CIM, SNMP
  - Virtualization management
    - DMTF OVF
    - Libvirt
    - Activation engine - VSAE
- Non-Functional Requirements
  - Mostly centers around performance and scaling which is critical in a cloud environment
  - Details Documented in the NFR document
- Input provided to other work products for performance, resiliency, component model, etc
- Applied Patterns
  - Private cloud, Cloudburst, Telco scenario

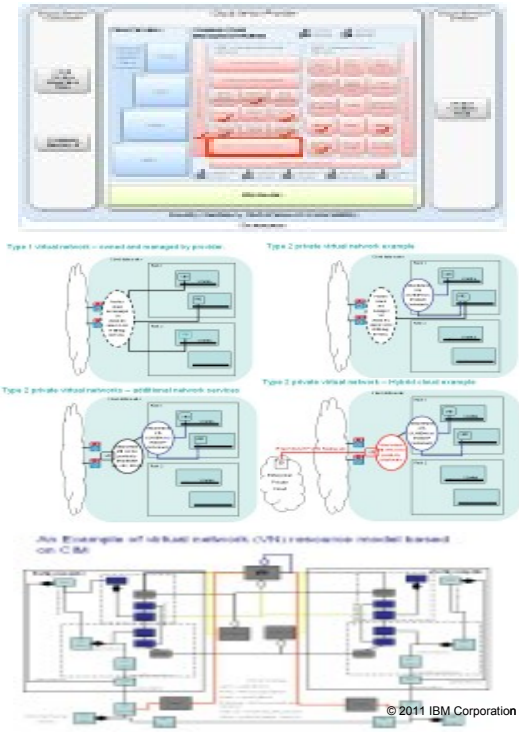


Virtualization Management – Networking

Virtualization Management – Networking Summary

IBM Cloud Computing Reference Architecture

- Identify BSS/OSS core components that are relevant to cloud (virtual) networking.
  - Capability extensions for virtual networking architecture and services.
- Identify a set of important cloud networking connectivity types (and patterns) to support cloud users (customers):
  - Shared cloud infrastructure network(s).
  - Private networks w/in cloud infrastructure (with variation of access over Internet).
  - Private cloud to cloud virtual network (including hybrid cloud – from customer's network to cloud).
- Identify/define resources in resource model (i.e., CIM), critical for supporting cloud networking as part of infrastructure authoritative topology DB.



Virtualization driving changes in network infrastructure, creating new opportunities for network management

Network requirements for virtualization and cloud are similar, especially for Enterprise/private clouds.

Hybrid Clouds and CSP Clouds have additional requirements for scaling, concurrent updates, etc. These are the focus of several startups.

We need to develop comprehensive capability for managing virtual networks, matching capabilities for physical networks

- Leverage work from ITM, Director.

-Lead in supporting optimization of new data center networks (e.g. Cisco FabricPath, TRILL, …)

Competitors (HP, Solarwinds)

Virtualization Management – Networking TWP Summary

- Define subcomponents (and their expected capabilities) and services of Networking Virtualization Management in Virtualization and Platform Management overall component.
- Identify Architecture Decisions and Non-Functional Requirements related to networking.
- Extend cloud use cases to include networking connectivity (i.e., connecting cloud services to virtual networks and automated network provisioning).
- Relate cloud networking to on-going standard work in DMTF SVPC WG for virtual networking within OVF.
- Identify IBM products/assets that provide foundations for building out support for cloud networking and automated network provisioning.

Virtualization driving changes in network infrastructure, creating new opportunities for network management

Network requirements for virtualization and cloud are similar, especially for Enterprise/private clouds.

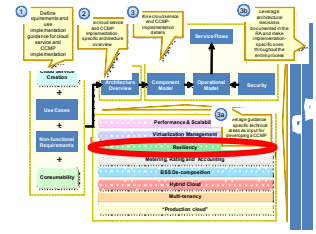
Hybrid Clouds and CSP Clouds have additional requirements for scaling, concurrent updates, etc. These are the focus of several startups.

We need to develop comprehensive capability for managing virtual networks, matching capabilities for physical networks

- Leverage work from ITM, Director.

-Lead in supporting optimization of new data center networks (e.g. Cisco FabricPath, TRILL, …)

Competitors (HP, Solarwinds)



Resiliency

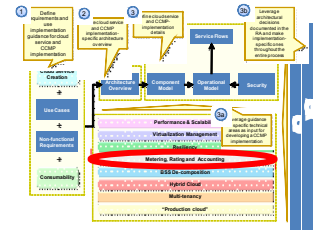
CC RA 2.0 Resiliency

CC RA 2.0 Resiliency

- Defined cloud Resilience architecture
- Based on lessons learned from implementations
- Mapped architecture to proposed CCMP resilience implementation
- Identified required products
- NFRs enhancements

Key learning from existing cloud implementations

- OSS/BSS interdependencies and operational requirements must be considered in coordinated resilience and automation design



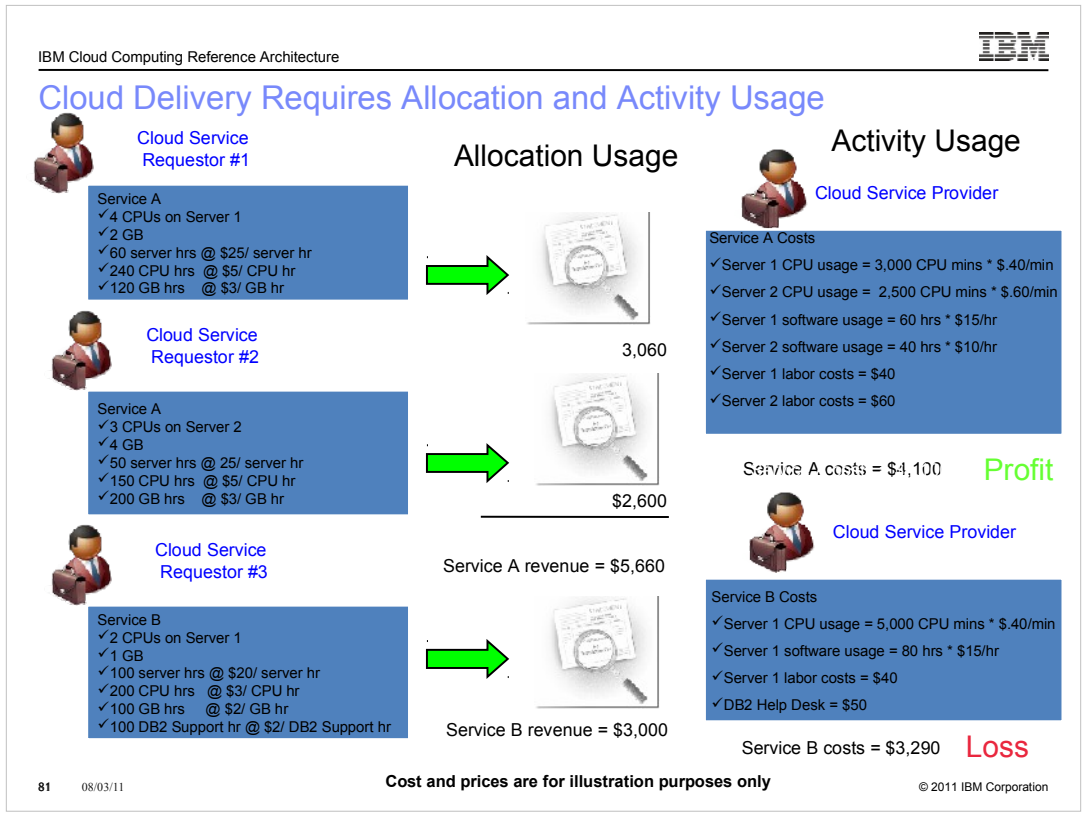
Metering, Rating and Accounting

Metering, Rating & Accounting: Executive Summary

- Metering, Rating and Accounting Domain addresses:
  - Different types of metered usage data (3 Key types) and the need to meter both Allocated and Activity based data
  - BSS-OSS mapping of runtime and design time data flows
  - Product specific detailed functional gap analysis with tool recommendations
  - Solution architecture along with implementation guidance
- 3 Applied Patterns from architecture as applicable to Public, Private clouds and an ISV context are explained

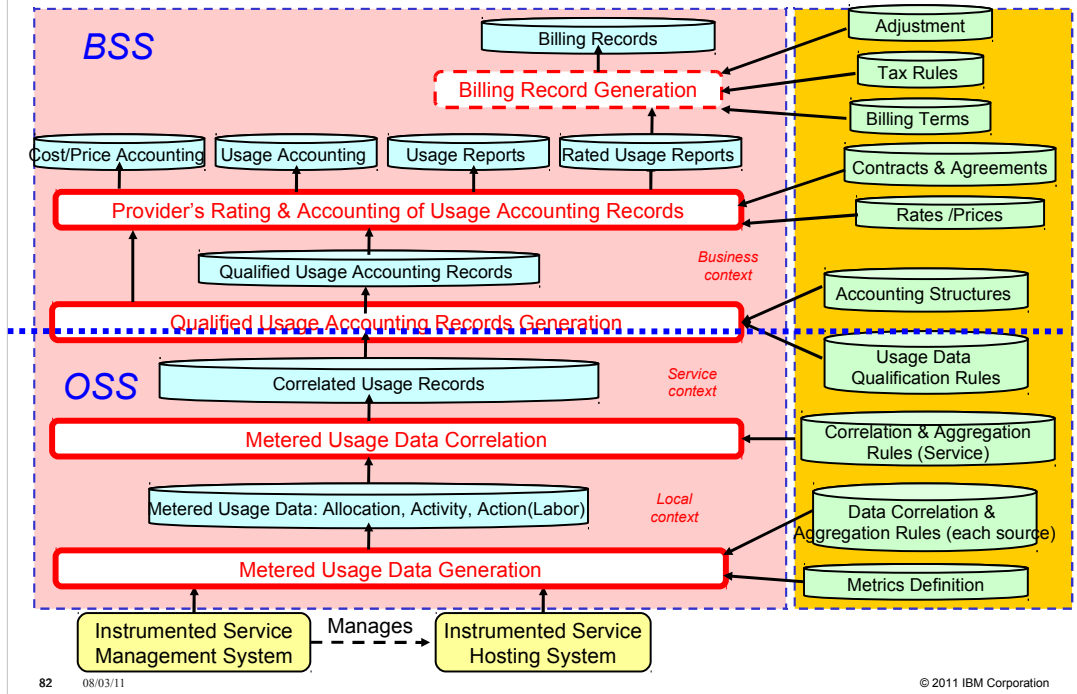
Terminology: Metered Usage Data Concepts

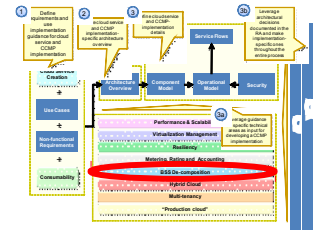
- Allocation Usage
  - The result of a consumer occupying a resource such that other consumers cannot use it
  - For example, the time period IT infrastructure topology (e.g. Servers, CPUs, Memory, Storage, Network, Database, Websphere Cluster) has been allocated to a particular cloud service.
  - More suitable as Service Usage Metric.
- Activity Usage
  - The result of activity performed by the consumer e.g. CPUSecs, Bytes transferred etc
  - More suitable as Cost Usage Metric.
- Action Usage
  - Actions initiated by the consumer that the provider may wish to charge for or track costs against
    - e.g. Backup/restore server, change virtual server configuration
  - Action may or may not involve manual steps



1. To determine if the service is Profit or Loss
2. To Maximize Cloud Utilization
3. Network bandwidth type service metrics require to measure usage

Metering Rating & Accounting: BSS – OSS Mapping

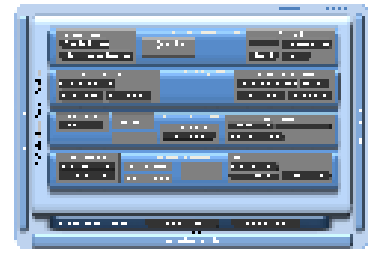




BSS Decomposition

Business Support Services Decomposition

- Business Support Services (BSS) is a set of components that enables interaction of customers with the service provider. BSS works closely with Operational Support Services (OSS), which is a set of computer systems and related processes used to deliver the services. BSS and OSS functions for all IBM Cloud services are provided by the Common Cloud Management Platform (CCMP)



- BSS Decomposition RA content - 44-page doc with the following sections:
  - BSS positioning within CCMP
  - Overview of BSS functions required for all realizations
  - Public Cloud realization
  - Private Cloud realization
  - Brief discussion of hybrid cloud and product development realizations
  - Future direction – BSS as a Service for promoting re-use and modularization

CCMP RA 2.0 conceptual CM for Business Support Services

Private Cloud Realization - using Tivoli Service Request Manager – Service Provider Edition + Tivoli Usage and Accounting Manager

TSRM-SP provides all these functions

Notes on METERING

TSRM provides Allocation & Activity-based Metering.

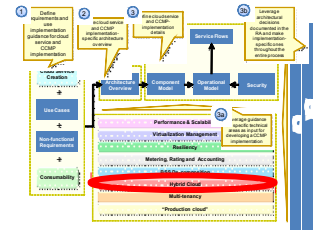
TSRM tracks labor, materials, tools, services costs.

TUAM provides Usage-based Metering

RATING & BILLING

TSRM & TUAM provide partial rating functions & bill data generation – but not direct invoicing

NO Peering and Settlement
NO accounts payable/receiving

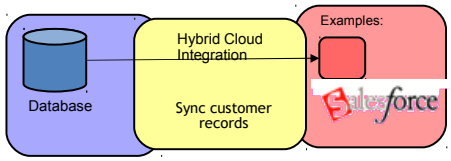


Hybrid Cloud

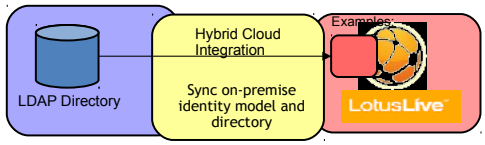
Hybrid Cloud Workstream: Scope and Dimensions

- **Scope and Purpose:**
  - **Use Cases:** Identify use cases and scenarios for hybrid cloud setup, operations, and management
  - **Patterns:** Identify solution patterns for integration of on-premise with services in public cloud(s)
  - **Life-cycle:** Identify and define workload migration and life cycle events for services in hybrid cloud
  - **Roles:** Identify roles associated with hybrid cloud operations and services
  - **Decisions:** Define architectural decisions for hybrid cloud integration framework and for hybrid cloud management services
- **Perspectives**
  - **Operating Perspective:** Seamlessly move peek workloads from on-premise to public Cloud
  - **Sourcing Perspective:** Different types of workloads to be provisioned by the most effective Cloud from the perspective of cost, functionality, availability, performance, security, etc.
  - **Management Perspective:** Unified view and capability to manage resources and information on-premise and in off-premise Clouds combined with management and integration of workloads and resources across hybrid cloud
- Dimensions:
  - Integration: How to connect on-premise services and data to off-premise counterparts… business data mapping and service integration
  - Security: How to integrate on-premise/off-premise identities, policies, auditing systems; how to ensure proper security of off-premise cloud workload; How to secure management and payload interactions
  - Monitoring: Integrate monitoring of off-premise infrastructure and applications with on-premise management system; Enable on-premise monitoring and event infrastructure to reach into clouds
  - Management: Manage Capacity in the cloud; provisioning- and de-provisioning based on Monitoring data), capacity overflow from on-premise to Cloud
  - Governance: Who can, does, should use which cloud-based services…service request management of on- and off-premise resources

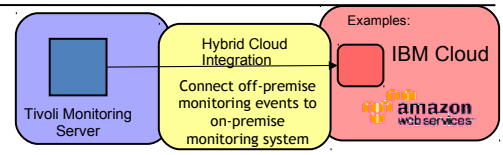
Hybrid Cloud: Use Case Scenarios



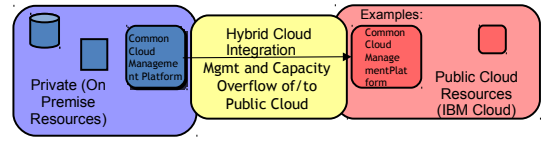
On-premise to off-premise Business Application Integration:
Example:
On-premise Database to Salesforce Cloud



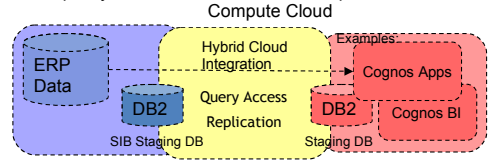
Security: Directory Integration & Identity Federation
Example
Synchronize on premise ODW LDAP and LotusLive
Domino directory info and facilitate SSO



Hybrid Monitoring
Example
Federate Monitoring info of Workload in IBM Public Cloud



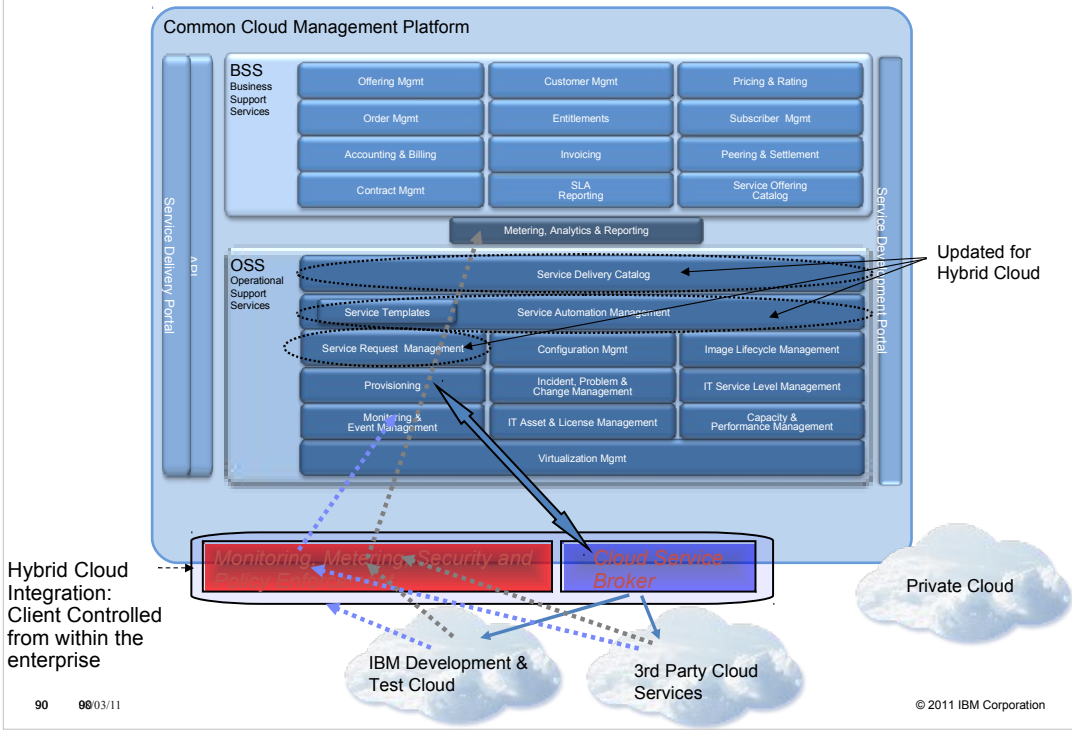
Governance & Hybrid Workload Management
Examples:
Governance for acquiring Resources from IBM Compute Cloud and/or Amazon; Workload Management and capacity overflow from CCMP based private Cloud to IBM Compute Cloud

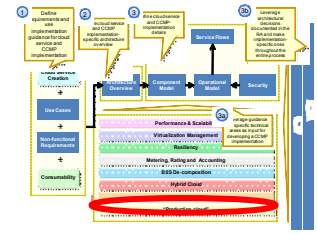


Cognos and other Analytic Applications in the Cloud:
Initial Load to Cloud; Change Data Capture Replication to & from Cloud; Query Access; Data Cleansing

* Cloud Management Platform – Hybrid Cloud Model

IBM Cloud Computing Reference Architecture





Multi-tenancy

Multi-tenancy - scope

- Done
- Partial

- Multi-tenancy requirements and customer expectations
- Requirements mapping to CC RA component for impact and gap analysis
- Multi-tenant patterns
- Leveraging existing multi-tenancy assets and capabilities
- Specification of how secure Multi-Tenancy should be implemented across the CC RA components

Multi-tenancy deliverables

Multi tenant NFRS	adopted	NFR deliverable
Use Cases and extensions	adopted	Use case deliverable
Component impact analysis	adopted	Component deliverable

Multi-tenancy definition

Tenant definition

Multi-tenant patterns whitepaper

Multi-tenant definition

Multi-tenancy refers to the ability of services to be offered to multiple user entities (tenants) in a way so that each tenant operates as logically isolated, while, in fact, using physically shared resources.

Tenant definition

A business entity that is contractually responsible for the consumption of Cloud Service from a service provider.

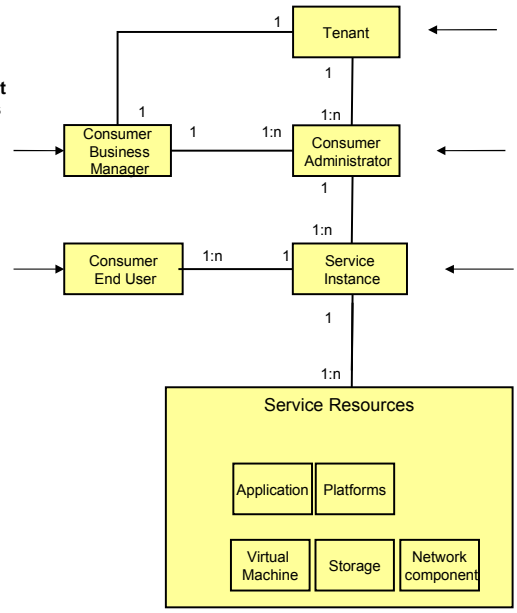
A role within the business entity that manages contracts and optionally approves commission of services from the Cloud

A role within the business entity that will request commission and decommission services from the Cloud.

The users of the Service

The Cloud service instance requested by the Service Owner

The resources consumed by the service



Multi-tenancy whitepaper

1. **Primitive set of patterns of multi-tenancy**
2. **Associated multi-tenant characteristics that will have to be supported in any Cloud implementation.**
3. **Multi-tenancy of managing and manage environments**
4. **Multi-tenancy needs of public and private clouds.**