Use Cases and Interactions for Managing Clouds

A White Paper from the Open Cloud Standards Incubator

Version: 1.0.0
Status: DMTF Informational
Publication Date: 2010-06-18
Document Number: DSP-IS0103

CONTENTS

Figures

Tables

Foreword

The *Use Cases and Interactions for Managing Clouds* white paper was prepared by the Open Cloud Standards Incubator of the DMTF.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability.

This living document is specifically targeted to assist the Cloud Incubator in being successful across transitions in leadership, charter adjustments, and membership.

Acknowledgments

The Open Cloud Standards Incubator wishes to acknowledge contributions from the following people:

- Kane, Valerie – Advanced Micro Devices
- Vancil, Paul – Advanced Micro Devices
- Kowalski, Vincent – BMC Software
- Lipton, Paul – CA, Inc.
- Moscovich, Efraim – CA, Inc.
- Waschke, Marvin – CA, Inc.
- Armstrong, Joseph – Cisco
- Sankar, Krishna – Cisco
- Serr, Frederick – Cisco
- Siefer, Mike – Cisco
- Tomsu, Peter – Cisco
- Wheeler, Jeff – Cisco
- Pardikar, Shishir – Citrix Systems Inc.
- Sirjani, Abolfazl – Citrix Systems Inc.
- Landau, Richard – Dell
- Ericson, George – EMC
- Linn, John – EMC
- Durand, Jacques – Fujitsu
- Song, Zhexuan – Fujitsu
- Aguiar, Glaucimar – Hewlett-Packard Company
- Cook, Nigel – Hewlett-Packard Company
- Maciel, Fred – Hitachi, Ltd.

- Baskey, Michael – IBM
- Johnson, Mark – IBM
- Pendarakis, Dimitrios – IBM
- Cox, Billy – Intel Corporation
- Li, Hong – Intel Corporation
- Raghu, Rekha – Intel Corporation
- Parchem, John – Microsoft Corporation
- Carter, Steve – Novell
- Jose, Jaimon – Novell
- Carlson, Mark – Oracle
- Vambenepe, William – Oracle
- Yu, Jack – Oracle
- Owens, Ken – Savvis
- Qualls, Richard – Sun Microsystems, Inc.
- Norbeck, Don – SunGard Availability Services
- Ronco, Enrico – Telecom Italia
- Galán Márquez, Fermín – Telefónica
- Bumpus, Winston – VMware Inc.
- Lamers, Lawrence – VMware Inc.
- Matthews, Jeanna – VMware Inc.

Use Cases and Interactions for Managing Clouds

1 Scope

This document is one of two documents that together describe how standardized interfaces and data formats can be used to manage clouds. This document focuses on use cases, interactions, and data formats; the other document focuses on the overall architecture.

The scope of this document includes:

- Management use cases across the entire lifecycle of a cloud service. The use cases are examples and not a complete list of all use cases related to managing clouds. For example, use cases for federating and brokering among clouds are not included.
- Interaction sequences between consumers, developers, and providers to implement the use cases
- Data artifacts exchanged in the interaction sequences

The following aspects of clouds are specifically outside the scope of this document:

- Prescriptive definition of required use cases
- Formal definition of a data model, or any associated database schema

The companion to this document, *Architecture for Managing Clouds* (DSP-IS0102), is a white paper that describes other aspects of managing clouds. That document describes:

- Reference architecture for managing clouds. This reference architecture was first introduced in the DMTF white paper, *Interoperable Clouds* (DSP-IS0101). The concepts are further explored and described in this document.
- Requirements for the architected interfaces in general, including requirements on the protocols, resource model, and security mechanisms
- Role of policies and constraints. Given the focus on the interfaces for managing clouds rather than the internal details of a cloud implementation, a useful abstraction is to define the desired capabilities of the cloud using policies and constraints. These are interpreted to be *what* the cloud service provider should provide rather than *how* it offers it.
- Patterns of consumer/provider interactions that repeat across many use cases with different operations and data payloads, depending on the use case
- Example of how a cloud service provider may implement the management interfaces, along with a discussion of design consideration. It should be emphasized that this section is not prescriptive; rather, it is illustrative.

2 References

DMTF DSP0243, *Open Virtualization Format Specification 1.1*, http://www.dmtf.org/standards/published_documents/DSP0243_1.1.pdf

DMTF DSP-IS0101, *Interoperable Clouds, A White Paper from the Open Cloud Standards Incubator 1.0*, http://www.dmtf.org/standards/published_documents/DSP-IS0101_1.0.pdf

DMTF DSP-IS0102, *Architecture for Managing Clouds, A White Paper from the Open Cloud Standards Incubator 1.0,* http://www.dmtf.org/standards/published_documents/DSP-IS0102_1.0.pdf

ITIL® V3 Glossary, *Glossary of Terms, Definitions, and Acronyms*, v3.1.24, 11 May 2007, http://www.best-management-practice.com/gempdf/ITIL_Glossary_V3_1_24.pdf

NIST Definition of Cloud Computing, http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc

3 Terms and Definitions

For the purposes of this document, the following terms and definitions apply.

3.1
cloud service
a publicly available service or a private service that is used within an enterprise

3.2
cloud service developer
designs, implements, and maintains service templates

3.3
cloud service consumer

- approves business / financial expenditures for consumed services
- accounts for used service instances
- requests service instances and changes to service instances (typically on behalf of the consumer business manager)
- provides access to services for service users

3.4
cloud service provider
an organization that supplies cloud services to one or more internal or external consumers

3.5
data artifacts
as used in this document, the control and status elements exchanged across the provider interface using the Infrastructure

3.6
deployment
the process of creating a service instance in a reserved or prepared environment

3.7
Infrastructure as a Service
IaaS
the capability provided to the consumer is to provision 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 limited control of select networking components (for example, host firewalls). (Source: *NIST Definition of Cloud Computing*)

3.8

notification

a signal from the cloud service provider to the cloud service consumer that a condition exists that requires attention

A notification can be either polled or sent asynchronously.

3.9

profile

a specification that defines how a set of standards, specifications, or models apply to a specific management domain

In the context of this document, a profile may refer to DMTF CIM models and associated behavior and may also refer to other documents as well.

3.10

provider interface

the interface through which cloud service consumers access and monitor their contracted services

The interface covers SLA negotiation, service access, service monitoring, and billing. This interface is also the interface through which a cloud service developer interacts with a cloud service provider to create a service template that is added to the service catalog.

3.11

provisioning

the process of selecting, reserving, or creating an instance of a service offering

Service offerings are selected from the service provider's service catalog and are then provisioned into service instances.

Provisioning is also the process of selecting or reserving service resources from available pools, assembling them together, and configuring them, based on a specific request in the contract, in order to fulfill the contract.

For example, a server instance can be created from a template; assigned CPU, memory, storage, and network resources; and configured for the consumer to satisfy the contract requirements (apply patches, adjust security, configure firewall, and so on).

3.12

security manager

responsible for managing the credentials and authentication processes as they relate to the operations across the provider interface

Security requirements for the cloud include user authentication, identity and access management, data protection, multi-tenancy resource isolation, monitoring and auditing for compliance, incident response, user and customer privacy, as well as the underlying portability and interoperability of security components.

3.13

service catalog

a database of information about the cloud services offered by a service provider

The service catalog includes a variety of information about the services, including description of the services, the types of service, cost, supported SLAs, and who can view or use the services. More generally, the service catalog contains information about services through their entire lifecycle. It contains service templates (created by developers), service offerings (created by providers), and deployed service instances.

3.14

service contract

an agreement between the cloud service provider and cloud service consumer to state the terms of the services usage by the cloud service consumer

3.15

service instance

the instantiation of a service request

3.16

service-level agreement

SLA

in the context of cloud service providers, including brokers and federators, a negotiated, legally binding contract between the cloud service provider and cloud service consumer

The SLA includes agreements about services and responsibilities about service availability, performance, and billing. The SLA must be structured such that it can be propagated to all involved cloud service providers. The consumer contracts services from one cloud service provider who could be a broker, federator, or non-broker/non-federator, and that SLA is used as the basis for the broker or federator to programmatically contract for services from other cloud service providers.

3.17

service manager

responsible for managing the service instance and service topology

The service manager provides facilities to administrator to create virtual machine instances and services. The service manager also provides mechanisms for creation, monitoring, control, and reporting of services.

3.18

service offering

a service template combined with SLAs, constraints, costs, billing information and other data necessary to offer the service described in the template to a consumer

3.19

service request

a request by a consumer to instantiate a service offering

Requests require service contracts, which may be created prior to or simultaneously with service requests.

3.20

service template

a collection of items (machine images, connectivity definitions, storage, and so on) that are stored in the service catalog and can be provisioned at the cloud service provider

3.21

service topology item

the components of a service instance

3.22

service topology relationship

defines relationships (for example, depends on, hosts) between service topology items

3.23

template

often used as a synonym for service template

3.24
virtual image
an element (often as part of a package using the Open Virtualization Format) that encapsulates a workload consisting of all the code necessary to run the workload with the metadata that is necessary to configure the environment in which to run it

3.25
workload portability
provides the capability for the cloud service consumer to create a service package and then provision that package in different cloud service providers without substantial modifications

4 Symbols and Abbreviated Terms

3.1
DoS
denial of service

3.2
IaaS
Infrastructure as a Service

3.3
IDS
intrusion-detection system

3.4
NIST
National Institute of Standards and Technology

3.5
OVF
Open Virtualization Format

3.6
PaaS
Platform as a Service

3.7
QoS
quality of service

3.8
REST
Representational State Transfer

3.9
SLA
service-level agreement

3.10
SLO
service-level objective

3.11
URI
Uniform Resource Identifier

5 Introduction

The definitions of cloud computing — including private and public clouds, Infrastructure as a Service (IaaS), and Platform as a Service (PaaS) — are taken from work by the National Institute of Standards and Technology (NIST). In part, NIST defines cloud computing as "… a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (for example, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

NIST defines four cloud deployment models:

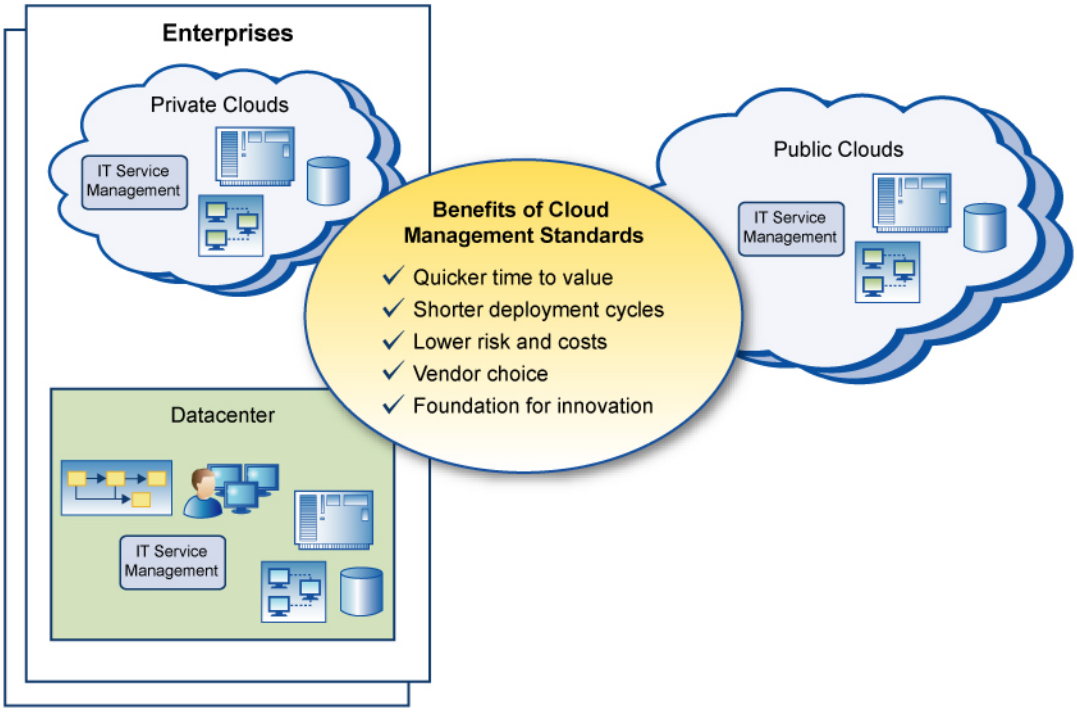
- public clouds (cloud infrastructure made available to the general public or a large industry group)
- private clouds (cloud infrastructure operated solely for an organization)
- community clouds (cloud infrastructure shared by several organizations)
- hybrid clouds (cloud infrastructure that combines two or more clouds)

The environment under consideration by the Open Cloud Standards Incubator includes all of these deployment models. The main focus of the Incubator is management aspects of IaaS, with some work involving PaaS. These aspects include service-level agreements (SLAs), quality of service (QoS), workload portability, automated provisioning, and accounting and billing. This scope is described in the white paper, *Interoperable Clouds*.

The fundamental IaaS capability made available to cloud consumers is a cloud service. Examples of services are computing systems, storage capacity, and networks that meet specified security and performance constraints. Examples of consumers of cloud services are enterprise datacenters, small businesses, and other clouds.

Many existing and emerging standards will be important in cloud computing. Some of these, such as security-related standards, apply generally to distributed computing environments. Others apply directly to virtualization technologies, which are expected to be important building blocks in cloud implementations. (The dynamic infrastructure enabled by virtualization technologies aligns well with the dynamic on-demand nature of clouds.) Examples of standards include SLA management and compliance, federated identities and authentication, and cloud interoperability and portability.

Figure 1 shows the scope of the Open Cloud Standards Incubator and the benefits of extending management and virtualization standards.



404

405 **Figure 1 – Scope and Benefits of DMTF Open Cloud Standards Incubator**

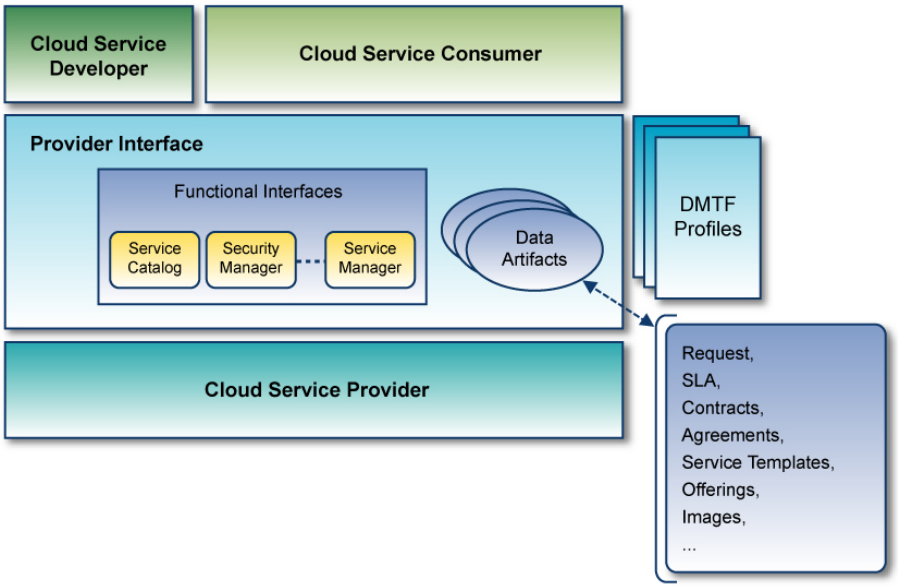
406 The Open Cloud Standards Incubator addresses the following aspects of the lifecycle of a cloud service:

407 • description of the cloud service in a template
408 • deployment of the cloud service into a cloud
409 • offering of the service to consumers
410 • consumer entrance into contracts for the offering
411 • provider operation and management of instances of the service
412 • removal of the service offering

413 When practical, existing standards (or extensions to them) will be integrated into the recommended
414 solution. Examples of standardization areas include resource management protocols, data artifacts,
415 packaging formats, and security mechanisms to enable interoperability.

416 6 Reference Architecture Overview

417 This section introduces the conceptual Cloud Service Reference Architecture (Figure 2), which describes
418 key components — such as actors, interfaces, data artifacts, and profiles — and the interrelationships
419 among these components.

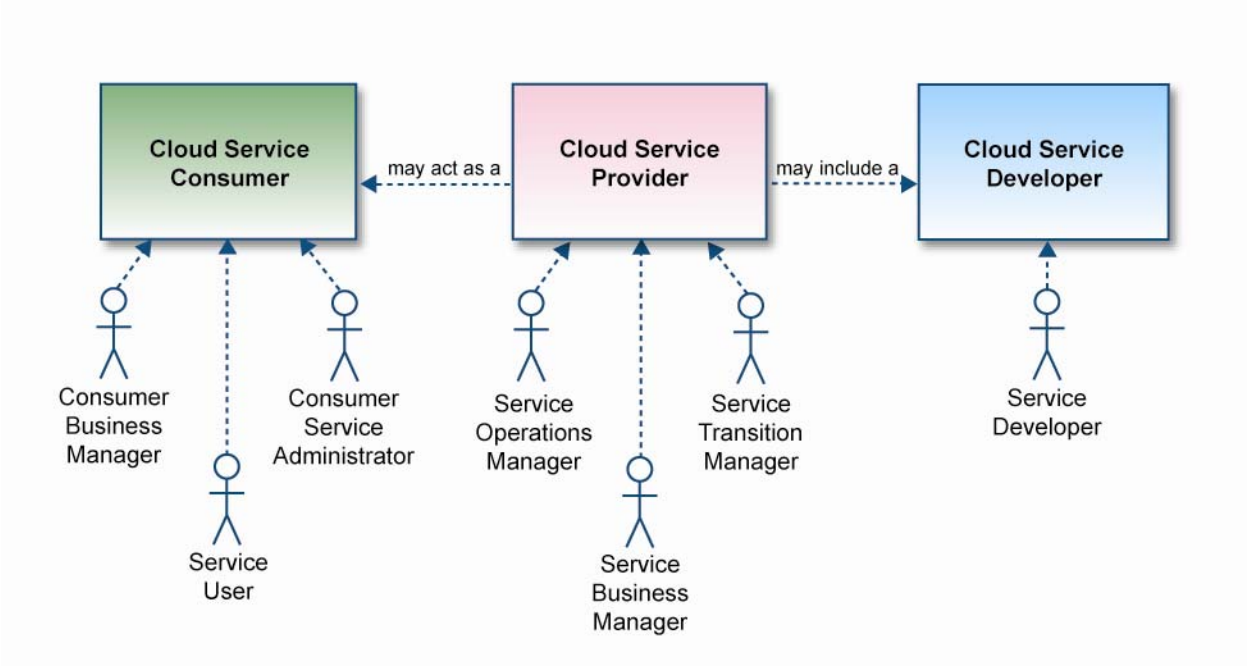


420

421 **Figure 2 – Cloud Service Reference Architecture**

422 # 7 Actors

423 Figure 3 shows the three actor categories (cloud service consumer, provider, and developer) and the
424 seven detailed actors within the categories. The actor categories are shaded in different colors to aid
425 understanding: consumer (green), provider (pink), and developer (blue).



426

427 **Figure 3 – Cloud Actors**

428 Table 1 describes each actor in more detail.

429 **Table 1 – Cloud Actor Descriptions**

Actor Category	Actor	Description
Cloud Service Developer	Service Developer	• Designs, implements, and maintains service templates (technical aspect). These templates can be used by providers to create offerings. The person performing this role could be employed by the same organization that is a cloud service provider or cloud service consumer.
Cloud Service Provider	Service Operations Manager	• Manages the technical infrastructure required for providing cloud services • Monitors and measures performance and utilization against SLAs • Provides reports from monitoring and measurement (used for audit and compliance)
	Service Business Manager	• Offers all types of services developed by cloud service developers • Accounts for services potentially offered by service providers themselves and services offered on behalf of cloud service developers • Establishes a portfolio of business relationships, and sets up accounts and terms for cloud service consumers (including the master account, which is the master relationship between the consumer and provider)
	Service Transition Manager	• Enables a customer to use the cloud service, including "onboarding", integration, and process adoption • Defines and creates service offerings based on templates that can be used by cloud service consumers and are populated into the catalog
Cloud Service Consumer	Consumer Business Manager	• Approves business and financial expenditures for consumed services • Accounts for used service instances • Establishes business relationships; sets up accounts, budget, and terms; and so on
	Consumer Service Administrator	• Requests service instances and changes to service instances (typically on behalf of the consumer business manager) • Purchaser of services with the business relationship; creates Service User roles • Creates users (including policies), allocates resources, such as compute and storage, generates reports (usage); performs actions for a specific set of resources
	Service User	• Uses service instances provided by a cloud service provider. The Service User role is distinct from the Consumer Service Administrator role because it has no responsibility for managing the service.

8 Lifecycle of a Cloud Service

Figure 4 shows the six lifecycle states of a typical cloud service with the use cases that are most relevant to each state. Use cases marked with an asterisk in the figure are described in detail in this document.

- Template – A developer defines the service in a template that describes the content of and interfaces to a service.
- Offering – A provider adds constraints, costs, and policies to a template to create an offering available for request by a consumer.
- Contract – A consumer and provider enter into a contract for services, including agreements on costs, SLAs, SLOs, and specific configuration options.
- Provision Service – A provider deploys (or modifies) a service instance per the contract with the consumer.
- Runtime Maintenance – A provider manages a deployed service and all its resources, including monitoring resources and notifying the consumer of key situations.
- End of Service – A provider and consumer agree to terminate a service. The provider halts a service instance and reclaims resources for redeployment to support other services.

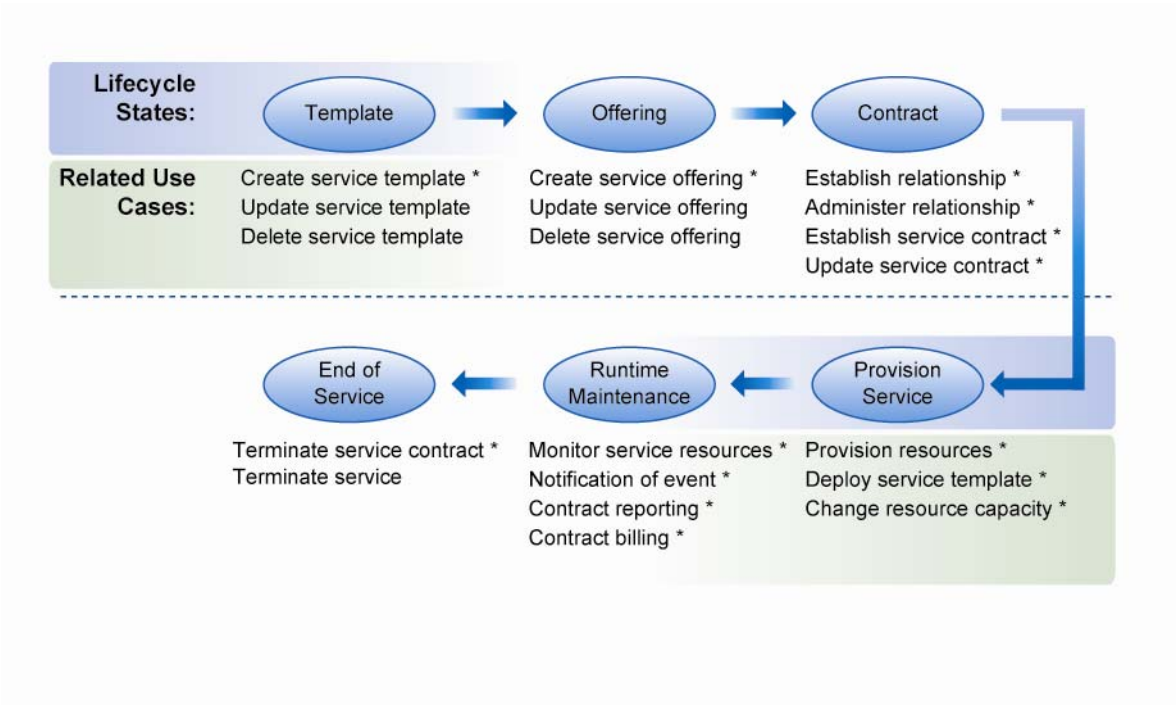


Figure 4 – Cloud Service Lifecycle and Use Cases

9 Overview of Primary Service Artifacts

Figure 5 shows some of the most important data artifacts associated with the lifecycle of a service, along with brief descriptions. A more complete picture of service-related objects is shown in Figure 20 on page 52.

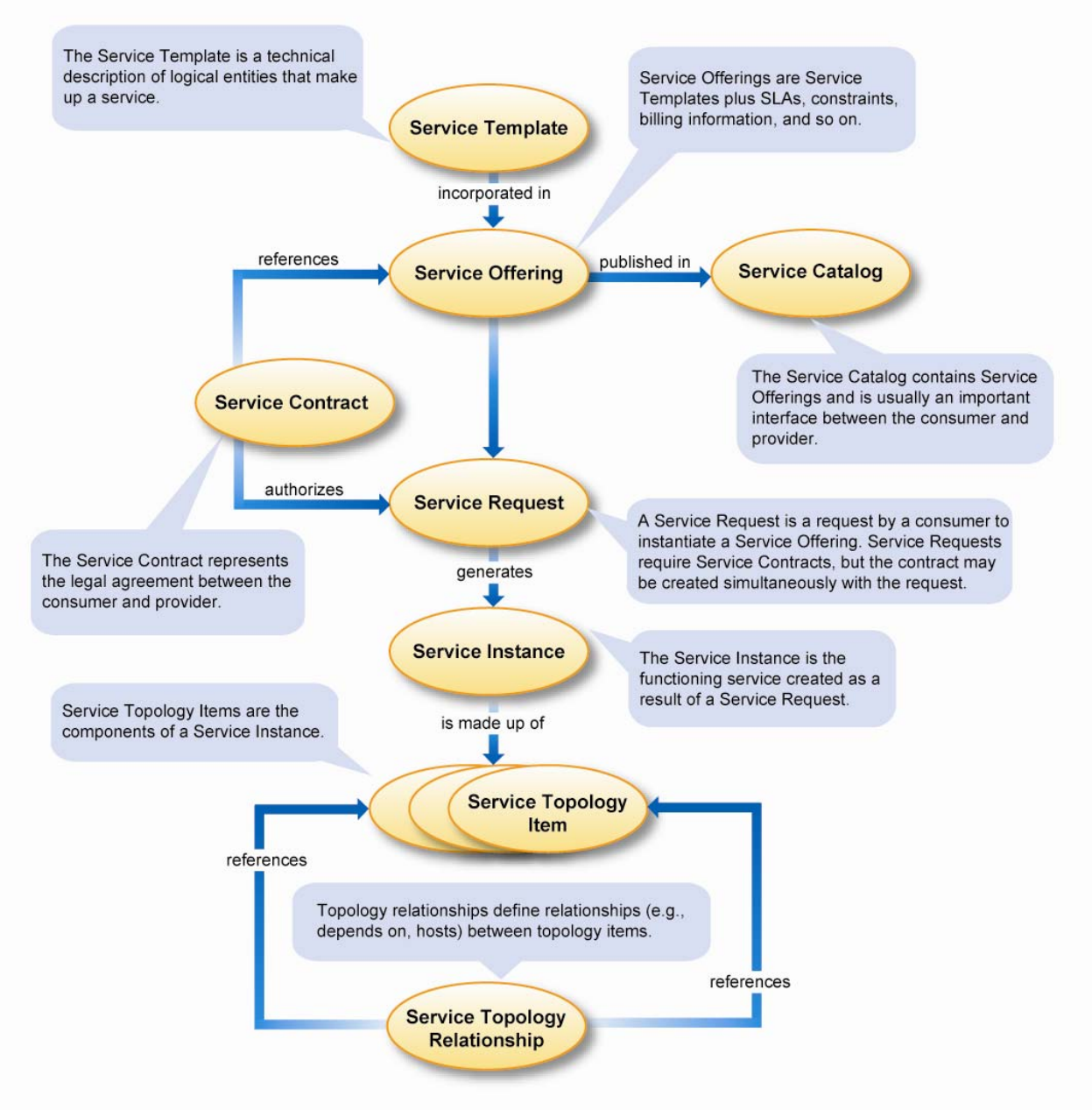


Figure 5 – Service Artifacts Overview

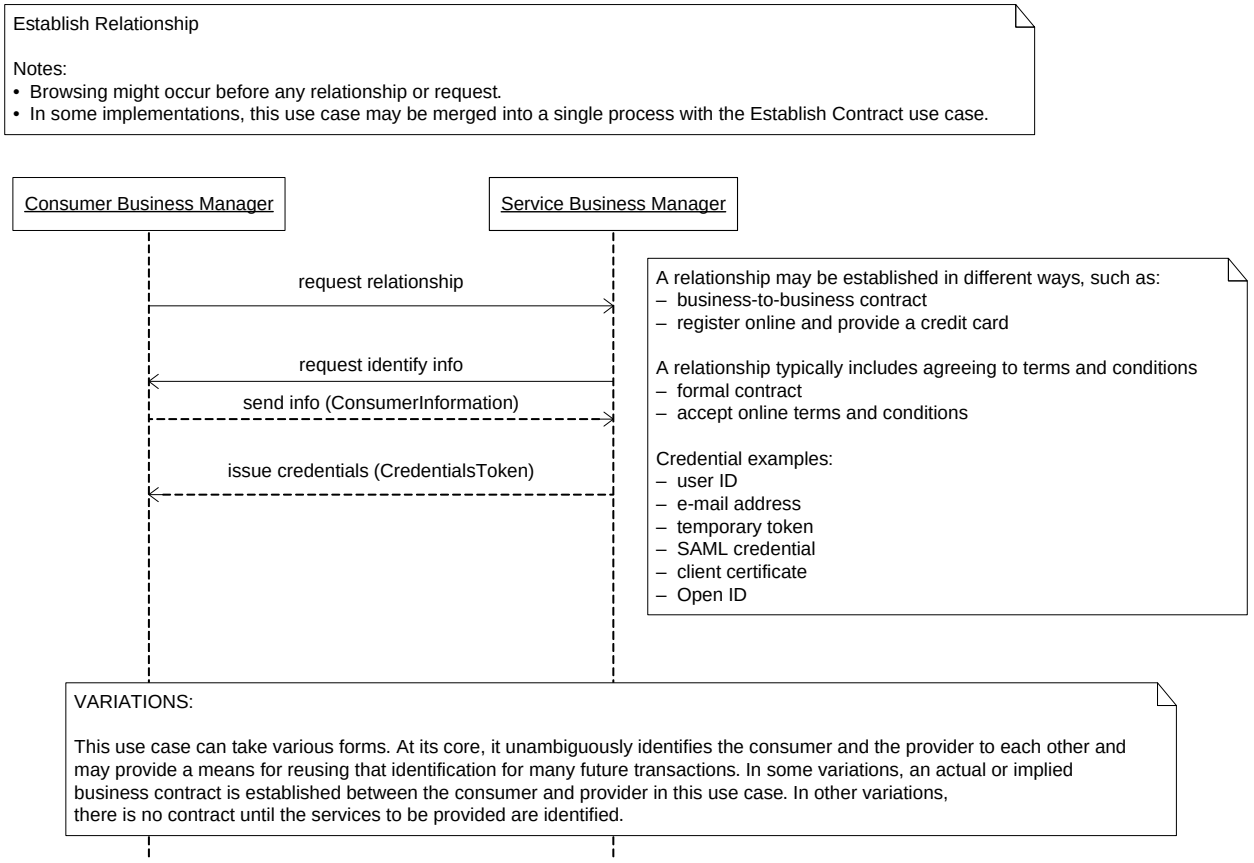
10 Use Cases

This section presents the management use cases for the lifecycle of a cloud service. The information for each use case is presented in a table format, with a figure that illustrates the process flow. See Figure 4 for a summary of the use cases and the service lifecycle states to which they apply.

10.1 Establish Relationship

<table>
<tr><th>DMTF-001</th><th colspan="2">Establish Relationship</th></tr>
<tr><td>Description</td><td colspan="2">A potential consumer of a cloud-based service establishes their identity with a cloud service provider for use in future transactions.</td></tr>
<tr><td>Desired outcome</td><td colspan="2">The cloud service consumer receives some form of token or proof of identification that can be used in a future transaction. The transaction may be in the immediate future, in which case establishment of identity may be combined with requesting a service contract and even service delivery. The token received in this use case may be good for a single use or it may be permanent.</td></tr>
<tr><td>Business events</td><td colspan="2">The triggering events for establishing a relationship are basically for the same events that trigger a request for a service contract because the relationship is formed for executing implied or actual service contracts. Following are examples of triggering events:
A consumer identifies a new need and seeks a service capable of meeting that need at an acceptable cost.
A service is publicized, and a potential consumer recognizes the service as part of a better solution than an existing solution.
A service is publicized, and a potential consumer discovers a possible new activity.
An existing service is found to be inadequate in utility, availability, capacity, reliability, cost, or in other ways, and the cloud service consumer seeks an alternative service.</td></tr>
<tr><td>Actors</td><td colspan="2">Service Business Manager or agent
Consumer Business Manager or agent</td></tr>
<tr><td>Involved components and services</td><td colspan="2">Cloud service provider identity service
User ID, email address, token, certificate or other credential</td></tr>
<tr><td>Dependencies</td><td colspan="2">None</td></tr>
<tr><td>Process assumptions</td><td colspan="2">The cloud service provider has an ITIL-style service catalog, as described in the ITIL Glossary of Terms, Definitions, and Acronyms, which is available to potential cloud service consumers.</td></tr>
<tr><td rowspan="3">Process flow</td><td>Step description</td><td>Data or data artifact required</td></tr>
<tr><td>1. The Consumer Business Manager (CBM) (or other consumer representative) requests the establishment of a relationship.</td><td>None</td></tr>
<tr><td>2. The Service Business Manager (or other provider representative, which may be an automated agent) requests identifying information.</td><td>Data requested of the consumer may include:
Name
Business name
Physical mailing address
Telephone numbers
Email addresses
Other identifying data
In some cases, payment information such as credit card numbers is requested at this point.</td></tr>
</table>

DMTF-001	Establish Relationship	
	3. The CBM provides the requested information.	• Consumer information • Contact information
	4. The CBM is returned a credential that may be used in future transactions.	Credentials token
Variations	This use case can take various forms. At its core, it unambiguously identifies the consumer and the provider to each other and may provide a means for reusing that identification for many future transactions. In some variations, an actual or implied business contract is established between the consumer and provider. In other variations, no contract is established until the services to be provided are identified.	
Additional information	None	
Design notes	This use case is often combined with use cases that request a service contract and even service delivery. In some real-world situations, a cloud service consumer identifies themselves with a credit card number, selects a service such as the use of a computing service with standard capacity and features for a specified period, pays, and receives access to the service in a single episode that is completed in a few minutes. In other cases, a consumer may register with a provider and receive a user ID and password. Sometime later, perhaps months later, the consumer uses the ID and password to request a contract and then after another interval, requests that the service be provisioned. Finally, after yet another interval, the consumer begins to use the service.	

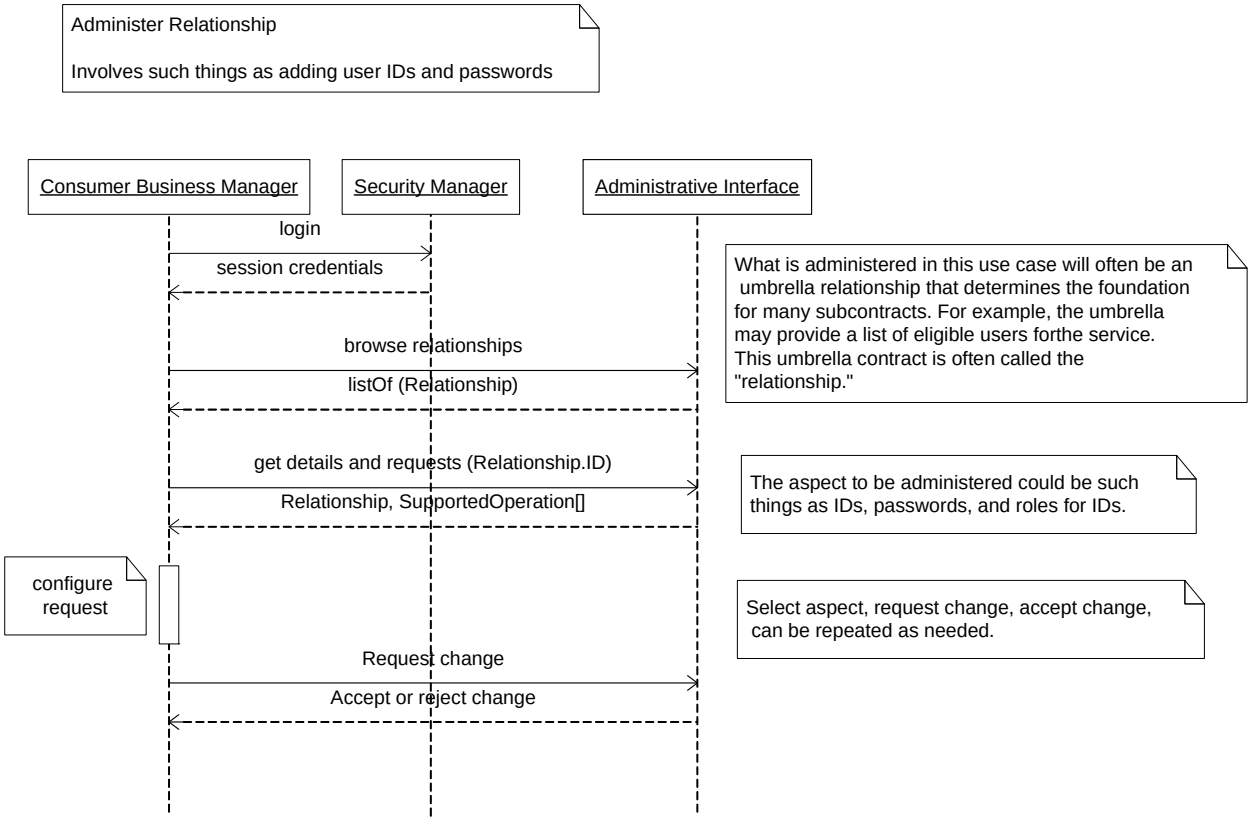


458

459 **Figure 6 – Establish Relationship**

10.2 Administer Relationship

DMTF-002	Administer Relationship	
Description	A potential consumer of a cloud-based service requests administration of a contract. Administration is distinguished from changing a service because administration does not affect the technical delivery of a service. Usually, contract administration involves actions like adding new users or changing user passwords that are associated with an umbrella contract (usually called the “relationship”), not a contract for a specific service.	
Desired outcome	The service consumer is properly authenticated and is able to make the administrative change.	
Business events	The triggering event could be as follows: • A consumer wants to add or remove a user for the services offered by the provider. This may be due to a change in the consumer’s business or the on-boarding or off-boarding of an employee. • Due to policy or a security issue, a password must be changed, or another runtime event, such as auditing, monitoring, or customer reporting, must be changed.	
Actors	• Service Business Manager (SBM) • Consumer Business Manager (CBM)	
Involved components and services	• Service catalog • Service contract • External (third party) identity management service	
Dependencies	• Relationship between consumer and provider has been established • External identity management service (in case third-party attestation is needed)	
Process assumptions	• The consumer has a relationship with the provider. • Although it is likely, the consumer need not have a contract for any specific service. • The cloud service provider has federated identity management and standard authentication mechanisms such as Oauth/OpenID, LDAP/AD, X509 CA, or SAML.	
Process flow	**Step description**	**Data or data artifact required**
	1. CBM logs in.	Established account, security profile, and credentials
	2. CBM requests administrable aspects from a catalog for browsing.	Consumer security policies
	3. Service catalog returns a list of administrable aspects for browsing.	None
	4. The CBM chooses an aspect and makes a request.	None
	5. The SBM accepts the CBM request, makes the change, and establishes a consumer security profile.	None
	6. The CBM accepts the change.	None
	Steps 4–6 may be repeated to deal with many administrative changes in the same session.	
Variations	None	
Additional information	None	
Design notes	None	



461

462 **Figure 7 – Administer Relationship**

463 ## 10.3 Establish Service Contract

DMTF-003	Establish Service Contract
Description	A potential consumer of a cloud-based service requests a service contract for a cloud-based service.
Desired outcome	The cloud service consumer receives and ratifies in some form, online or offline, a contract that supports a reasonable business and technical decision to subscribe to the delivery of a service. The contract should clearly depict or indicate the business and technical information necessary for the consumer to use the service effectively and in a businesslike fashion.
Business events	The triggering event could be as follows: • A consumer identifies a new need and seeks a service capable of meeting that need at an acceptable cost. • A service is publicized, and a potential consumer recognizes the service as part of a better solution than an existing solution. • A service is publicized, and a potential consumer discovers a possible new activity. • An existing service is found to be inadequate in utility, availability, capacity, reliability, cost, or in other ways, and the service consumer seeks an alternative service. No matter what the triggering event, the potential consumer must evaluate the service contract offered by the cloud service provider and make a decision.

<table>
<tr><th>DMTF-003</th><th colspan="2">Establish Service Contract</th></tr>
<tr><td>Actors</td><td colspan="2">• Service Business Manager (SBM), who offers the service contract
• Consumer Business Manager (CBM) or executive, who accepts the contract</td></tr>
<tr><td>Involved components and services</td><td colspan="2">• Service catalog
• Service contract
• The proffered service</td></tr>
<tr><td>Dependencies</td><td colspan="2">• Service offering available in the service catalog</td></tr>
<tr><td>Process assumptions</td><td colspan="2">The cloud service provider has an ITIL-style service catalog, as described in the ITIL Glossary of Terms, Definitions, and Acronyms, which is available to potential cloud service consumers.

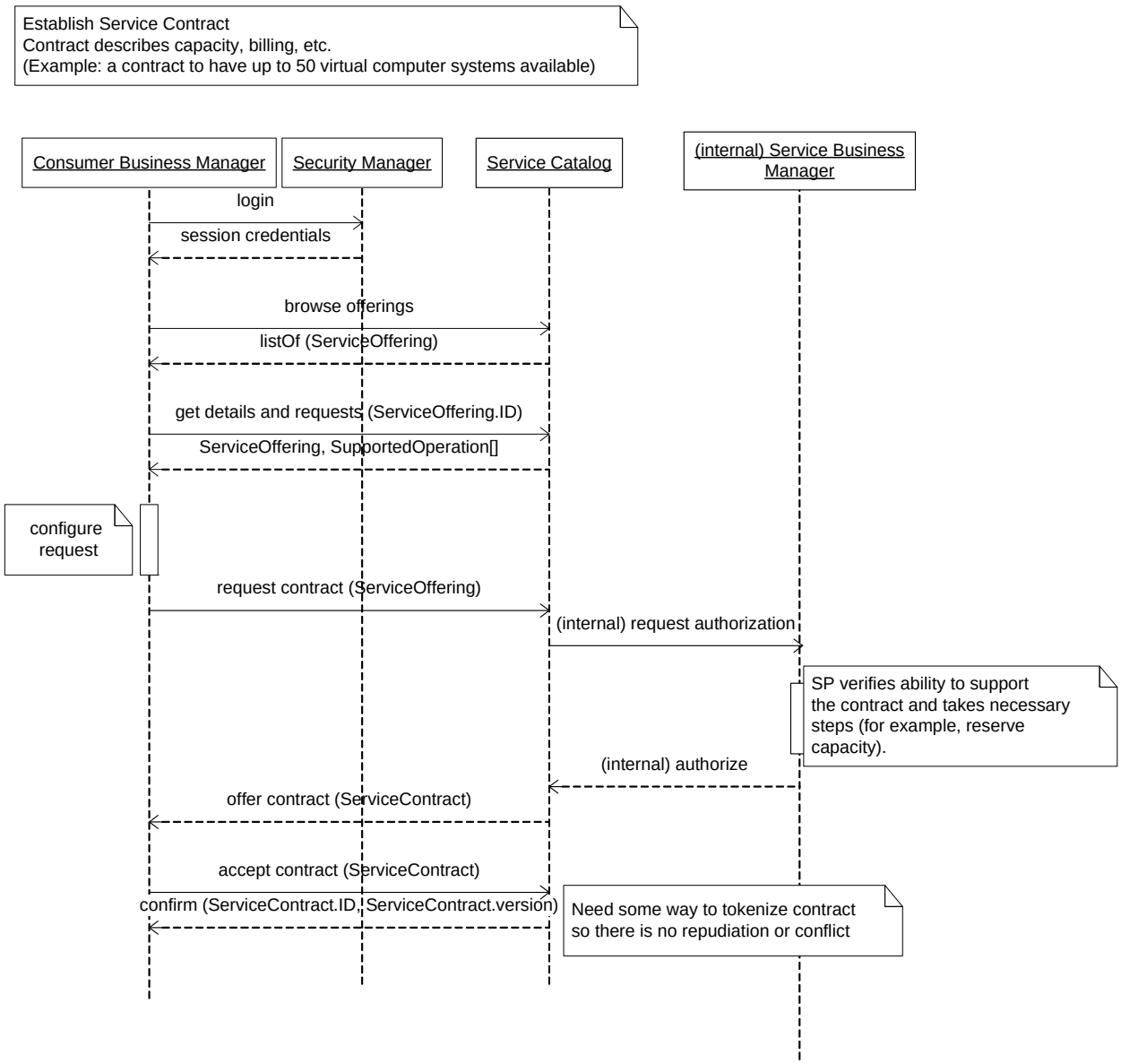
A service offering has already been created and is stored in the service catalog (see the Create Service Offering use case in 10.13).</td></tr>
<tr><td rowspan="10">Process flow</td><td>Step description</td><td>Data or data artifact required</td></tr>
<tr><td>1. The CBM logs in.</td><td>Established account, security profile, and credentials</td></tr>
<tr><td>2. The CBM browses available service offerings.</td><td>Service offering</td></tr>
<tr><td>3. The CBM gets the details about service offerings, including configurable parameters and supported requests.</td><td>• Service offering
• Supported operations</td></tr>
<tr><td>4. The CBM chooses options and makes a request.</td><td>Service request</td></tr>
<tr><td>5. (internal) The SBM analyzes the request and their ability to support it.</td><td>None</td></tr>
<tr><td>6. (internal) The SBM authorizes the contract.</td><td>None</td></tr>
<tr><td>7. The service catalog returns a contract for CBM acceptance.</td><td>Service contract</td></tr>
<tr><td>8. The CBM accepts the contract.</td><td>Service contract</td></tr>
<tr><td>9. The SBM confirms acceptance, providing credentials satisfactory to support non-repudiation requirements.</td><td>None</td></tr>
<tr><td>Variations</td><td colspan="2">This use case could take place inside a single organization or between organizations. When the use case is within a single organization, the costs and benefits may be tracked in the form of charge backs, or not tracked at all, and the service level agreement and service contract are usually documents, but not legal agreements. The organization determines how closely the agreements and financial tracking conform to the pattern for interorganization agreements.

In the two-tier variation, the two tiers are the end user of services implemented with cloud services and the cloud service provider. In this variation, the service user is exposed directly to the service provided by the cloud service provider and is most likely aware of the cloud service provider's presence. The service contract typically is set up by an enterprise manager and made available to the service user. After the initial setup, the service user directly consumes services provided by the cloud service provider.

In the three-tier variation, the three tiers are the end user of services implemented with cloud services, the enterprise data center that uses the cloud services to supply services to the service user, and the cloud service provider. In this scenario, the service user has no exposure to the cloud service provider and likely has no awareness of the cloud service provider's presence.

The SBM may decline to authorize the contract.</td></tr>
</table>

DMTF-003	Establish Service Contract
Additional information	None
Design notes	None



464

465 **Figure 8 – Establish Service Contract**

10.4 Update Service Contract

DMTF-004	Update Service Contract	
Description	A consumer of a cloud service contract and a provider of a cloud service contract agree to update the contract.	
Desired outcome	The contract is revised to the mutual satisfaction of the consumer and provider and is ratified by both parties.	
Business events	The triggering events for updating an existing cloud service contract can come from either the provider or the consumer. Possible consumer triggering events are as follows: • A consumer identifies a new need and seeks a service capable of meeting that need at an acceptable cost. The consumer then requests that the new service be added to the contract or that greater capacity be added to an existing service. • Due to changing business circumstances, the consumer may not need contracted services and wants to reduce the capacity or scope of the contract. • The consumer may want to keep an existing service but change its details (for example, exchange Windows servers for Linux servers) while keeping the service essentially the same. Possible provider triggering events are as follows: • The provider may be forced to modify a service offering. For example, a storage provider may replace hardware and consequently change its backup plans. If the backup plan is a detail of a service contract, the contract may have to change. • A natural disaster or similar event may temporarily change the nature of a service, and a new contract may have to be established.	
Actors	• Service Business Manager (SBM) or agent • Consumer Business Manager (CBM) or agent	
Involved components and services	• User ID, email address, token, certificate or other user credential • Service catalog	
Dependencies	• Service contract established between the consumer and provider	
Process assumptions	• A relationship and contract between the consumer and provider already exist. • The initial contract is written in such a way that it can be extended or modified rather than terminated and rewritten.	
Process flow	**Step description**	**Data or data artifact required**
	1. The CBM logs in.	Established account, security profile, and credentials
	2. The CBM browses service contracts.	Service contract
	3. The CBM gets the details about service contracts, including configurable parameters and supported requests.	• Service contract • Supported operations
	4. The CBM configures the requested changes based on the configurable parameters and makes a request.	Service contract
	5. *(internal)* The SBM analyzes the request and their ability to support it.	None
	6. *(internal)* The SBM authorizes the contract change.	None

DMTF-004	Update Service Contract	
	7. The service catalog returns a modified contract for acceptance.	Service contract
	8. The CBM accepts the contract.	Service contract
	9. The SBM confirms acceptance, providing credentials satisfactory to support non-repudiation requirements.	None
Variations	The transaction can be initiated by either the consumer or provider, but is most frequently initiated by the consumer. Most automated contract modification systems are designed so that consumers can easily adjust the amount of service. Less often, providers must request that a contract be modified. In such cases, the provider proposes the change. In both cases, both the consumer and the provider must accept the contract. The SBM could decline to authorize the contract change.	
Additional information	None	
Design notes	The steps in this use case may be combined and shortened in automated situations when contracts are simple and options are few. In its simplest form, the user logs in to the provider's system and a form is displayed showing the options the consumer has chosen. The consumer clicks on the options to change, clicks Accept, and the transaction is complete. All the steps can be combined because there are few options. On the other hand, a large cloud-based outsourcing contract may be quite complex, involving many iterations of negotiations.	

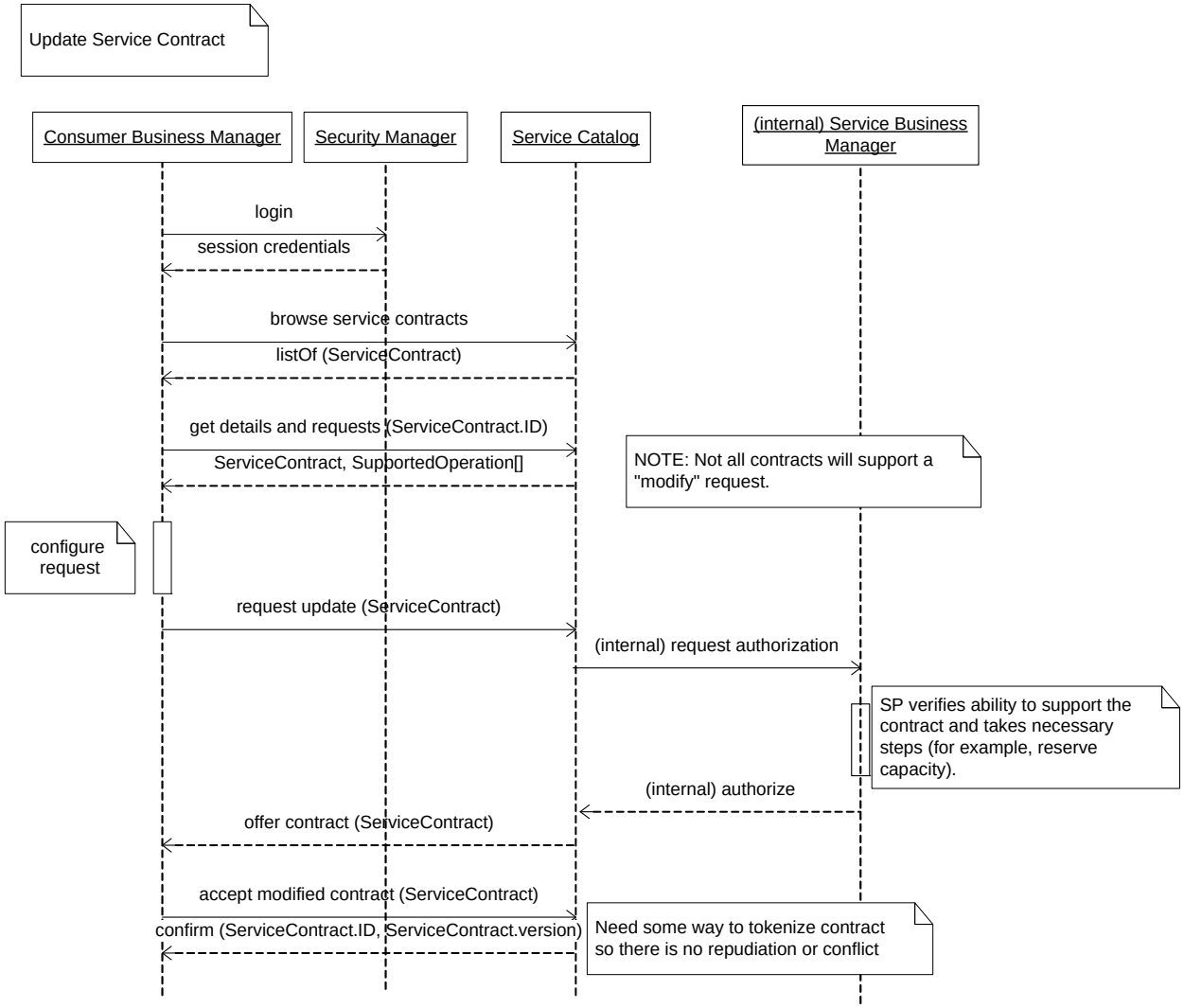


Figure 9 – Update Service Contract

10.5 Contract Reporting

DMTF-005	Contract Reporting
Description	A cloud service consumer requests and receives a report about an established service contract.
Desired outcome	An authorized Consumer Service Administrator or Consumer Business Manager receives reports of actual performance metrics against predefined metrics.
Business events	A predefined condition or event for reporting of SLA metrics occurs.
Actors	• Service Business Manager (SBM) • Consumer Service Administrator (CSA) or Consumer Business Manager (CBM)

<table>
<tr><th>DMTF-005</th><th colspan="2">Contract Reporting</th></tr>
<tr><td>Involved components and services</td><td colspan="2">• Service catalog
• Metering and billing
• Cloud service provider management infrastructure (availability, fault, performance, accounting, and so on)</td></tr>
<tr><td>Dependencies</td><td colspan="2">• Service offering is published to service catalog with associated service metrics.
• The cloud service provider infrastructure has the capability to record and process SLA metrics and analytics.
• The CBM or CSA has an access mechanism for or notification of SLA reports.</td></tr>
<tr><td>Process assumptions</td><td colspan="2">• The service catalog contains reporting metrics associated with defined service offerings.
• Established format and protocols exist for SLA reporting.</td></tr>
<tr><td rowspan="7">Process flow</td><td>Step description</td><td>Data or data artifact required</td></tr>
<tr><td>1. The CBM logs in.</td><td>Established account, security profile, and credentials</td></tr>
<tr><td>2. The CBM browses report definitions.</td><td>Report definition</td></tr>
<tr><td>3. The CBM gets the details about report definitions, including configurable parameters and supported requests.</td><td>• Report definition
• Supported operations</td></tr>
<tr><td>4. The CBM configures the desired report using the configurable parameters and makes a request.</td><td>Report definition</td></tr>
<tr><td>5. The cloud service provider creates a report.</td><td>None</td></tr>
<tr><td>6. The cloud service provider returns the report.</td><td>Report definition</td></tr>
<tr><td>Variations</td><td colspan="2">Anonymous or delegated, role-based access to performance metrics in SLAs might exist.
SLAs for cloud service consumers might be customized or standard.
When cloud service providers are delivering services built on the infrastructure of other cloud service providers, the actor that receives SLA reports may be cloud service consumer management (SLA management and reporting).
Contract reporting may provide for actual performance against defined service level objectives or agreements.
Contract reporting may be temporal in nature (for instance, with a cadence), on demand, or provided via messages when predefined conditions are met.</td></tr>
<tr><td>Additional information</td><td colspan="2">None</td></tr>
</table>

DMTF-005	Contract Reporting
Design notes	Examples of types of reports are as follows: • Service usage (overall, per user) • Consumption vs. contracted limits • Availability vs. contracted levels • Performance / SLA compliance • SLA violations • Committed vs. available (burstable) • Metrics per operational tasks (for example, quantity, response time) • Operational task activity audit log • Security incidents

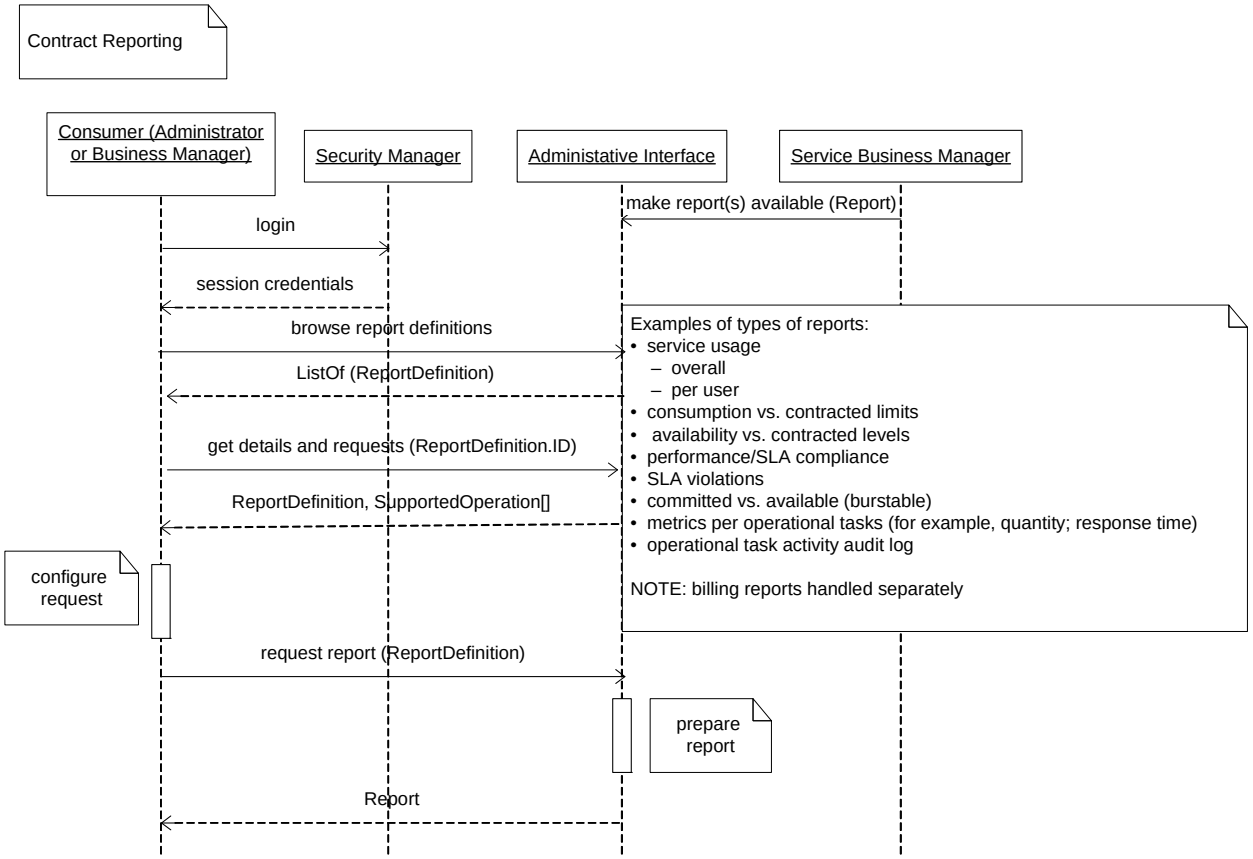
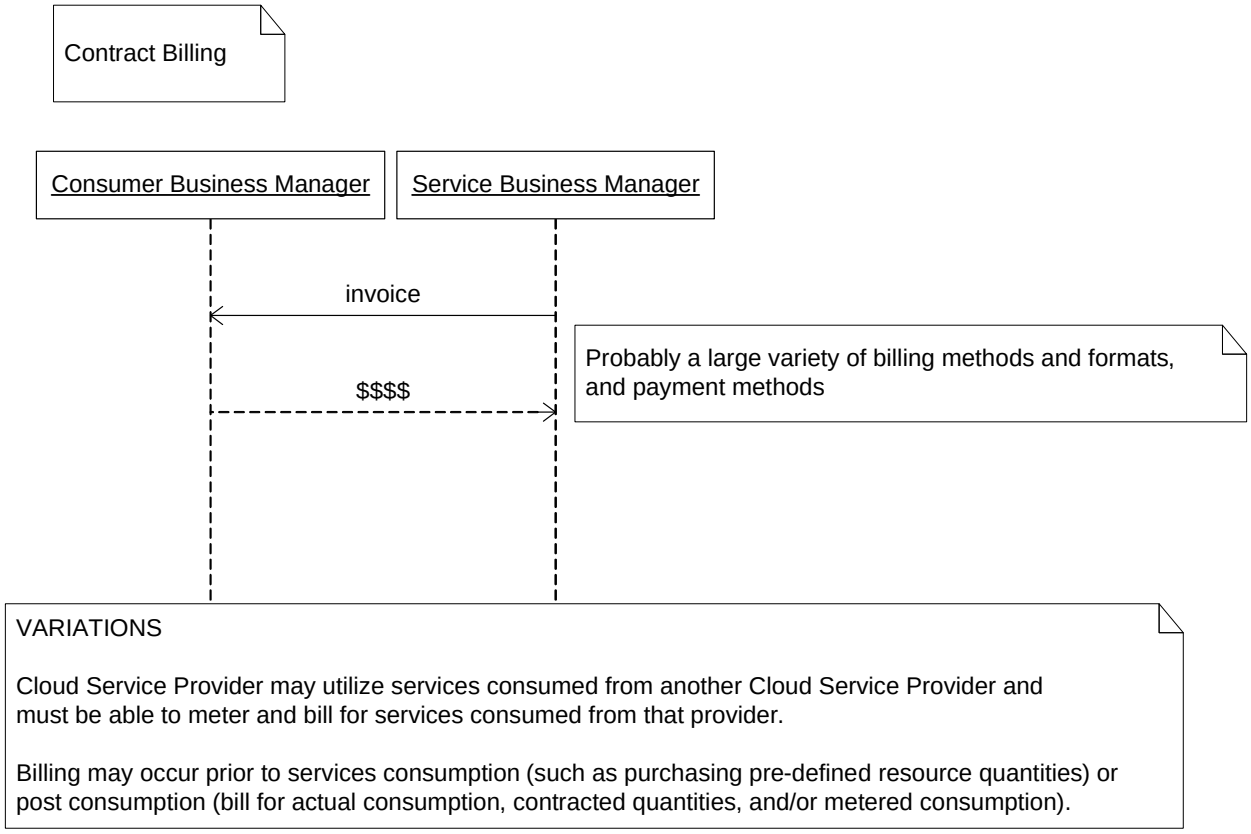


Figure 10 – Contract Reporting

10.6 Contract Billing

DMTF-006	Contract Billing	
Description	A cloud service provider issues an invoice for contracted or consumed services.	
Desired outcome	The cloud service provider bills the cloud service consumer via an invoice (metering and accounting) for resources consumed.	
Business events	The triggering event is a periodic or on-demand publication or pull of billing for services consumed (per contract terms and mechanisms).	
Actors	• Consumer Business Manager (CBM) • Service Business Manager (SBM)	
Involved components and services	• Billing system • Metering system (optional) • Accounting system (optional) • Identity and security • Service catalog	
Dependencies	• A contract is defined. • A relationship is established. • Predefined temporal or other event occurs to push or pull contract billing information (that is, an invoice).	
Process assumptions	• Protocols and formats for transmission and receipt of billing information are established. • The service catalog contains metrics that enable metering and or billing functions to occur. • The SBM provides a mechanism to make invoice information available to the CBM. • The service contract contains the definition and process of how invoice metrics are to be calculated.	
Process flow	**Step description**	**Data or data artifact required**
	1. The SBM provides or makes an invoice available.	Invoice (contents are provider-specific)
	2. The CBM accesses the billing information and invoice.	Invoice (contents are provider-specific)
Variations	The cloud service provider may use services consumed from another cloud service provider and must be able to meter and bill for services consumed from that provider. Billing may occur before service consumption (such as purchasing predefined resource quantities) or after consumption (bill for actual consumption, contracted quantities, or metered consumption).	
Additional information	None	
Design notes	None	



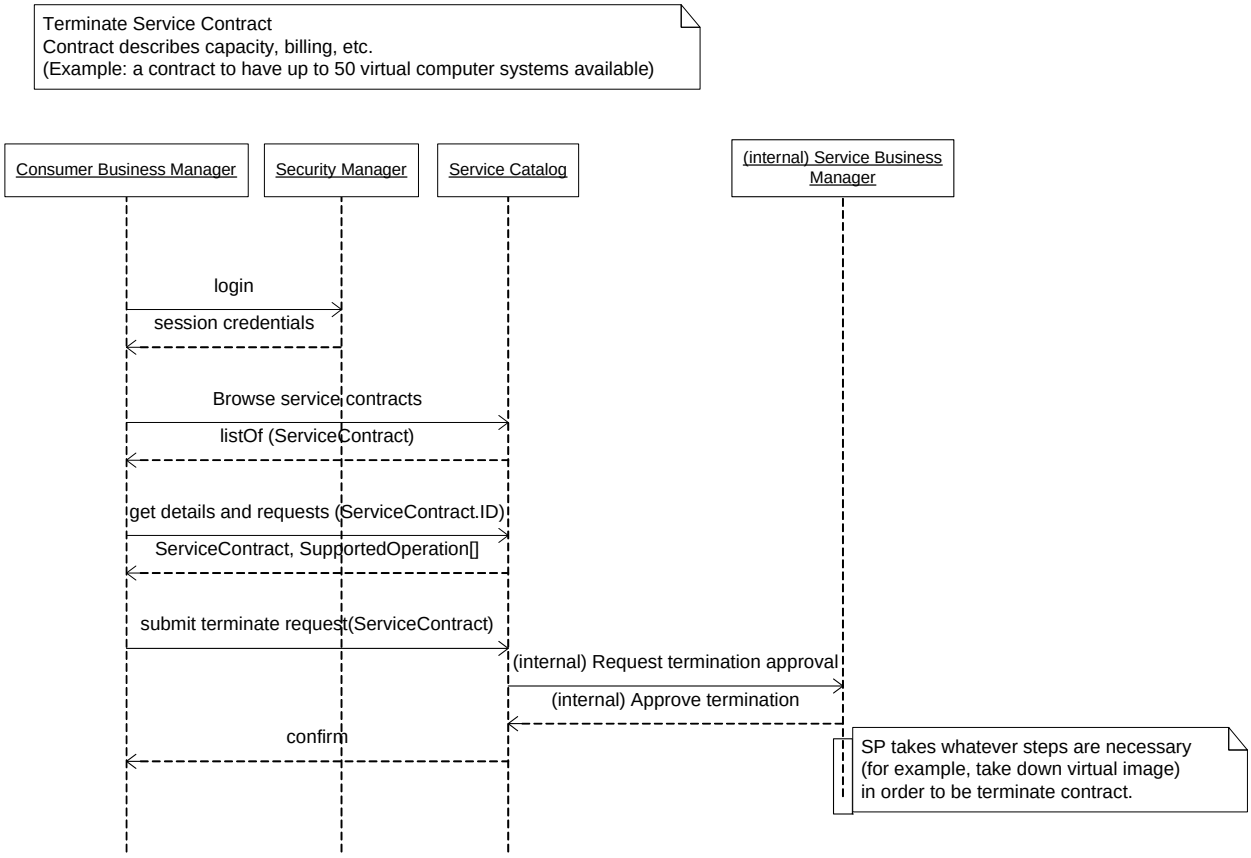
473

474 **Figure 11 – Contract Billing**

10.7 Terminate Service Contract

DMTF-007	Terminate Service Contract	
Description	A consumer of a cloud service contract and a provider of a cloud service contract agree to terminate a cloud service contract.	
Desired outcome	The contract is terminated to the mutual satisfaction of the consumer and provider and is ratified by both parties. The service instance is removed and resources are released to the pool and are ready for request by another consumer, without compromising any confidential information from the previous consumer.	
Business events	The triggering events for terminating an existing cloud service contract can come from either the provider or the consumer. Possible consumer triggering events are as follows: • Due to changing business circumstances, the consumer may not need contracted services and want to terminate the contract. • The need for the service was temporary, and the service is no longer needed. • The service was unacceptable for some reason, and the consumer decides to end the relationship. Possible consumer triggering events are as follows: • Due to changing business circumstances, the provider can no longer offer the service. • A natural disaster or similar event may force termination of the service. • The provider may find a consumer to be unacceptable for some reason and want to terminate the service.	
Actors	• Service Business Manager (SBM) or agent • Consumer Business Manager (CBM) or agent	
Involved components and services	• User ID, email address, token, certificate or other user credential, consumer security profile • Service catalog	
Dependencies	• Service contract established between a consumer and provider	
Process assumptions	• A relationship and contract between the consumer and provider already exist. • The initial contract is written in such a way that it can be terminated without penalty.	
Process flow	**Step description**	**Data or data artifact required**
	1. The CBM logs in.	Established account, security profile, and credentials
	2. The CBM browses service contracts.	Service contract
	3. The CBM gets the details about service contracts, including configurable parameters and supported requests.	• Service contract • Supported operations
	4. The CBM configures the terminate request and submits it.	Service contract (may include terms of termination, such as final billing agreement and penalties)
	5. *(internal)* The SBM approves the terminate request.	None

DMTF-007	Terminate Service Contract	
	6. The SBM confirms the termination to the CBM.	• Possible return data or data maintained after contract termination • Evidence of storage data disposal • Evidence of consumer profile disposal • Archival information per the original service contract agreement • Information about backup and retrieval of data and images
	7. *(internal)* The SBM terminates the service contract, reclaiming resources, etc.	None
Variations	The transaction can be initiated by either the consumer or provider, but is most frequently initiated by the consumer. When the contract is terminated by the provider, notification is usually not through the service catalog, although the same steps must take place. Typically, neither party is able to stop the other party from terminating, although there may be penalties for termination. Most automated contract modification systems are designed so that consumers can easily terminate a service.	
Additional information	None	
Design notes	The steps in this use case can be combined and shortened in automated situations when contracts are simple and options are few. In its simplest form, the user signs on to the provider's system and a form is displayed showing the options the consumer has chosen. The consumer clicks on an option to terminate, and the transaction is complete. All the steps can be combined because there are few options. On the other hand, a large cloud-based outsourcing contract may be quite complex, involving many iterations of negotiations, even for termination.	



476

477 **Figure 12 – Terminate Service Contract**

478 10.8 Provision Resources

DMTF-008	Provision Resources (from a contracted pool)
Description	Within the context of an existing contract, an administrator allocates resources from the contracted pool. For example, an administrator allocates 10 virtual systems from a contracted pool of up to 50 virtual systems. Typically, these resources are reserved for the requesting user until they are returned to the pool. The resources could be of a wide variety, such as virtual system platforms or a preconfigured mini data center that contains virtual systems and virtual storage, connected via a virtual network. The cloud service consumer might have contracted for up to six such mini data centers.
Desired outcome	The resources are allocated from the pool and are ready for use by the consumer.
Business events	Triggering events are as follows: • A Service User (for example, a test engineer requesting an allocation of test systems) has a business goal to accomplish, such as executing a series of test cases. • A Consumer Business Manager and a Consumer Service Administrator have decided to contract with a cloud service provider for this type of request.

<table>
<tr><th>DMTF-008</th><th colspan="2">Provision Resources (from a contracted pool)</th></tr>
<tr><td>Actors</td><td colspan="2">• Consumer Service Administrator (CSA)
• Service User (for example, a test engineer who works for the company that is the consumer)

The roles of Service User and CSA may sometimes be combined in a single person.
• Service Operations Manager and Service Transition Manager

In this use case, the actions of these provider actors are performed by the Service Manager, Provisioning Manager, and Metering Manager, which are functions that are most likely automated.</td></tr>
<tr><td>Involved components and services</td><td colspan="2">• Service catalog
• Service manager
• Internal services (for example, provisioning and metering)</td></tr>
<tr><td>Dependencies</td><td colspan="2">• There is an existing contract between the cloud service consumer and cloud service provider.
• The CSA's identity or authentication credentials are registered with the provider.
• The CSA is authorized (with the provider) to request the type and quantity of resources.
• There is an entry in a service catalog for this type of request.</td></tr>
<tr><td rowspan="8">Process flow</td><td>Step description</td><td>Data or data artifact required</td></tr>
<tr><td>1. The CSA logs in.</td><td>Established account, security profile, and credentials</td></tr>
<tr><td>2. The CSA browses (or searches) the service catalog to see which offerings are available.

The service catalog filters the list of all offerings to show only those that service user is authorized to request.</td><td>Service offering</td></tr>
<tr><td>3. The CSA selects an offering.

If there are configuration options, such as specifying the number of resources that will be requested or specifying a security environment, they are displayed to the service user.</td><td>• Service offering
• Supported operation</td></tr>
<tr><td>4. The CSA submits the request to the service manager for this type of service.</td><td>Service request</td></tr>
<tr><td>5. The service manager tests the ability to support the service, such as testing for available capacity or prepaid tokens.</td><td>None</td></tr>
<tr><td>6. If the request can be supported, the service manager offers the capacity to the service user.

For example, an "Offer" window with the details of the offer could be displayed.</td><td>Service request</td></tr>
<tr><td>7. The CSA accepts the offer.</td><td>Service request</td></tr>
</table>

DMTF-008	Provision Resources (from a contracted pool)	
	8. The service manager acknowledges that the request is being fulfilled. This acknowledgement could be repeated periodically, potentially with a progress indicator (for example, request is 64% complete).	None
	9. The service manager initiates the provisioning of the resources, such as triggering a provisioning manager and passing the parameters of the request.	None
	10. *(optional)* The CSA may query for the status of the request and receive a response. This may repeat any number of times.	Service request
	11. *(internal)* The provisioning manager configures and schedules the workflows, which start executing per the schedule.	None
	12. *(internal)* The workflows invoke various provisioning steps, each of which may itself be a workflow. Each provisioning step may initiate manual or automated processing. Examples: • Plug blades into a rack. • Allocate computing systems to a pool. • Configure zones in a system. • Deploy and activate monitoring agents.	None
	13. *(internal)* The status of the provisioning steps is tracked, and an audit log is maintained. If there are no errors, the process completes and the fulfilled offer is ready to be used.	None
	14. The service manager notifies the CSA that the provisioning has completed either successfully or unsuccessfully. This notification could be an asynchronous message sent to the user (for example, an email message) or an update to a feed that the service user periodically checks.	Service instance
	15. *(internal)* If applicable, the service manager notifies the metering manager to start the meter (to know how much to charge). This is not applicable if the contract is a fixed price.	None

DMTF-008	Provision Resources (from a contracted pool)
Variations	In lieu of a separate flow to authenticate the user, the request may contain the authentication credentials. A provider might require prepayment in addition to verifying that capacity is available. For example, the provider could provide prepaid tokens, which the service user might provide as part of the request. Capacity might not be available, so the request must be declined. There could be errors in one or more steps, leading to a failed process. This might trigger a different use case, such as Service Request Provisioning Failure. In virtualized infrastructure environments, the provisioning operation may simply be an allocation of capacity without specific resource selection.
Additional information	This process description is intentionally high level and abstract, and could apply to many types of services. It would be useful to develop at least two examples that are specific to one technology. The purpose of the examples are not to limit the possibilities of what a service may be, but rather to illustrate in a more tangible way how a provisioning process would work. Related use cases: • Establish Service Contract (see 10.3) • Administer Relationship (see 10.2) (for example, administer user identity/authentication credentials and request authorizations) • Contract Billing (see 10.6) • Monitor Service Resources (see 10.11)
Design notes	In the case of virtualized infrastructure, the act of provisioning may also include service package deployment (see the Deploy Service Template use case in 10.9).

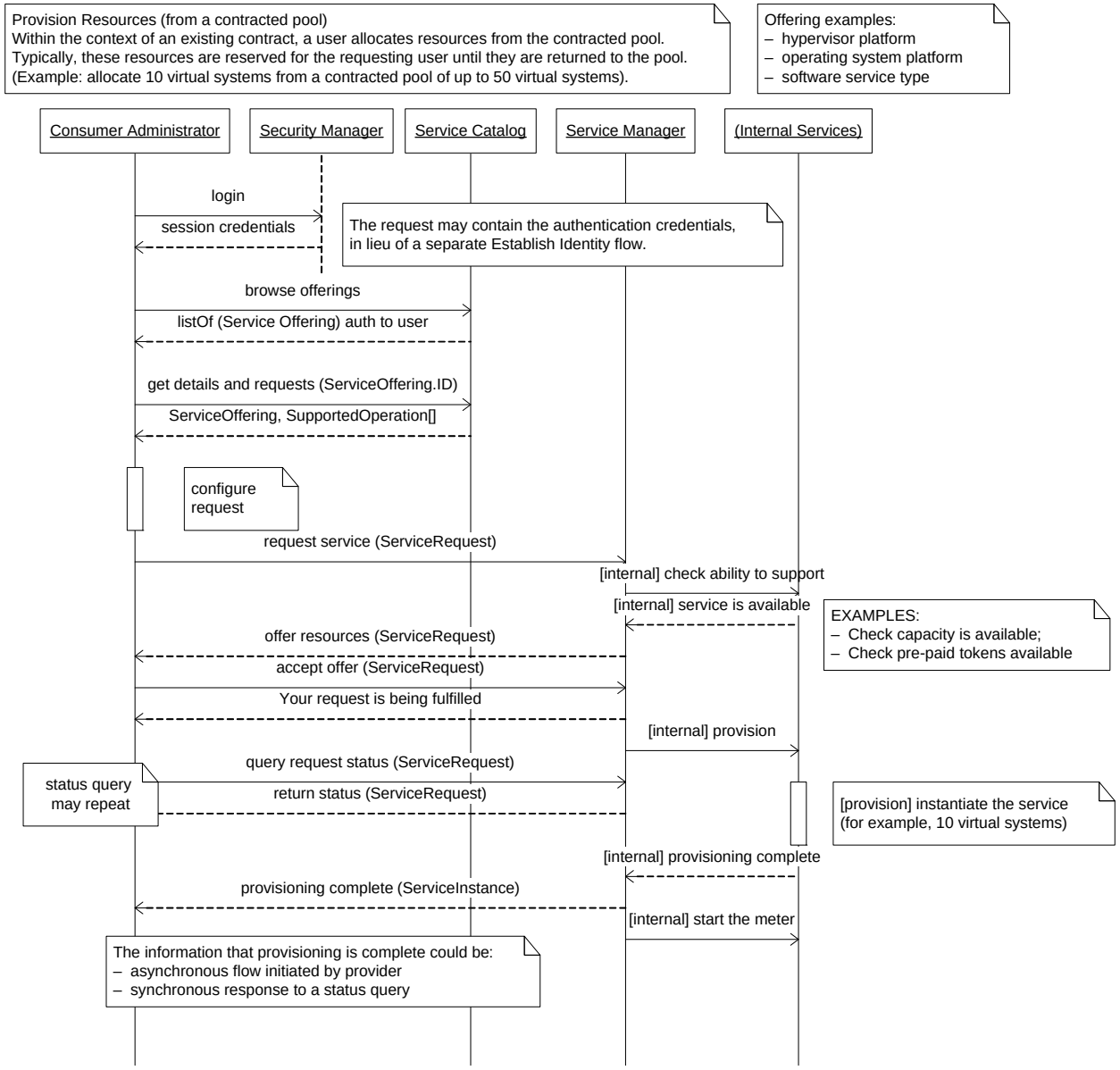


Figure 13 – Provision Resources

10.9 Deploy Service Template

DMTF-009	Deploy Service Template
Description	A cloud service consumer deploys a parameterized service template in the context of a service offering.
Desired outcome	A service instance is created after a successful deployment of an instance of the parameterized template.
Business events	A cloud service consumer decides to deploy a service template onto a service provider's cloud for consumption.

DMTF-009	Deploy Service Template	
Actors	• Service User For this use case, the Service User role and Consumer Service Administrator are combined in a single actor referred to as the Service User. • Cloud Service Provider Service Operations Manager and Transition Manager roles In this use case, the actions of these provider actors are performed by the service manager, provisioning manager, and metering manager, which are functions that are most likely automated.	
Involved components and services	• Identity and security • Service catalog • Service manager • Metering system (optional)	
Dependencies	• Mechanisms and protocols exist for upload or download of service templates (for example, software in a virtualized software stack — OVF is the recommended format) to the cloud service provider infrastructure. • The service catalog has a physical or logical representation or pointers to the service templates. • The cloud service provider has management infrastructure for deployment and provisioning. • The service consumer has previously provisioned resources.	
Process assumptions	• The service template has been created in a template catalog. • The service offering for the service template has been created in a service catalog. • Virtual resources on which to deploy the virtual image have been allocated to the consumer. • Virtual resources are configured pre- and post-installation as part of deployment as needed.	
Process flow	**Step description**	**Data or data artifact required**
	1. The service user (for example, a tester) logs in.	Established account, security profile, and credentials
	2. The service user browses (or searches) the service catalog to see which offerings are available. The service catalog filters the list of all offerings to show only those that this service user is authorized to request.	Service offering
	3. The service user selects an offering. If there are configuration options, such as specifying the number of resources that will be requested or specifying a security environment, they are displayed to the user.	• Service offering • Supported operation
	4. The service user configures a request and submits the request to the service manager for this type of service.	Service request

DMTF-009	Deploy Service Template	
	5. The service manager acknowledges that the request is being fulfilled. This acknowledgement could be repeated periodically, potentially with a progress indicator (for example, request is 64% complete).	None
	6. *(internal)* The service manager initiates the provisioning of the resources, such as triggering a provisioning manager and passing the parameters of the request.	None
	7. *(optional)* The service user may query for the status of the request and receive a response. This may repeat any number of times.	Service request
	8. *(internal)* The cloud service provider deploys the service template.	None
	9. The service manager notifies the service user that the provisioning has completed either successfully or unsuccessfully. This notification could be an asynchronous message sent to the user (for example, an email message) or an update to a feed that the service user periodically checks.	Service instance
Variations	Metering, billing, or both may be invoked during upload or deployment activities. If the service user does not locate an existing virtual image in the cloud service provider's service catalog that will serve their purpose, they may upload a new image. Deployment of virtual images may occur to a physical infrastructure (for example, the image is uploaded and run on the provider's infrastructure) or to one or more intermediary phases, such as the creation of a "design" that contains the virtual images from a service catalog, but they are not yet deployed into "production." Deployment of virtual images may be performed programmatically within the cloud service provider's infrastructure by the cloud consumer's existing services, programs, or images. Resources may not be available or be made available for deployment. Service templates may contain complex and nested topologies of multiple images, services, networks, and so on. A long delay could occur between the time when a request is made and when it is fulfilled. For example, the request could be an advance booking made weeks in advance of when it is needed.	
Additional information	Related use cases are as follows: • Create Service Offering (see 10.13) • Establish Service Contract (see 10.3) • Provision Resources (see 10.8)	

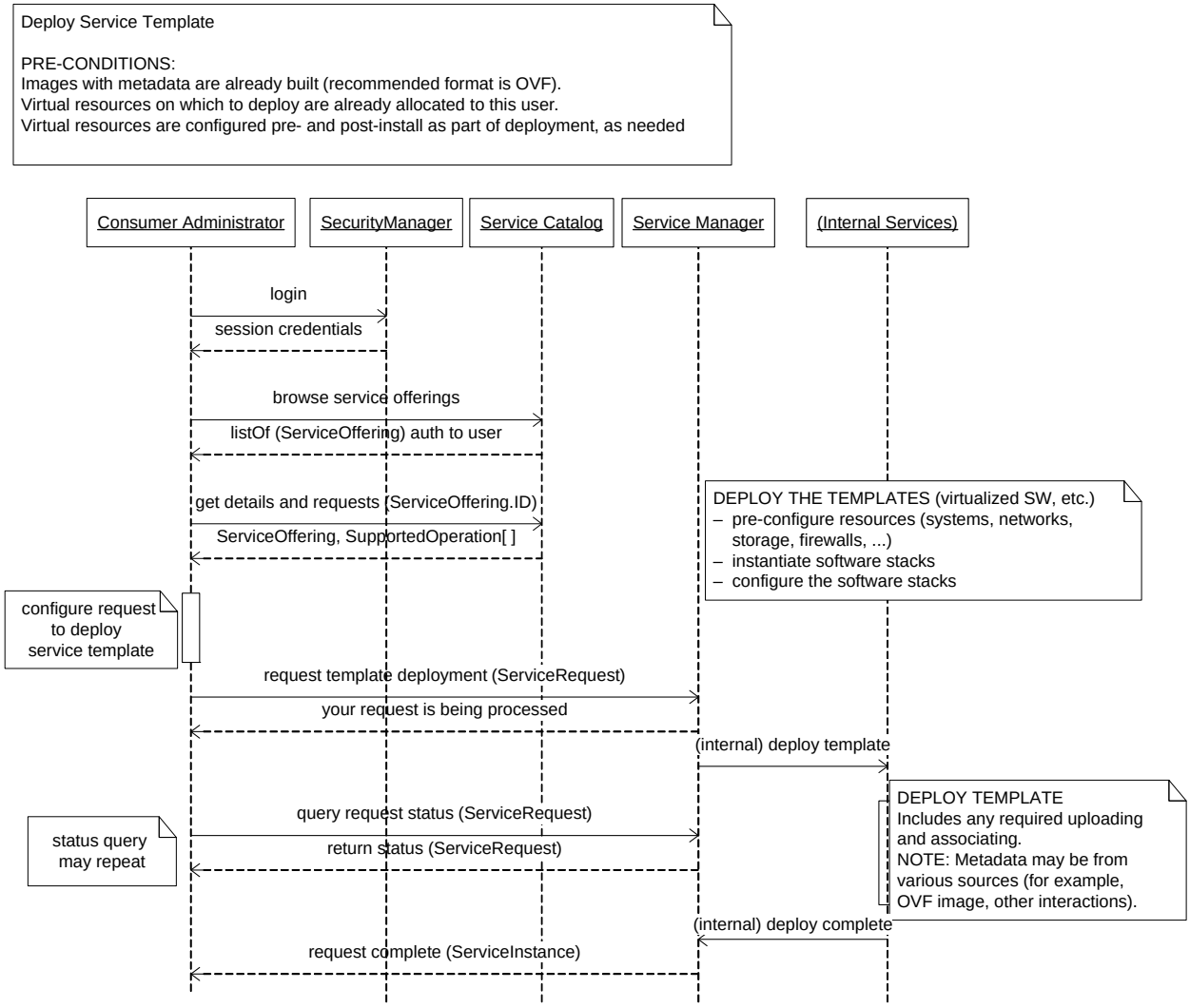


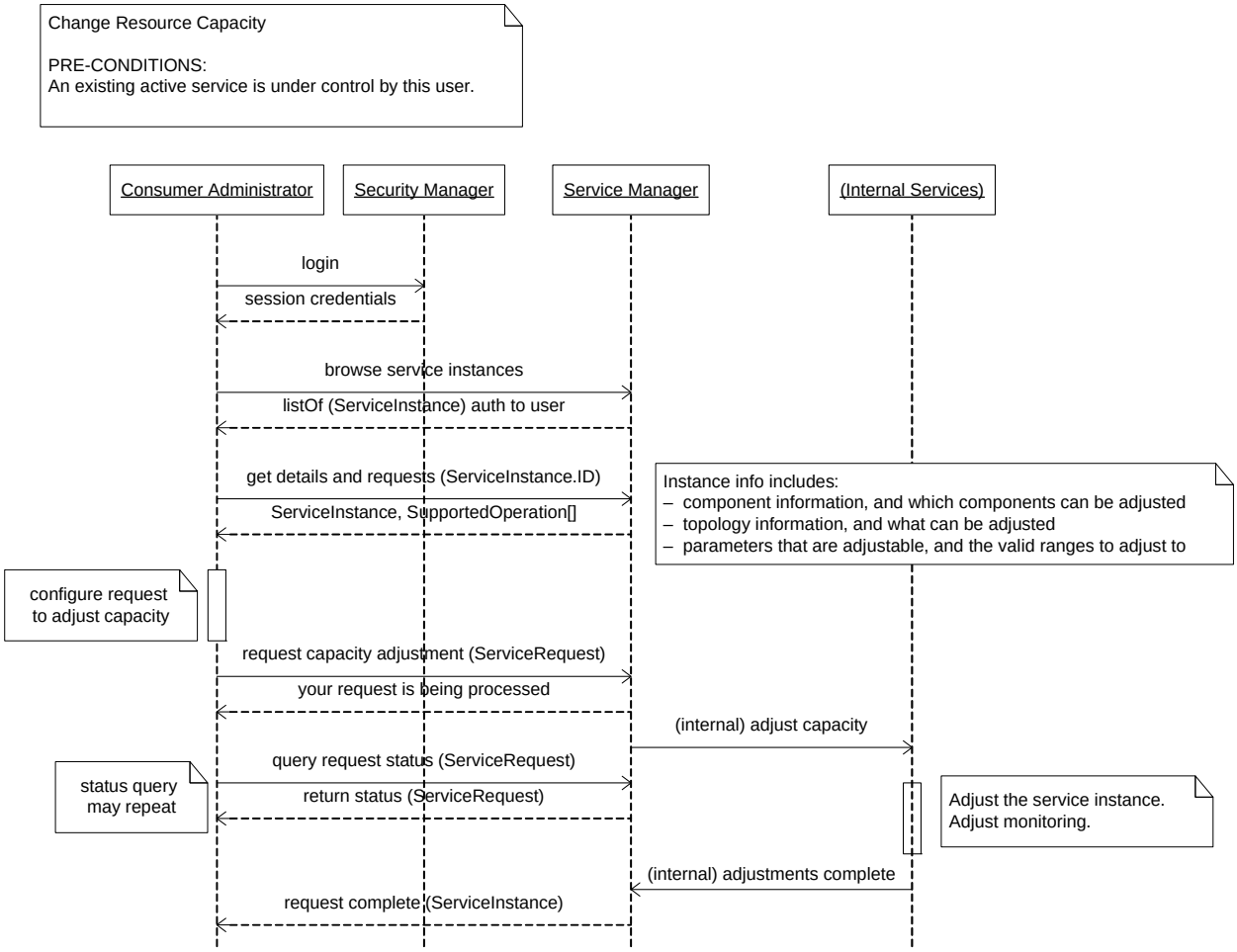
Figure 14 – Deploy Service Template

10.10 Change Resource Capacity

DMTF-010	Change Resource Capacity
Description	A cloud service consumer adds or changes the capacity or resources associated with a service instance, which is an instance of a service template. This can include adding or removing whole resources, or expanding or contracting resource limits associated with the service.
Desired outcome	The service has updated resources.
Business events	A cloud service consumer decides to expand or contract resources associated with a service based on use or another consideration.
Actors	For this use case, the Service User role and Cloud Service Consumer Service Administrator (CSA) are combined in a single actor referred to as the Service User.

<table>
<tr><th>DMTF-010</th><th colspan="2">Change Resource Capacity</th></tr>
<tr><td>Involved components and services</td><td colspan="2">• Service instance
• Service catalog</td></tr>
<tr><td>Dependencies</td><td colspan="2">• The service has been previously created based on a service catalog entry.
• The service user has the authority to consume or release capacity.</td></tr>
<tr><td>Process assumptions</td><td colspan="2">A service instance has been previously created</td></tr>
<tr><td rowspan="10">Process flow</td><th>Step description</th><th>Data or data artifact required</th></tr>
<tr><td>1. The service user logs in.</td><td>Established account, security profile, and credentials</td></tr>
<tr><td>2. The service user browses service instances.</td><td>Service instance</td></tr>
<tr><td>3. The service user gets the details about a service instance, including configurable parameters and supported requests.</td><td>• Service instance
• Supported operations</td></tr>
<tr><td>4. The service user configures the request to change based on the configurable parameters and submits the request to the service manager.</td><td>Service request</td></tr>
<tr><td>5. The service manager acknowledges that the request is being fulfilled.</td><td>None</td></tr>
<tr><td>6. (internal) The service manager initiates the reprovisioning of the resources. Reprovisioning involves testing capacity availability.</td><td>None</td></tr>
<tr><td>7. (optional) The service user may query for the status of the request and receive a response. This may repeat any number of times.</td><td>Service request</td></tr>
<tr><td>8. (internal) The cloud service provider makes the changes and adjusts monitoring.</td><td>None</td></tr>
<tr><td>9. The service manager notifies the service user that the provisioning has completed either successfully or unsuccessfully.

This notification could be an asynchronous message sent to the user (for example, an email message) or an update to a feed that the service user periodically checks.</td><td>Service instance</td></tr>
<tr><td>Variations</td><td colspan="2">• Sufficient capacity is not available to grant the request.
• Errors occur during the reprovisioning.</td></tr>
<tr><td>Additional information</td><td colspan="2">None</td></tr>
<tr><td>Design notes</td><td colspan="2">The configurable parameters in the service request could take various forms, such as a form that is filled out or providing a new service template definition. For example, if the service template defines a logical group of homogeneous virtual servers (for example, a group of homogeneous VirtualSystem elements in an OVFVirtualSystemCollection), the new template might adjust the number of virtual servers in the group or the capacity of each server in the group.</td></tr>
</table>



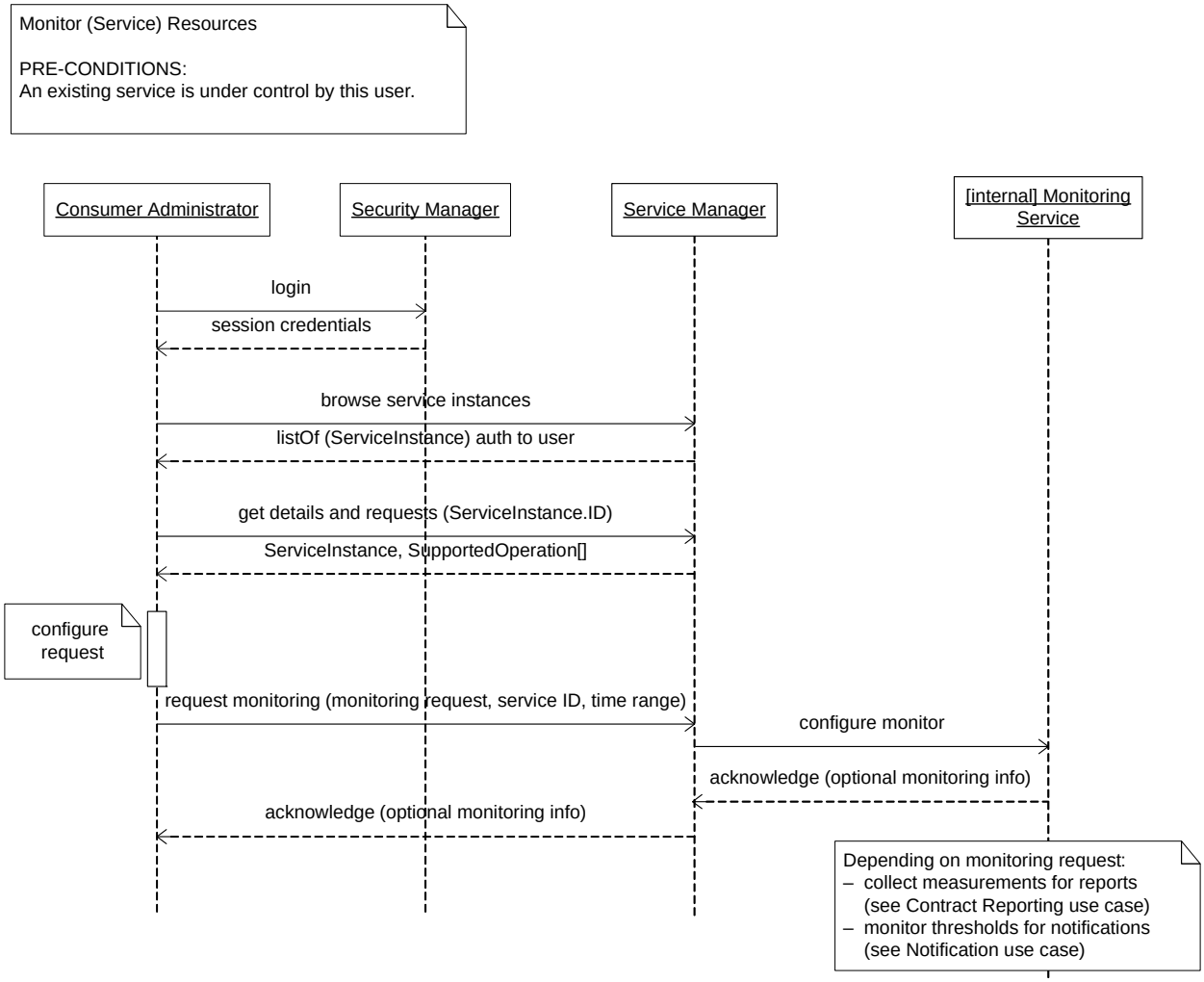
485

486 **Figure 15 – Change Resource Capacity**

487 10.11 Monitor Service Resources

DMTF-011	Monitor (Service) Resources
Description	A cloud consumer configures a monitor for a deployed service instance and resources that support the service instance. A monitor may collect data (for example, resource consumption, throughput, response times, or availability) or establish an exception threshold.
Desired outcome	Monitored data is recorded for later reporting or to generate a notification to the cloud consumer.
Business events	The cloud consumer's business or IT organizations issue reporting requests. Constraints are related to service levels or monitors. These may be in addition to formal SLA information.
Actors	Consumer Service Administrator (CSA)
Involved components and services	Service instance
Dependencies	The service has been previously created.

<table>
<tr><th>DMTF-011</th><th colspan="2">Monitor (Service) Resources</th></tr>
<tr><td>Process assumptions</td><td colspan="2">• The service has been previously created.
• A credential to allow retrieval of information exists.</td></tr>
<tr><td rowspan="7">Process flow</td><th>Step description</th><th>Data or data artifact required</th></tr>
<tr><td>1. The CSA logs in.</td><td>Established account, security profile, and credentials</td></tr>
<tr><td>2. The CSA browses service instances.</td><td>Service instance</td></tr>
<tr><td>3. The CSA gets the details about service instances, including configurable parameters and supported requests.</td><td>• Service instance
• Supported operations</td></tr>
<tr><td>4. The CSA submits a monitoring request to collect measurements (available in a report) or to monitor thresholds (which could result in report data or a notification). The request identifies one or more instantiated services or resources, and a time range</td><td>Monitoring request (described by a supported operation)</td></tr>
<tr><td>5. (internal) The service manager configures the internal monitoring service.</td><td>None</td></tr>
<tr><td>6. The monitoring request is acknowledged, along with optional monitor information (for example, a monitor ID).</td><td>Optional monitoring information</td></tr>
<tr><td>Variations</td><td colspan="2">None</td></tr>
<tr><td>Additional information</td><td colspan="2">None</td></tr>
<tr><td>Design notes</td><td colspan="2">None</td></tr>
</table>



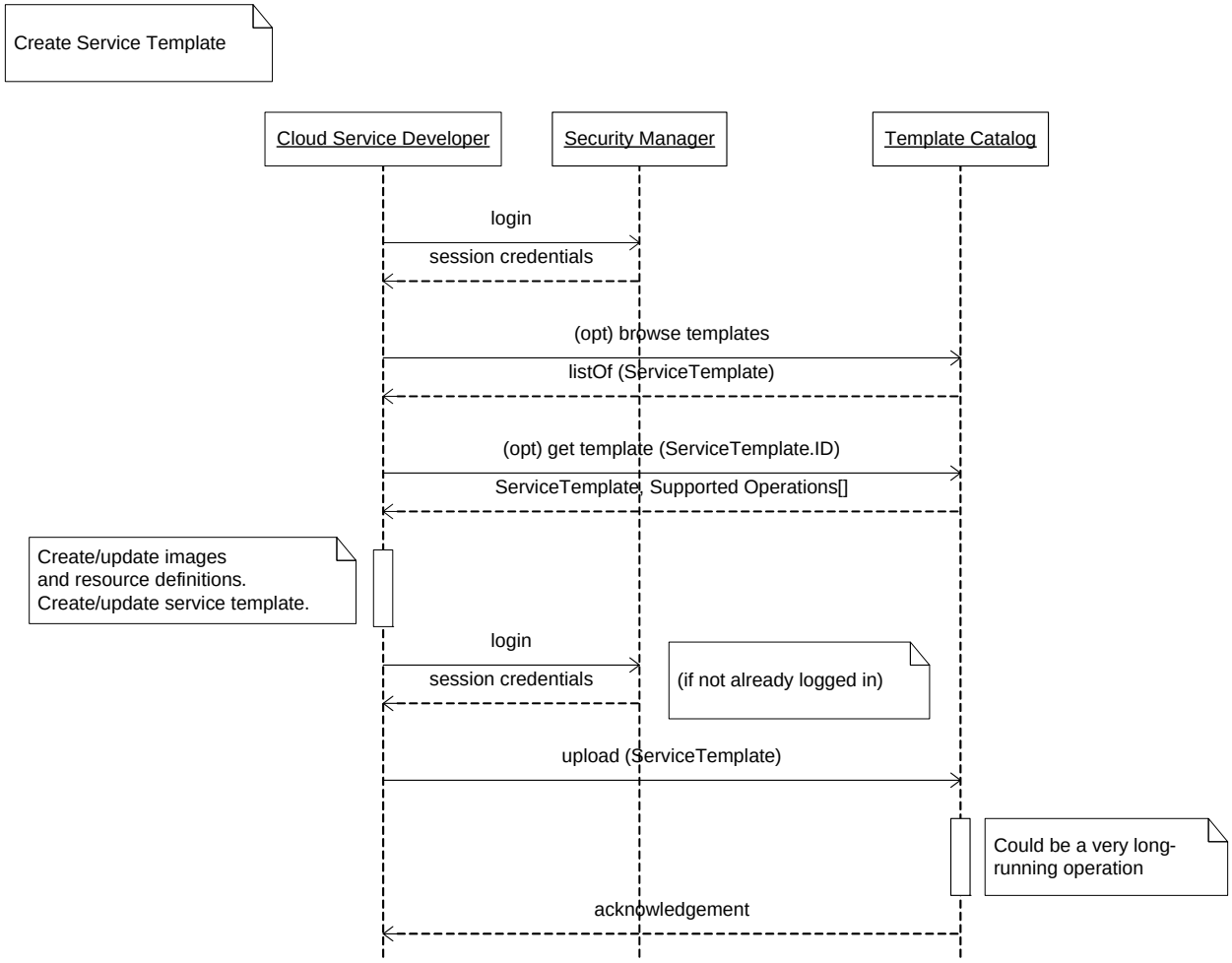
488

489 **Figure 16 – Monitor Service Resources**

490 ## 10.12 Create Service Template

DMTF-013	Create Service Template
Description	A cloud service developer creates a template of a service that may later be used to create an instance of a service.
Desired outcome	A template of a service is available to be cataloged and published as an available service that may be deployed by one or more cloud service consumers.
Business events	A cloud service provider decides for technical or business reasons to make new or updated services available to their cloud service consumers.
Actors	Service Developer
Involved components and services	Virtual machine images, packaging tools, image repository, or resources definitions and specifications

DMTF-013	Create Service Template	
Dependencies	• Virtual hard disks in an open industry format (VHD, VMDK, ISO, and so on) • Enumeration of resource capabilities and capacities • Enumeration of security model options • Licenses requirements definitions for all required components and operating systems • Storage to hold the newly created cloud service template	
Process assumptions	After this cloud service template has been created and verified, it will go on to the next step of being cataloged and published.	
Process flow	**Step description**	**Data or data artifact required**
	1. A cloud service developer decides to make a new service available.	A clear definition of the service to be offered and on which platforms and or resources to be made available
	2. *(optional)* The service developer logs in if they want to browse and download existing templates in the template catalog.	Established account, security profile, and credentials
	3. *(optional)* The service developer browses service templates.	Service template
	4. *(optional)* The service developer gets the details about service templates.	• Service template • Supported operations
	5. A service is created or updated using one or more resources and or images.	Images and or resource definitions
	6. A service template is created or updated.	Service template
	7. The service developer logs in, if not already logged in.	Established account, security profile, and credentials
	8. The service developer uploads a service template.	Service template
Variations	This use case has two major types. One set of templates are image based; the other type are resource templates.	
Additional information	This use case will usually be a prerequisite to the Create Service Offering use case described in 10.13.	
Design notes	None	



491

492 **Figure 17 – Create Service Template**

493 ## 10.13 Create Service Offering

DMTF-014	Create Service Offering
Description	The lifecycle of a new service offering is initiated and publicized for potential subsequent: • Advertisement • Contract assignment • Provisioning • Monitoring • Update • Consumption • Deletion
Desired outcome	Cloud service stakeholders are supplied with an inventory of consumable services available from a cloud service provider with the associated service metadata and descriptors. Service offerings may be public, private, or otherwise controlled by access or entitlement in the service catalog.

DMTF-014	Create Service Offering	
Business events	A cloud service stakeholder chooses to initiate a new service offering.	
Actors	• Cloud Service Developer (CSD) • Service Business Manager (SBM) • Consumer Service Administrator (CSA)	
Involved components and services	• Service catalog • Service publishing (push or pull) • Identity and security	
Dependencies	• The CSD has created a service template with associated metadata or service descriptors. • The service template is published to the template catalog. • Cloud service consumers use identity and security (authentication, authorization, access control, and entitlement) according to cloud service provider policy and associated service metadata • Cloud service consumers have a mechanism and access to (pull) or notification of (push) a service offering repository or catalog.	
Process assumptions	• A service operations manager provides a framework for definition of scope and infrastructure descriptors, parameters for the structure, and functionality of service offerings. • A taxonomy exists for service offering descriptions. • A service offering can be created from scratch or based on an existing template or service offering, or consumed from another cloud service provider. • Service SLAs, entitlement, billing, and contract terms are established and stored in the service offering artifact. • Format and protocols have been established for describing services and taxonomy for associated metadata. • The CSD may create a service on behalf of the cloud service provider (public service) or the cloud service consumer (private service). The Service Business Manager (SBM) would make the service available from the cloud service provider. The CSA would make the service available to the service users. • The CSD is able to create, test, and publish a service offering on an infrastructure (for example, within the construct of SaaS, PaaS, or IaaS layers). • A method exists for bundling or packaging the contents of and describing the service offerings.	
Process flow	**Step description**	**Data or data artifact required**
	1. The SBM logs in.	Established account, security profile, and credentials
	2. *(optional)* The SBM browses service offerings.	Service offering
	3. The SBM browses service templates.	Service template
	4. *(optional)* The SBM gets the details about service templates.	• Service template • Supported operations
	5. The SBM configures a service offering around a service template and submits a create request.	• Service template • Service offering

DMTF-014	Create Service Offering	
	6. The SBM activates the service offering, making it available for subscription.	None
Variations	New service offering information or notifications may be "pushed" to consumers of a repository or catalog in addition to being available on a "pull" basis. In that case, the CSA would be provided with "advertised service offerings." The cloud service provider SBM may not elect to offer and publicize service offerings. Note that people within the same business organization may perform both the Cloud Service Developer and Cloud Service Consumer (or Cloud Service Provider) roles.	
Additional information	Related use cases are as follows: • Create Service Template	
Design notes	None	

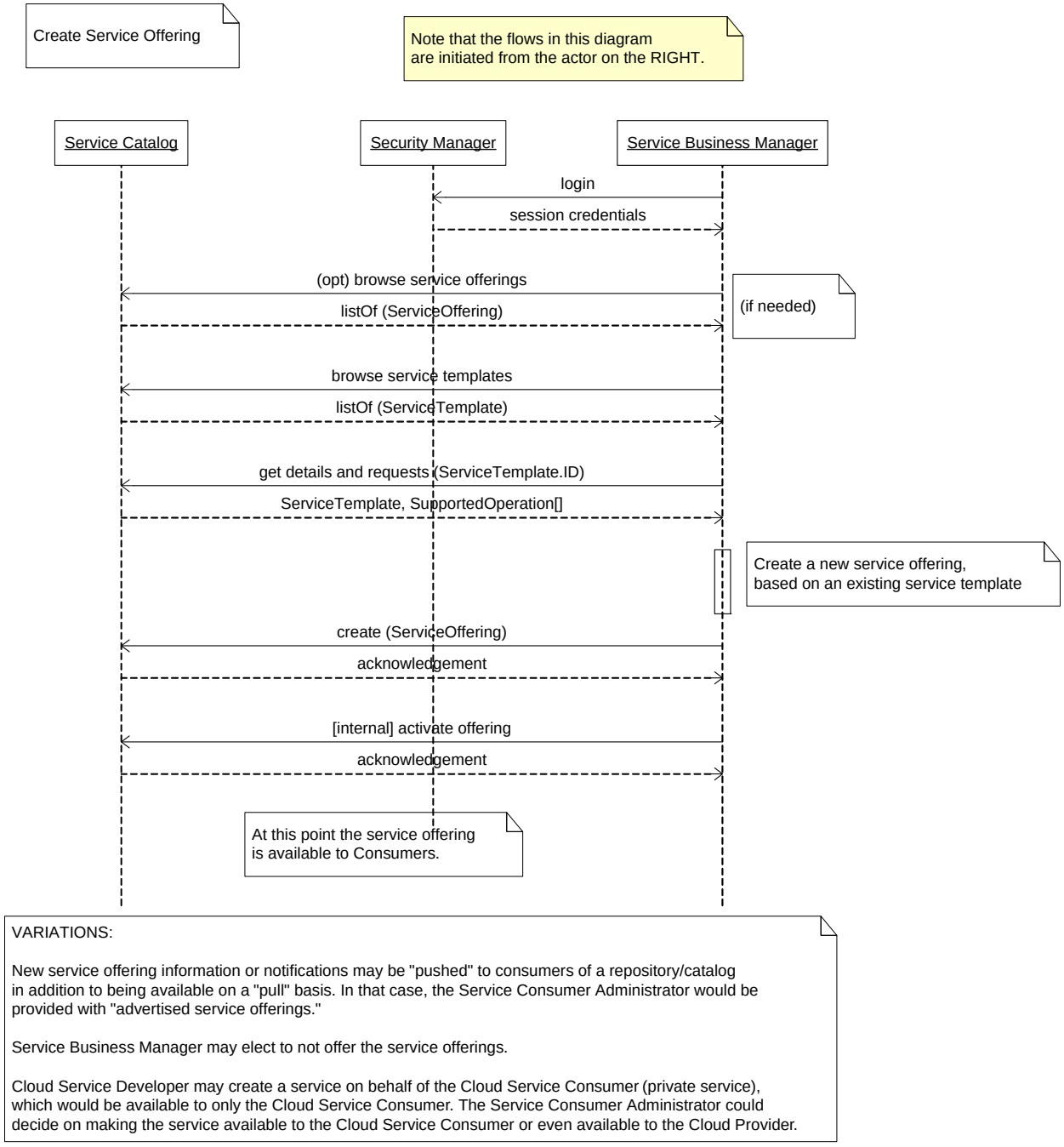
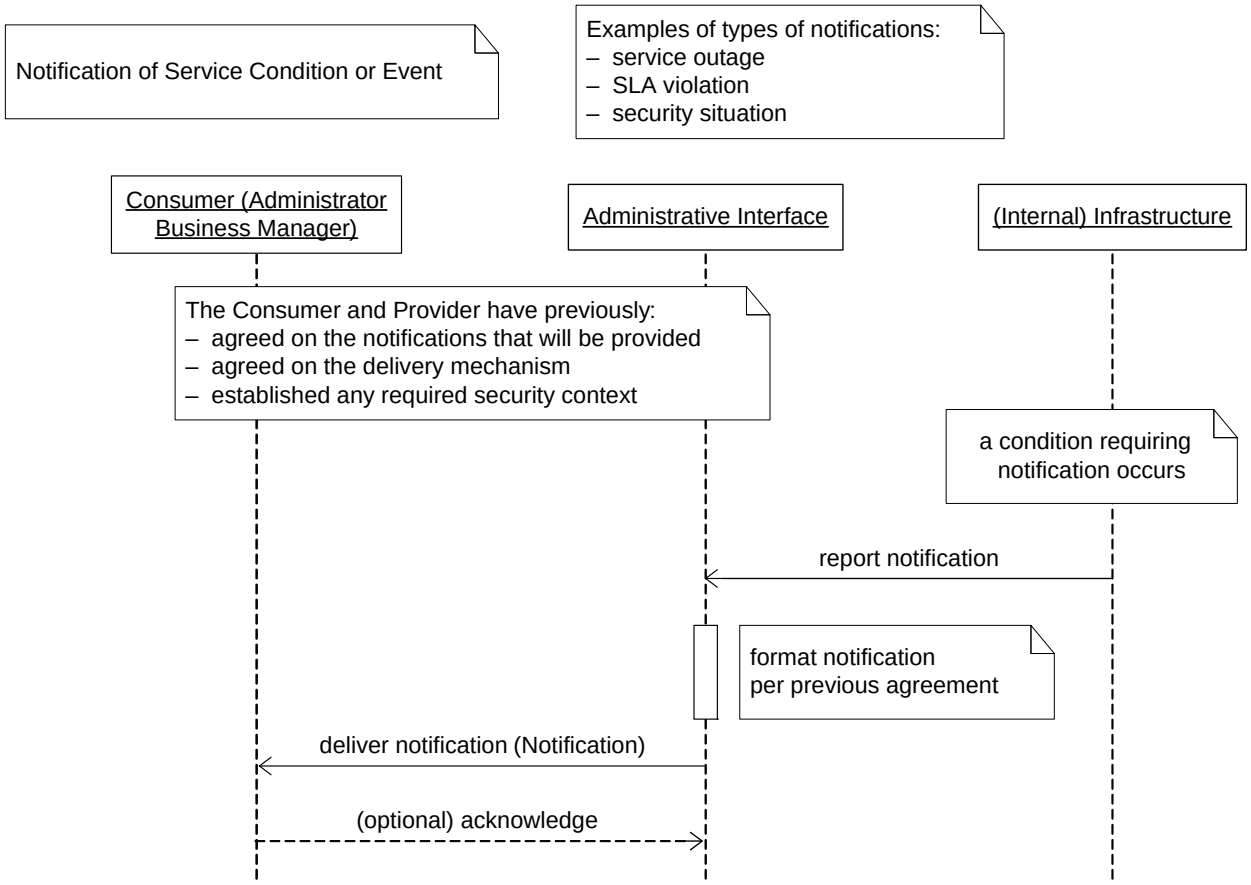


Figure 18 – Create Service Offering

10.14 Notification of Service Condition or Event

DMTF-015	Notification	
Description	A service has been configured and is in operation. Certain conditions or runtime operational events have been identified or detected that are significant enough to demand immediate notification of the condition or event to the service customer. An example is the detection of an intrusion or an unexpected configuration change.	
Desired outcome	Identified conditions generate an asynchronous event notification (including responses by cloud service providers to the event) to the notification target (normally the cloud service consumer).	
Business events	Any identified condition can trigger the event, but events are generally of two types: • Security events (triggered by IDS/IPS, such as DoS, malware, data exposure, data lost, VM attacks, or access violation) • Unexpected configuration changes (also security related) Notification may also occur when a contractual event, such as termination of the service, occurs.	
Actors	• Service Transition Manager • Service Operations Manager	
Involved components and services	• Condition monitors (IDS, malware detection, event/response logs, access logs, usage/health/user-experience monitors) • Service catalog (list of conditions that can be monitored) • Service request (list of conditions or events that need to be monitored and notified)	
Dependencies	The service must be configured and instantiated with the appropriate notification conditions provided. The cloud service provider must support the condition monitoring, and the service consumer must have a receiver listening for the notifications. (*Listening* would involve a receiver like a feed reader that periodically checks for new notifications.) The cloud service provider reviews service activation to ensure that requested conditions can be monitored, and then accepts the activation. When the service instance is created, the notification conditions then appear in the service instance.	
Process flow	**Step description**	**Data or data artifact required**
	1. During operation, a condition occurs that triggers a notification.	None
	2. A notification is delivered to the indicated target address.	Service notification
	3. *(optional)* Receipt of the notification is acknowledged.	None
Variations	Additional conditions could be added after instantiation, or trigger levels could be modified.	
Additional information	Presumably the cloud service provider would use various software tools to monitor for the indicated conditions. When the condition occurs, an asynchronous notification would be sent to the service consumer.	
Design notes	None	



497

498 **Figure 19 – Notification of Service Condition or Event**

11 Data Artifacts Overview

Figure 20 shows the data artifacts used in the use cases in section 10 that represent a service and its externally accessible elements throughout the lifecycle of the service. The properties in each artifact, their cardinality, and data type are suggestions that could form the base for a formal data model.

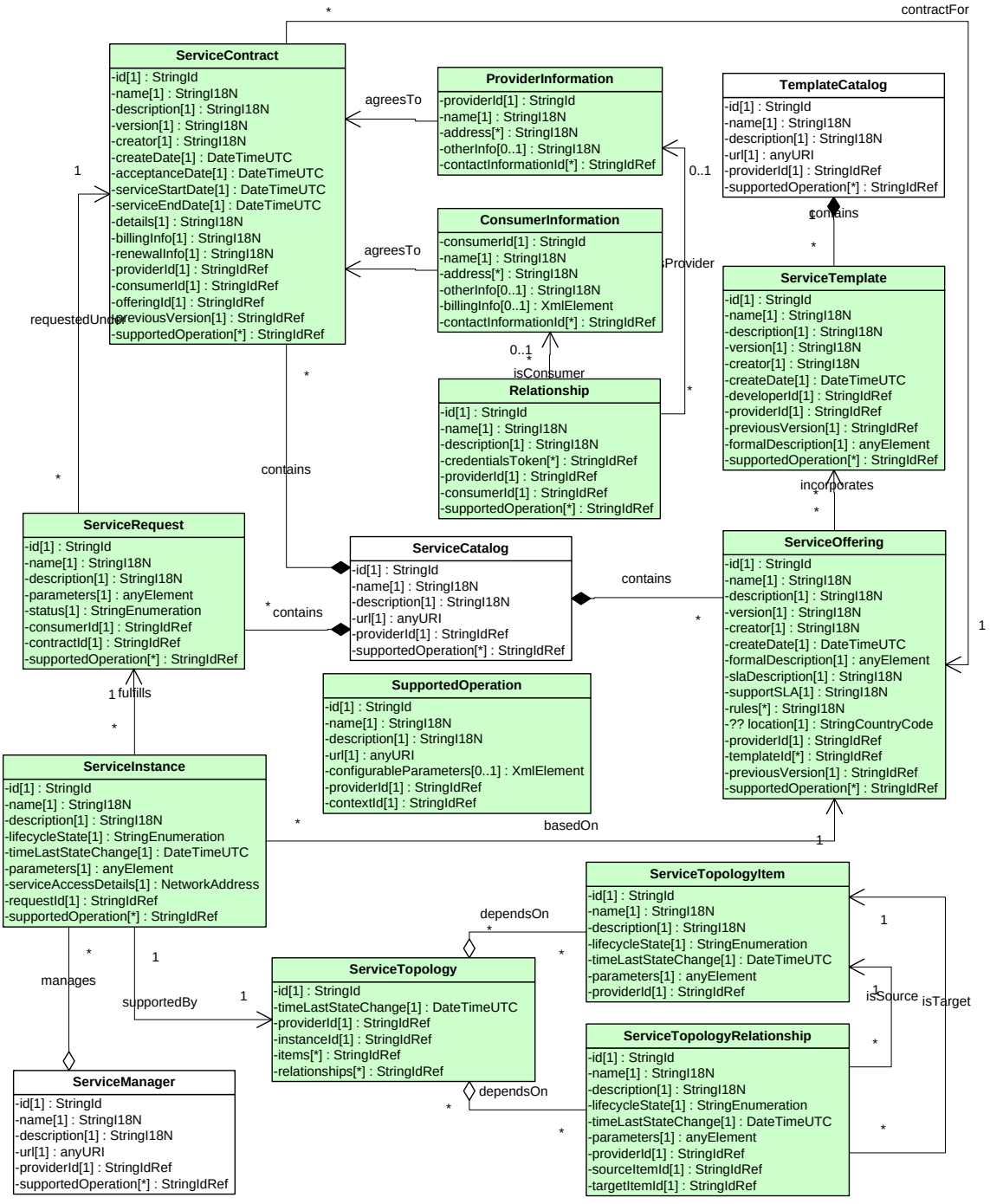
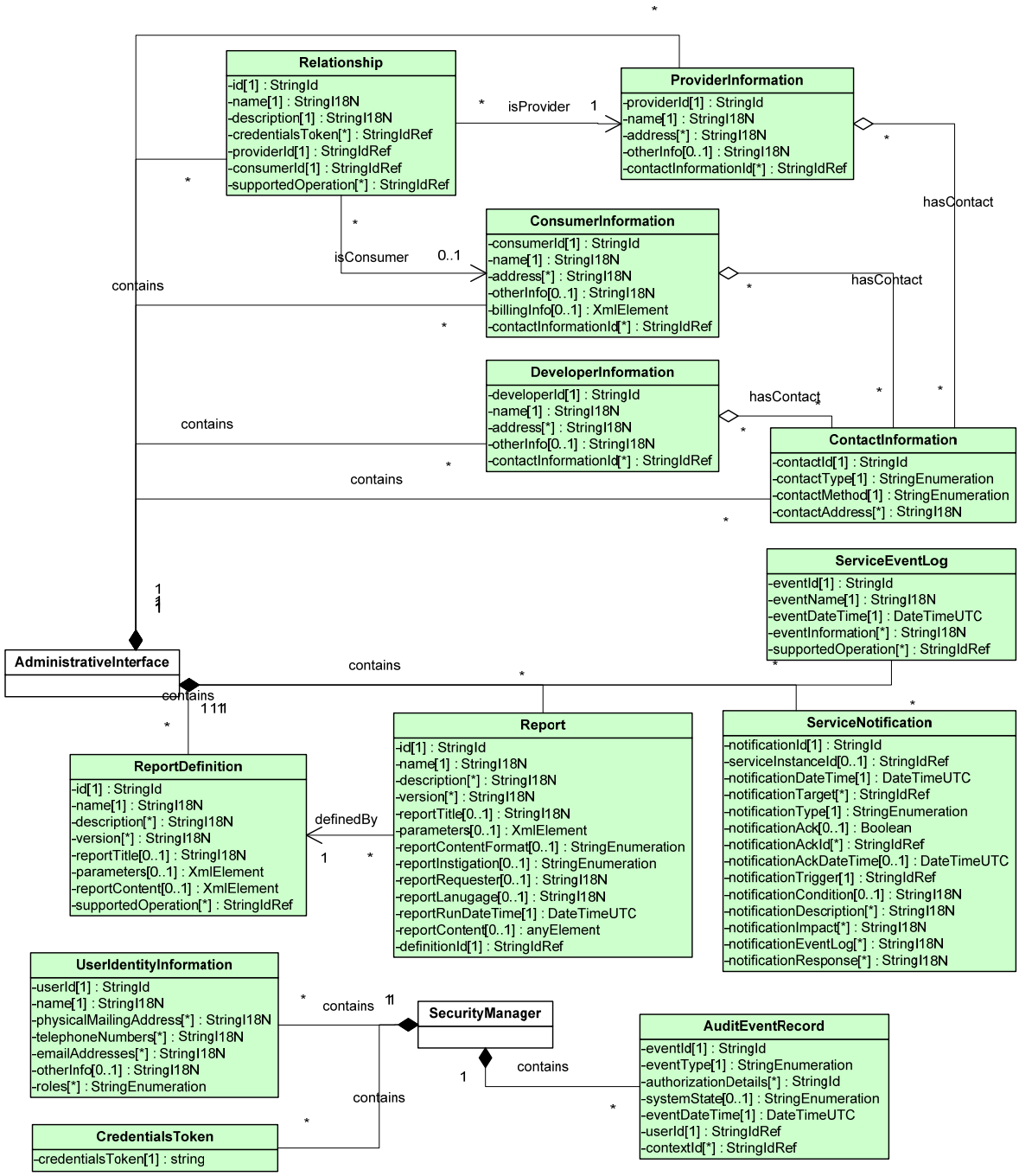


Figure 20 – Service Artifacts UML Diagram

505 Figure 21 shows data artifacts that represent the relationship between consumers, developers, and
506 providers, including the user and contact information.



507

508 **Figure 21 – Relationship Artifacts UML Diagram**

12 Data Artifact Details

The tables in this section describe artifacts used in the use cases in section 10. They contain more complete descriptions than the artifact UML diagrams (Figure 20 and Figure 21). These artifacts do not represent a formal data model, but rather are suggestions that could form the base of a formal model.

12.1 Service Template

Artifact	Service Template (Service Description)	
Description	A service template specifies the logical hardware, network configuration, and software for a service. A template may be instantiated many times, creating many instances of the service. Templates range from simple specifications of a single virtual machine to complex definitions involving many virtual machines, applications, and complex configurations. Typically, templates are instantiated with virtual entities, although physical entities may also be used. The specifications in a template may vary in their specificity. Some may indicate precisely what is to be instantiated, and others may provide higher-level descriptions that could be instantiated in many different ways. Service templates may be hierarchical and contain other service templates. Depending on circumstances, templates may be created by a consumer, provider, or third-party.	
Role in use cases	Service templates provide a precise description of the technical components of services. Therefore, they play an underlying role in the entire service lifecycle. Service templates are created, updated, and deleted. They are stored and made available in a template catalog. Service templates are components of the service offering. Templates may be added to an offering, modified while part of an offering, or removed from an offering. As part of an offering, they are also part of a contract. Additions or updates to a contract may result in the addition or modification of templates that are part of the service offerings in the contract. When resources are provisioned, provisioning is the result of a service request, which contains a service offering. The service templates in the service offering provide the technical description of the service and are used in provisioning and changing service capacity. During runtime, information contained in the template may play a role in monitoring, event notification, reporting, and billing, although runtime use cases focus on the service offering and instance that contain the template. Even terminating a contract may involve information contained in the template.	
Core Properties		
Name	**Content**	**Description**
id	String (ID)	Formal identifier for the template. The ID must be unique within the cloud service provider domain. Global uniqueness may be desirable.
name	String(i18n)	A human-friendly name of this artifact
description	String (I18N)	A human-friendly description of this artifact
version	String(i18n)	Identifies revisions of this artifact. Each version may have its own ID.
creator	String (i18n)	Name of the person, group, or agency that originally created this artifact. This property could include contact information, if useful.
createDate	DateTime	Date and time that this artifact was originally created
formalDescription	None	The machine-readable description of the service that is used to create an instance of the service. An OVF package, or an extension to an OVF package, has been proposed for this.

Referenced Objects		
Name	**Content**	**Description**
developerID	ID (Reference)	The ID of the cloud service developer
providerID	ID (Reference)	The ID of the cloud service provider
previousVersion	ID (Reference)	The ID of the previous version of this template
supportedOperation	ID (Reference) [*]	The IDs of all the supportedOperation objects available for this object

12.2 Service Offering

Artifact	Service Offering
Description	The basis for a service offering is a service template, which describes the logical hardware, network configuration, and software for a service. A service offering is an entity that typically is published in a service catalog. The service offering contains additional material that provides the context necessary to offer the service to a consumer, such as provisions for security, service level agreements, support and maintenance limitations, and billing. Service offerings may be “parameterized.” In other words, the consumer of a service described in a service offering may have choices in the instantiation of the service, such as a service level (for example, gold, silver, or bronze), types of security, number of virtual machines, machine capacities, operating systems, and database types. What is parameterized and how the choices affect the service contract are all characteristics of the provider’s technical and business model and the provider-consumer relationship. The requests that can be made with respect to a service offering, and the parameters of those requests, are defined in supported operation artifacts. In typical service catalog practices, service offerings are designed and created by cloud service providers and published in a service catalog. In some cases, for example a provider with only one or two offerings, the role of the service catalog is minor and the service catalog is almost invisible.
Role in use cases	When a user contracts for a cloud service, the user is accepting a service offering (possibly for a specific set of parameters), so service offerings play a role in all contract use cases. When services are provisioned, everything is done according to the service templates that are contained in the service offerings along with the parameters to the template that are set in the offering and the request. In the runtime maintenance phase, the service offering is not changed, but the way the service is monitored, the way notifications are generated, and service reporting are all affected by the terms set in the offering and the service templates contained in the offering. Service offerings are created, read, and updated in the course of the service lifecycle. When the offering is created, service templates are included. These templates may be modified later, if permitted by the service contract. Service offerings are published in the service catalog and may be deleted so that they are no longer available to request. The most important activity for a service offering is when it is accepted by a consumer with a service contract.

Core Properties		
Name	**Content**	**Description**
id	String (ID)	The formal identifier of the service offering. The ID is unique at cloud service provider’s level.
name	String(i18n)	A human-friendly name of this artifact

description	String (I18N)	A human-friendly description of this artifact
version	String(i18n)	Revisions of this artifact. Each version may have its own ID.
creator	String (i18n)	Name of the person, group, or agency that originally created this artifact. This property could include contact information, if useful.
createDate	DateTime	Date and time that this artifact was originally created
slaDescription	String (I18N)	A human-friendly description of the SLA
supportSLA	String (I18N)	Support arrangements and support SLA (for example, maintenance windows and procedures)
rules	Array of String (I18N) [*]	Rules that will be applied on the offering. Rules may include costing and billing rules, mediation rules, and arbitration rules.
notificationTriggers	Array of String (i18N) [*]	List of triggers for notifications to the service consumer. The triggers may include thresholds as well as metrics on which the trigger is based.
location	String (Country Code)	The physical location (country) of the service
Referenced Objects		
Name	**Content**	**Description**
previousVersion	ID (Reference)	The ID of the previous version of this service offering
providerID	ID (Reference)	The ID of the cloud service provider
templateID	String (Reference) [*]	The ID of the service template on which this service offering is based. The ID is unique at the cloud service provider's level.
requests	Array of IDs [*]	Each element of the array is an ID that refers to a supported operation that describes a valid request for this service offering.
supportedOperation	ID (Reference) [*]	The IDs of all the supportedOperation objects available for this object

12.3 Supported Operation

Artifact	**Supported Operation**	
Description	Each service offering may expose various requests that operate on the offering, such as provisioning resources made available through the offering or deploying a service package that implements the offering. A supported operation artifact describes each request supported by a service offering.	
Role in use cases	Users of the service provider interface discover and read the supported operations to know what requests can be formed and how to form them. Implementations could support a Create-Request action against each supported operation artifact that creates a blank service request of the appropriate form, which the user would then populate and submit.	
Core Properties		
Name	**Content**	**Description**
id	String (ID)	The formal identifier of the request type. The ID is unique within the scope of a service offering.

name	String(i18n)	A human-friendly name of this artifact
description	String (I18N)	A human-friendly description of this artifact
url	String	The URL to which the request should be submitted
configurableParameters	XML Element [0..1]	XML description of configurable parameters from which the consumer can choose when making the request. Each parameter would typically be qualified by characteristics such as 'property-name', 'enumeration or range of choices', 'default', and 'optional/required'.
Referenced Objects		
Name	**Content**	**Description**
providerId	ID (Reference)	The ID of the cloud service provider
offeringId	String (Reference)	The ID of the service offering to which this request applies

12.4 Service Contract

Artifact	**Service Contract**	
Description	A service contract is the contract between the cloud service provider and the cloud service consumer. A contract is based on one service offering, and it is the result of the negotiation between both sides.	
Role in use cases	The basic role of a service contract exists between a consumer and a provider. The technical details of the service are in the service offering, which becomes part of the contract and contains the service templates that apply to the contract. In addition to instances of the service offering, contracts document the business aspects of the relationship. Contracts are established, modified, and eventually terminated.	
Core Properties		
Name	**Content**	**Description**
id	String (ID)	The formal identifier of the service contract. The ID is unique at cloud service provider's level.
name	String(i18n)	A human-friendly name of this artifact
description	String (I18N)	A human-friendly description of this artifact
version	String(i18n)	Revisions of this artifact. Each version may have its own ID.
creator	String (i18n)	Name of the person, group, or agency that originally created this artifact. This property could include contact information, if useful.
createDate	DateTime	Date and time that this artifact was originally created
acceptanceDate	DateTime	The date when the contract was accepted
serviceStartDate	DateTime	The date when the service starts
serviceEndDate	DateTime	The date when the service ends
details	String (I18N)	Details about how the service is provided. This field may include domain name, IP address, URIs, entry points, and so on.
billing	String (I18N)	Billing and payment information
renewal	String (I18N)	Contract renewal information

Referenced Objects		
Name	**Content**	**Description**
previousVersion	ID (Reference)	The ID of the previous version of this service contract
providerId	ID (Reference)	The ID of the cloud service provider
consumerId	ID (Reference)	The ID of the cloud service consumer
offeringID	String (ID)	The identifier of the service offering on which this contract is based. The ID is unique at the cloud service provider's level.
supportedOperation	ID (Reference) [*]	The IDs of all the supportedOperation objects available for this object

12.5 Service Request

Artifact	Service Request	
Description	A service request is used by a consumer to request the provisioning, deployment, or modification of a service and its resources. The request is for instances of a service offering in the service catalog. The service offering may have several configuration parameters that can be selected by the consumer before submitting the request.	
Role in use cases	The service request is a data structure that is exchanged when requesting provisioning, deployment, or modification of service instances. It carries the consumer ID, service offering ID, and optionally, any selectable configuration parameters. The cloud service provider might want to negotiate the request with the consumer. For example, perhaps the provider can satisfy some but not all of the request at this time. The provider would return a modified service request showing what they can satisfy at this time. The consumer could then resubmit the request. The submitter sets the field to (the logical equivalent of) SUBMITTED when first submitting the request. The requester can query the Status field to determine when the request is complete and the result of the request. NOTE: The method is not yet determined. For example, if it is a RESTful interface, the immediate response to a request might be a URL that represents the service request object.	
Core Properties		
Name	**Content**	**Description**
id	ID (String)	The formal identifier of the service request. The ID is unique within the scope of a cloud service consumer (as defined by a token created during the Establish Relationship use case). The ID is used in operations to query the status of a possibly long-running request.
name	String (I18N)	A human-friendly name of the service request. The name would be appropriate to display to a cloud administrator or operator. I18N encoding would be expected.
description	String (I18N)	A human-friendly description of the service request. The name would be appropriate to display to a cloud administrator or operator. I18N encoding would be expected.

parameters	XML element	This is a very skeletal definition. Within these parameters would be found all the settings for the configurable service offering parameters. Examples of these parameters could be the number of systems to provision, the level of security required, and the dates and times the service needs to be active.
ConfigurableParameter-Settings	XML Element	XML description of the choices of configurable parameters from the service offering on which this request is based
Status	String (Enumeration)	The status of the request. Details are not yet determined. Possible examples are SUBMITTED, INVALID, PROCESSING, COMPLETED-GOOD, and COMPLETED-FAILED.
Referenced Objects		
Name	**Content**	**Description**
consumerId	ID (Reference)	The ID of the cloud service consumer
contractId	ID (Reference)	The ID of the service contract that is requested
instanceId	Array of IDs (Reference) [*]	The IDs of the service instances that are provisioned, if the request is satisfied
supportedOperation	ID (Reference) [*]	The IDs of all the supportedOperation objects available for this object

12.6 Service Instance

Artifact	Service Instance
Description	A service instance is a deployed instance (after a service request) of a service offering. A service instance is represented by a set of properties and a topology of nodes and links. This set is described in a service topology. Each node is a service topology item, such as a virtual system, a network, or an application instance. Each link is a service topology relationship, such as a dependency relationship between two service topology items. In a (possibly temporary) first simple model, service instances are not nested.
Role in use cases	In the lifecycle of a cloud service, as shown in Figure 4, a service instance is deployed after a consumer and provider enter into a contract, which in turn follows the establishment of a relationship between the consumer and provider. The service instance uses resources, such as computing, storage, and network resources. The resources may be provisioned as part of the instantiation of the instance, or they may be provisioned in a separate step. The instance may consist of one or more software images. These may be deployed as part of the instantiation of the instance, or they may be deployed in a separate step. The resources may be monitored and the capacity may be adjusted to meet the workload demand. When the consumer no longer needs the service, or the contracted time period has expired, the resources may be released through explicit operations or implicitly as a result of terminating the contract. The resources and images may be reprovisioned and redeployed, respectively, within the constraints of an established contract.

Core Properties		
Name	**Content**	**Description**
id	ID (String)	The formal identifier of the service instance. The ID is unique within the scope of a cloud service provider. *It is an open question whether the IDs are reused serially or are unique across space and time.*
name	String (I18N)	A human-friendly name of the service instance. The name would be appropriate to display to a cloud administrator or operator. I18N encoding would be expected.
description	String (I18N)	A human-friendly description of the service instance. The name would be appropriate to display to a cloud administrator or operator. I18N encoding would be expected.
lifecycleState	String (Enumeration)	The lifecycle state of the entire instance. Details are not yet determined. Possible examples are BEING_DEPLOYED, DEPLOYED_INACTIVE, and DEPLOYED_ACTIVE.
timeLastState-Change	DateTime	The day and time of the last lifecycle state change NOTE: This might be a placeholder. This model is probably too simplistic, though perhaps the complexity would be a separate object that contained, for example, the history of state changes.
parameters	XML element	This is a very skeletal definition. Within these parameters would be found all the defining characteristics of the service instance. *It is an open question whether the parameters are defined within the Service Instance versus in a separate type to which the Service Instance refers. There are parallels within the OVF and SVPC Profile definitions.*
serviceAccess Details	Network address (for example, a URL or IP address)	The network address (for example, a URL) at which the provisioned service can be accessed *It is an open question whether this is the network address of the service manager or the service itself.*
serviceNotifica-tionTriggers	Array of strings	List of triggers that will cause notifications to be sent to the consumer of the service instance. Each trigger should have an ID that is unambiguous within the scope of the service instance.
Referenced Objects		
Name	**Content**	**Description**
requestId	ID (Reference)	The ID of the service request that deployed this service instance
topologyID	ID (String)	The ID of the service topology that implements the service instance
supportedOp-eration	ID (Reference) [*]	The IDs of all the supportedOperation objects available for this object

519 12.7 Service Topology

Artifact	Service Topology
Description	A service topology represents the mapping between a service instance and the topology of nodes and links that implement it. Each node is a service topology item, such as a virtual system, a network, or an application instance.

	Each link is a service topology relationship, such as a dependency relationship between two service topology items. In a (possibly temporary) first simple model, service instances are not nested.	
Role in use cases	The service topology is usually changed indirectly based on an operation on a service instance which results in the addition of either an item or a relationship.	
Core Properties		
Name	**Content**	**Description**
id	ID (String)	The formal identifier of the service topology. The ID is unique within the scope of a cloud service provider. *It is an open question whether the IDs are reused serially or are unique across space and time.*
timeLastStateChange	DateTime	The day and time of the last lifecycle state change NOTE: This might be a placeholder. This model is probably too simplistic, though perhaps the complexity would be a separate object that contained, for example, the history of state changes.
Referenced Objects		
Name	**Content**	**Description**
providerId	ID (Reference)	The ID of the cloud service provider
instanceId	ID (Reference)	The ID of the service instance that this topology supports
items	Array of IDs [*]	Each element of the array is an ID that refers to a service topology item that is part of the service topology. A service topology item may be in more than one service topology.
relationships	Array of IDs [*]	Each element of the array is an ID that refers to a service topology relationship that is part of the service topology. A service topology relationship may be in more than one service topology.

12.8 Service Topology Item

Artifact	Service Topology Item
Description	A service topology item represents a node in a service instance topology. Each service topology item represents a component, such as a virtual system, network, or application instance. The service topology item is rather abstract. It may be subclassed in later versions of this document, or it may be left abstract, with the specification containing only the common properties and operations that apply to most any item.
Role in use cases	A service topology item will typically be created or assigned to a service (instance) topology not by an explicit 'create' operation against an item factory but rather as a derivative action of establishing a contract, provisioning resources, deploying images, or modifying capacity. Most aspects of a service topology item are implementation dependent, including: • The type of each service topology item • The level of detail available • The availability of operations directly on the item, and the specific capabilities of

	those operations • The lifecycle of each item, known state transitions, how the state transitions are affected, and so on	
Core Properties		
Name	**Content**	**Description**
id	ID (String)	The formal identifier of the service topology item. The ID is unique within the scope of a cloud service provider. *It is an open question whether the IDs are reused serially or are unique across space and time.*
name	String (I18N)	A human-friendly name of the service topology item. The name would be appropriate to display to a cloud administrator or operator. I18N encoding would be expected.
description	String (I18N)	A human-friendly description of the service topology item. The name would be appropriate to display to a cloud administrator or operator. I18N encoding would be expected.
lifecycleState	String (Enumeration)	The lifecycle state of the item. Details are not yet determined. Possible examples are OFFLINE, READY, ONLINE, and DEGRADED.
timeLastStateChange	DateTime	The day and time of the last lifecycle state change NOTE: This might be a placeholder. This model is probably too simplistic, though perhaps the complexity would be a separate object that contained, for example, the history of state changes.
parameters	XML element	This is a very skeletal definition. Within these parameters would be found all the defining characteristics of the item.
Referenced Objects		
Name	**Content**	**Description**
providerId	ID (Reference)	The ID of the cloud service provider
Extended Properties		
Name	**Content**	**Description**
NOTE: This section, if it exists at all, would be refined per item type.		

12.9 Service Topology Relationship

Artifact	**Service Topology Relationship**
Description	A service topology relationship represents a link in a service instance topology. Each service topology relationship represents a relationship between two items, such as runsOn, uses, dependsOn, and so on.

Role in use cases	A service topology relationship will typically be created or assigned to a service (instance) topology not by an explicit 'create' operation against a relationship factory but rather as a derivative action of establishing a contract, provisioning resources, deploying images, or modifying capacity. Most aspects of a service topology relationship are implementation dependent, including: • The type of each service topology relationship • The level of detail available • The availability of operations directly on the relationship, and the specific capabilities of those operations • The lifecycle of each relationship, known state transitions, how the state transitions are affected, and so on

Core Properties		
Name	**Content**	**Description**
id	ID (String)	The formal identifier of the service topology relationship. The ID is unique within the scope of a cloud service provider. *It is an open question whether the IDs are reused serially or are unique across space and time.*
name	String (I18N)	A human-friendly name of the service topology relationship. The name would be appropriate to display to a cloud administrator or operator. I18N encoding would be expected.
description	String (I18N)	A human-friendly description of the service topology relationship. The name would be appropriate to display to a cloud administrator or operator. I18N encoding would be expected.
lifecycleState	String (Enumeration)	The lifecycle state of the relationship. Details are not yet determined. Possible examples are PLANNED, PENDING, and ONLINE.
timeLastStateChange	DateTime	The day and time of the last lifecycle state change NOTE: This might be a placeholder. This model is probably too simplistic, though perhaps the complexity would be a separate object that contained, for example, the history of state changes.
parameters	XML element	This is a very skeletal definition. Within these parameters would be found all the defining characteristics of the relationship.

Referenced Objects		
Name	**Content**	**Description**
providerId	ID (Reference)	The ID of the cloud service provider
sourceItem	ID (Reference)	The ID of the service instance item that is the "from" source of the relationship, in the style of <source> <relationshipName> <target> Example: Service_ID=A dependsOn System_ID=B
targetItem	ID (Reference)	The ID of the service instance item that is the "to" target of the relationship, in the style of <source> <relationshipName> <target>. Example: Service_ID=A dependsOn System_ID=B

Extended Properties		
Name	**Content**	**Description**
NOTE: This section, if it exists at all, would be refined per relationship type.		

12.10 Consumer Information

Artifact	Consumer Information	
Description	Consumer information is the identifying information that is provided by the cloud consumer when a relationship is established.	
Role in use cases	Consumer information is used in establishing and administering relationships. It may be created or modified in conjunction with the use cases.	
Core Properties		
Name	**Content**	**Description**
consumerId	ID (String)	A globally unique identifier of a consumer. It is used to establish namespaces in which other fields are unique.
name	String (I18N)	A human-friendly name of the consumer
address	String (I18N) [*]	Human-friendly addresses of the consumer
otherInfo	String (I18N) [0..1]	Other identifying information This could be used for something like an Open ID.
billingInfo	XML Element [0..1]	Billing and payment information This is probably a complex structure.
Referenced Objects		
Name	**Content**	**Description**
contactInformationId	ID (Reference) [*]	The IDs of various contact points into the consumer, such as administrative contact and technical contact

12.11 Developer Information

Artifact	Developer Information	
Description	Developer information is the identifying information that is provided by the cloud service developer when a relationship is established.	
Role in use cases	Developer information is used in establishing and administering relationships. It may be created or modified in conjunction with the use cases.	
Core Properties		
Name	**Content**	**Description**
developerId	ID (String)	A globally unique identifier of a cloud service developer. It is used to establish namespaces in which other fields are unique.
name	String (I18N)	A human-friendly name of the developer
address	String (I18N) [*]	Human-friendly addresses of the developer
otherInfo	String (I18N) [0..1]	Other identifying information This could be used for something like an Open ID.

Referenced Objects		
Name	**Content**	**Description**
contactInformationId	ID (Reference) [*]	The IDs of various contact points into the cloud service developer, such as administrative contact and technical contact

524 12.12 Provider Information

Artifact	Provider Information	
Description	Provider information is the identifying information that is provided by the cloud service provider when a relationship is established.	
Role in use cases	This information is created before use cases go into effect. The consumer must learn about this information by means not specified in this document.	
Core Properties		
Name	**Content**	**Description**
providerId	ID (String)	A globally unique identifier of a cloud service provider. It is used to establish namespaces in which other fields are unique.
name	String (I18N)	A human-friendly name of the provider
address	String (I18N) [*]	Human-friendly addresses of the provider
otherInfo	String (I18N) [0,.1]	Other identifying information This could be used for something like an Open ID.
Referenced Objects		
Name	**Content**	**Description**
contactInformationId	ID (Reference) [*]	The IDs of various contact points into the cloud service provider, such as administrative contact and technical contact

525 12.13 Relationship (Between Organizations)

Artifact	Relationship	
Description	This artifact represents an organizational relationship (not to be confused with a service topology relationship) between a consumer and a provider. This relationship would typically be a business relationship.	
Role in use cases	This relationship is created in the Establish Relationship use case.	
Core Properties		
Name	**Content**	**Description**
id	ID (String)	A globally unique identifier of a relationship. It is used to establish namespaces in which other fields are unique.
name	String (I18N)	A human-friendly name of the relationship
description	String (I18N)	A human-friendly description of this artifact
credentialsToken	String (I18N) [*]	Human-friendly addresses of the cloud service provider

Referenced Objects		
Name	**Content**	**Description**
providerID	ID (String)	A globally unique identifier of a cloud service provider
consumerID	ID (String)	A globally unique identifier of a consumer
credentialsToken	ID (Reference) [*]	The IDs of various CredentialTokens
supportedOperation	ID (Reference) [*]	The IDs of all the supportedOperation objects available for this object

526 ## 12.14 Contact Information

Artifact	Contact Information	
Description	Contact information is the information required for persons or agencies who are active in these use cases to receive messages and reports and execute their roles in the service. This artifact recognizes that some information will be required but does not attempt to specify what information will be needed in any specific situation.	
Role in use cases	Contact information is created at or before the time when relationships are created. Maintenance of contact information is not covered in this document.	
Core Properties		
Name	**Content**	**Description**
contactId	ID (String)	A unique identifier of this consumer contact
contactType	String (Enumeration)	Examples are Administrative, Technical, and Billing.
contactMethod	String (Enumeration)	Indicates how a provider will contact the consumer with some message related to the administration of a contract. Examples are email message, secure message box, web page hosted by the consumer, phone call, and so on.
contactAddress	String (I18N) [*]	Address details for the administrative contact. Examples are an email address and a web page URL. NOTE: This is an oversimplification.

527 ## 12.15 Credentials Token

Artifact	Credentials Token	
Description	A credentials token is returned by the cloud service provider after a relationship has been established or a new user is created by the consumer. This token may be used in future transactions to uniquely identify the consumer or its designated users. The token may be temporary and need to be renewed, or it may be permanent.	
Role in use cases	A credentials token is created by the cloud service provider and used in establishing and administering relationships.	
Core Properties		
Name	**Content**	**Description**
credentialsToken	String (Opaque?)	The credential token can take various forms: • User ID • Email address

		• User name / password • Temporary token • SAML credential • Client certificate • Open ID • Cloud Service Provider specific ID

12.16 User Identity Information

Artifact	User Identity Information	
Description	User identity information is the identifying information that is provided by the cloud consumer or its designated users when a relationship is administered. This artifact describes one user of the cloud service provider interface, such as a system administrator or a business administrator. User IDs are created or modified by the consumer. Users are associated with all the operations performed on the cloud resources on behalf of the consumer.	
Role in use cases	Credentials tokens are created, modified, and deleted in establishing and administering relationships.	
Core Properties		
Name	**Content**	**Description**
userId	ID (String)	A unique identifier of a user in the namespace of the consumer
name	String (I18N)	A human-friendly name of the consumer
physicalMailingAddress	String (I18N) [*]	Consumer's physical mailing address
telephoneNumbers	Array of String (I18N) [*]	Consumer's phone numbers
emailAddresses	Array of String (I18N) [*]	Consumer's email addresses
otherInfo	String (I18N) [0..1]	Other identifying information This could be used for something like an Open ID.
roles	Array of String (I18N) [*]	Roles this user can perform. It implies access level to various resources and operations within the scope of the consumer cloud space.

12.17 Report Definition

Artifact	Report Definition	
Description	A report definition describes the input parameters and output content of a report. A report can be of many types, including but not limited to: • Service usage (overall, per user) • Consumption vs. contracted limits • Availability vs. contracted levels • Performance / SLA compliance • SLA violations • Committed vs. available (burstable) • Metrics per operational tasks (for example, quantity, response time) • Operational task activity audit log • Security incidents The report may combine information from multiple artifacts, such as the agreed-upon performance metrics from an SLA and the measured performance. The report may be limited to events or states that existed during a given time period. Reports are generally designed and implemented by the service operations manager of the cloud service provider. The reports are typically requested and received by the consumer service administrator of the cloud service consumer.	
Role in use cases	Report definitions may be created at any time and would be made available to the consumer. They could be stored in the service catalog, perhaps linked to service contracts, offerings, and instances. They might be made available through some other means, such as when a consumer forms a relationship with a provider.	
Core Properties		
Name	**Content**	**Description**
id	ID (String)	A unique identifier of this artifact
name	String(i18n)	A human-friendly name of this artifact
description	String (I18N)	A human-friendly description of this artifact
version	String(i18n)	Revisions of this artifact. Each version may have its own ID.
parameters	XML element	This is a very skeletal definition. Within these parameters would be found all the options selectable by the report requester. Examples are the resources (for example, contracts, service instances, service items) to include, the date and time range, and the supported output formats (for example, XML, HTML, comma-separated variables, formatted text, PDF).
outputContent	XML Element	This is a very skeletal definition. This element describes the content of the report, including the report type and descriptions of which data (for example, the column headings and formats) is included. This could also be a reference to an external file or other entity rather than embedded in this artifact.

Referenced Objects		
Name	**Content**	**Description**
supportedOperation	ID (Reference) [*]	The IDs of all the supportedOperation objects available for this object

12.18 Report

Artifact	Report	
Description	A report is the output derived from the combination of a report definition and the parameters configured by the requester. A report can be of many types, including but not limited to: • Service usage (overall, per user) • Consumption vs. contracted limits • Availability vs. contracted levels • Performance / SLA compliance • SLA violations • Committed vs. available (burstable) • Metrics per operational tasks (for example, quantity, response time) • Operational task activity audit log • Security incidents Reports are generally designed and implemented by the service operations manager of the cloud service provider. The reports are typically requested and received by the consumer service administrator of the cloud service consumer.	
Role in use cases	Reports are established and governed by the service contract and may be specified in the service offering and affected by technical details in the service template and service instance. Changing the contents of a report may require an administrative change to the service contract or may be allowed under terms of the contract, offering, and instance. Reports are requested in the Contract Reporting use case, or they may be scheduled automatically.	
Core Properties		
Name	**Content**	**Description**
id	ID (String)	A unique identifier of this artifact
name	String(i18n)	A human-friendly name of this artifact
description	String (I18N)	A human-friendly description of this artifact
version	String(i18n)	Revisions of this artifact. Each version may have its own ID.
reportTitle	String	Human-readable descriptive title of the report
parameters	XML element	This is a very skeletal definition. Within these parameters would be found all the options selectable by the report requester. Examples are the resources (for example, contracts, service instances, service items) to include, the date and time range, and the supported output formats (for example, XML, HTML, comma-separated variables, formatted text, PDF).
reportContentFormat	String	The form in which the report is expressed (for example, XML, HTML, comma-separated variables, formatted text, PDF)

reportInstigation	String	How the report is started: • By request • Scheduled • Passed threshold
reportRequester	ID (String)	If the report is requested, the ID of the person or entity requesting the report
reportLanguage	String	The human language in which the report is expressed. Suggest xml:lang attribute.
reportRunDateTime	DateTime	Date and time that the report was run
reportContent	String	Structured data in the designated format. This could be a reference to an external file or other entity rather than embedded in this artifact.
Referenced Objects		
Name	**Content**	**Description**
definitionId	ID(Reference)	The ID of the report definition on which this report is based

12.19 Service Event Log

Artifact	**Service Event Log**	
Description	A service event log contains logged information about a predetermined set of events. This information can be used for problem diagnostics and to compare against contract-defined SLAs.	
Role in use cases	Service event logging is established and governed by the service contract and may be specified in the service offering and affected by technical details in the service template and service instance. Changing event logging rules may require an administrative change to the service contract or may be allowed under terms of the contract, offering, and instance. Logged events are included in reports that are requested in the Contract Reporting use case, or reports on the log may be scheduled automatically. They may also play a role in service event notification.	
Core Properties		
Name	**Content**	**Description**
eventId	ID (String)	A unique identifier of an event in the namespace of the cloud service provider
eventName	String (I18N))	A human-friendly name of the event (such as Network down, Access Denied, or Network Restored)
eventDateTime	Array of DateTime()	The date and time of the event
eventInformation	Array of String(I18N)	The event log information
supportedOperation	ID (Reference) [*]	The IDs of all the supportedOperation objects available for this object

12.20 Service Notification

Artifact	**Service Notification**
Description	This artifact represents the data content of a notification from the cloud service provider to the cloud service consumer when a predetermined or fault condition requires that the consumer be informed of a condition or event occurring on the service.

Role in use cases	Service notifications depend on a number of steps that precede actual notification. Service contracts may specify when notifications are to occur. They may be included in the service template and service offering that appears in the service catalog. The triggers for notification are part of the service instance, which is created when the resources are provisioned. The notification is created in the Notification of Service Condition or Event use case.	
Core Properties		
Name	**Content**	**Description**
notificationId	ID (String)	A unique identifier of the notification in the namespace of the cloud service provider. This could be a URI/IRI.
notificationDateTime	DateTime	The date and time of the notification
notificationTarget	Array of ID(String)	The persons, agents, or other entities that are the intended recipients of the notification. This could be a consumerId, a contactId, or an ID of another kind of agent, such as an application to which the notification is directed.
notificationType	String	General indication of the type of notification. Some example values: • Billing notice • Performance Impairment • Threshold Warning • Service Failure • Resource Allocation Change
notificationAck	Boolean (Yes/No)	Was the notification acknowledged?
notificationAckId	Array of ID (String)	IDs of the entities acknowledging the notification; a subset of the notificationTarget
notificationAckDateTime	DateTime	The date and time of acknowledgement. Usually, just the date and time of the first acknowledgement is adequate.
notificationTrigger	ID(String)	The ID of the triggering context that caused this notification to be emitted
notificationCondition	String	Machine-processable data on the conditions that brought about the notification
notificationDescription	String	Human-readable description of the significance of the notification. Possible content includes a description of the triggering condition.
notificationImpact	String	Human-readable description of the condition or event on the impact to the service provided (for example, performance, security, capacity, user experience)
notificationEventLog	String	Machine-processable data of the event sequence
notificationResponse	String	Human-readable description of necessary actions (for example, remedy, access control, recovery) taken by the cloud service provider in response to the event

Referenced Objects		
Name	**Content**	**Description**
notificationContextId	ID(Reference)	The ID of the artifact (for example, Service Instance or Relationship) to which this notification refers

12.21 Audit Event Record

Artifact	Audit Event Record	
Description	This artifact describes an auditable action that involves one or more resources or service instances in a cloud provider's IT infrastructure. Data content of a notification from the cloud service provider to the cloud service consumer when a predetermined or fault condition requires that the consumer be informed of a condition or event occurring on the service.	
Role in use cases	*As described in section 13.3, the security-related use cases that would use this artifact will be completed in future work.*	
Core Properties		
Name	**Content**	**Description**
eventId	ID (String)	A unique identifier of the event in the namespace of the cloud service provider. This could be a URI/IRI.
eventType	String	Indicates the type of auditing event. Note that this could be a type from a formal taxonomy or classification system.
authorizationDetails	ID (String) [*]	Permission or authorization information, including information on delegated identities
systemState	String	Platform for state information at the time the event occurred
eventDateTime	DateTime	The date and time of the notification
Referenced Objects		
Name	**Content**	**Description**
userId	ID (Reference)	The ID of the user responsible for the event
contextID	ID (Reference) [*]	The IDs of the components that participated in the actual event

13 Future Work

This section describes work for which a need is recognized but which has not yet been done.

13.1 Interface Granularity

The incubator's interface is "narrower" than available cloud interfaces. For example, what will the consumer do if it needs both? Use them side by side? Have a DMTF interface that is extendable so more detailed interfaces can be exposed by an implementation that is under the umbrella of the DMTF interface?

13.2 Interface Extendibility

We may need ways to add new types of entities to catalogs and to compose existing entities in a catalog into another entity (for example, put a firewall in front of an existing service). We need to determine how to represent the composed service without changing the original service entry.

13.3 Additional Use Cases

Detailed descriptions have not been created for several use cases that have been identified as useful. These are the use cases listed in Figure 4 that do not have an asterisk next to the use case name.

13.4 Security and Governance Use Cases

Several use cases that are specific to security management or governance have been identified and mapped to the service lifecycle states (see Figure 22). There has not been time to explore them further and to select those worth developing to the level of other use cases in this document. In particular, handling auditing events is an area that has been identified. Adding such use cases may result in adding additional actors that are specific to the security domain.

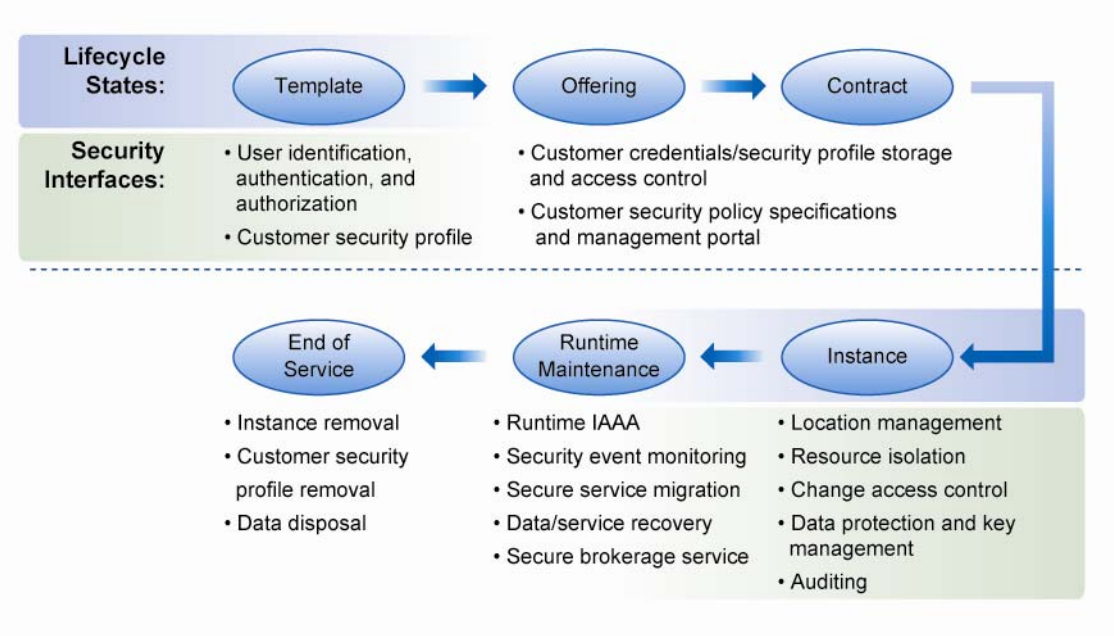


Figure 22 – Security-Specific Use Cases

13.5 Log Handling

Unresolved questions related to the Service Event Log artifact are as follows:

- Is it really a log, or is it an event?
- Is there a need for a log, and if so, what does it contain, and what are the operations on the log?
- Is there a need for a service record artifact that is distinct from a service notification?
- Which use cases would use these artifacts?

13.6 Cloud Service Developer Actor

The Cloud Service Developer actor may need extra clarifications and new interactions or use cases. As it is stated in the role definitions (see section 7), the cloud service developer can be an actor within an organization that is also a cloud service provider, a cloud service consumer, or a third-party. This last possibility may result in defining new interactions between the service provider and the service developer (for example, the developer receives income based on the use of his service templates, or the developer imposes rules and policies that constrain the use of the service templates). Such interactions and supporting data artifacts (for example, a contract between the cloud service provider and the developer to reflect their business relationship) may be explored.

Annex A
(informative)

Change Log

Version	Date	Description
1.0.0	2010-06-18	Released as DMTF Informational