International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.3528

(02/2022)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

Cloud Computing

Cloud computing – Framework and requirements of container management in inter-cloud

Recommendation ITU-T Y.3528

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Computing power networks	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
FUTURE NETWORKS	Y.3000–Y.3499
CLOUD COMPUTING	**Y.3500–Y.3599**
BIG DATA	Y.3600–Y.3799
QUANTUM KEY DISTRIBUTION NETWORKS	Y.3800–Y.3999
INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES	
General	Y.4000–Y.4049
Definitions and terminologies	Y.4050–Y.4099
Requirements and use cases	Y.4100–Y.4249
Infrastructure, connectivity and networks	Y.4250–Y.4399
Frameworks, architectures and protocols	Y.4400–Y.4549
Services, applications, computation and data processing	Y.4550–Y.4699
Management, control and performance	Y.4700–Y.4799
Identification and security	Y.4800–Y.4899
Evaluation and assessment	Y.4900–Y.4999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T Y.3528

Cloud computing – Framework and requirements of container management in inter-cloud

Summary

Recommendation ITU-T Y.3528 provides the framework and functional requirements of container management in the inter-cloud. It provides an overview, framework, functional requirements and use cases of container management in the inter-cloud. The functional requirements are derived from the corresponding typical use cases.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.3528	2022-02-13	13	11.1002/1000/14858

Keywords

Container, framework, functional requirements, inter-cloud, management.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T Y.3528

Cloud computing – Framework and requirements of container management in inter-cloud

1 Scope

This Recommendation provides framework and requirements of container management in the inter-cloud. It addresses the following:

− overview of container management in the inter-cloud;

− framework of container management in the inter-cloud;

− functional requirements of container management in the inter-cloud;

− use cases of container management in the inter-cloud.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T X.1601] Recommendation ITU-T X.1601 (2015), *Security framework for cloud computing*.

[ITU-T Y.3502] Recommendation ITU-T Y.3502 (2014) | ISO/IEC 17789:2014, *Cloud computing – Reference architecture*.

[ITU-T Y.3511] Recommendation ITU-T Y.3511 (2014), *Framework of inter-cloud computing*.

[ITU-T Y.3516] Recommendation ITU-T Y.3516 (2017), *Cloud computing – Functional architecture of inter-cloud computing*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 cloud computing [b-ITU-T Y.3500]: Paradigm for enabling network access to a scalable and elastic pool of shareable physical or virtual resources with self-service provisioning and administration on-demand.

NOTE – Examples of resources include servers, operating systems, networks, software, applications, and storage equipment.

3.1.2 cloud service [b-ITU-T Y.3500]: One or more capabilities offered via cloud computing invoked using a defined interface.

3.1.3 cloud service customer [b-ITU-T Y.3500]: Party which is in a business relationship for the purpose of using cloud services.

NOTE – A business relationship does not necessarily imply financial agreements.

3.1.4 cloud service provider [b-ITU-T Y.3500]: Party which makes cloud services available.

3.1.5 container [b-ITU-T Y.3535]: A set of software to provide isolation, resource control and portability for virtualization processing of the application.

NOTE 1 – Container runs on the kernel in a bare-metal machine or virtual machine.

NOTE 2 – "Application" implies business logic including a required library or binary to run in a container.

3.1.6 hybrid cloud [b-ITU-T Y.3500]: Cloud deployment model using at least two different cloud deployment models.

3.1.7 inter-cloud computing [ITU-T Y.3511]: The paradigm for enabling the interworking between two or more cloud service providers.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API Application Programming Interface

CSC Cloud Service Customer

CSP Cloud Service Provider

DNS Domain Name System

IaaS Infrastructure as a Service

SaaS Software as a Service

5 Conventions

In this Recommendation:

The keywords "**is required to**" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.

The keywords "**is recommended**" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

In the body of this Recommendation and its annexes, the words should and may sometimes appear, in which case they are to be interpreted, respectively, as is recommended and can optionally. The appearance of such phrases or keywords in an appendix or in material explicitly marked as informative is to be interpreted as having no normative intent.

6 Overview of container management in inter-cloud

6.1 Advantages of deploying containers in inter-cloud

As an attractive lightweight cloud technology, containers are widely used by cloud service providers (CSPs), due to the fact that they allow cloud service customers (CSCs) to deploy applications in any environment faster and more efficiently than using virtual machines or physical machines. However, currently most containers are deployed and managed in a single cloud level, which limits their capabilities. For example, deployment in this way, when on-premises resources of the cloud/data centre are busy or unavailable, the CSP cannot accommodate this situation with scaling out. And if the cloud/data centre is out of service through an outage or other disaster, the CSP cannot provide satisfied resilience for the services deployed on these containers. With the popularity of the inter-cloud and hybrid cloud, all of these issues can be solved by deploying containers across different

clouds. Accordingly, the ability to effectively deploy and manage containers in an inter-cloud environment becomes more and more important.

Adopting container deployment in the inter-cloud brings advantages as below:

– **Better availability and resilience**: Deploying containers in the inter-cloud means even when containers in one cloud are out of service for some reason, the service deployed in these containers still can be provided by other containers deployed in other clouds.

– **Low latency**: Deploying containers in multiple regions can minimize latency by serving CSCs through the containers that are closest to them.

– **Better scalability**: There are scalability limits to a single container cluster and large-scale cluster failures can lead to large losses of capacity in a single cloud. Building a container cluster in the inter-cloud can achieve better scalability.

6.2 Challenges for container management in inter-cloud

Although the use of inter-cloud for deployment of containers brings many advantages, it also brings many challenges that have not been previously explored in a single cloud environment. This includes:

– **Resource selection**: How to find the best or more suitable resource to deploy new containers in an inter-cloud environment is challenging because the selection process is based on various criteria, for example, resource availability and utilization, workload, cost, performance, etc. And the fluctuations in workload and resource availability and utilization make this selection process more complex.

– **Resource monitoring**: Once a set of clouds has been selected their resource should be monitored in a uniform manner, even if they have a different cloud management platform.

– **Container scheduling**: How to select the appropriate affinity when placing a container is challenging because the scheduling policies are complex, and the performance and resilience of the service deployed in containers should be balanced. To deploy containers closer or further away could get better performance or better resilience, respectively.

– **Container migration**: Many reasons, such as maintenance or lack of hardware resources, could lead containers migration. When migrating containers in an inter-cloud environment, there are some constraints to be considered. For example, some containers should be reserved for required resources. Before the migration the target cloud should be validated for the required resources.

For the reasons of the above challenges, container management in the inter-cloud is different from container management in a single cloud.

7 Framework of container management in inter-cloud

7.1 Container management functional components in inter-cloud

The integrated container management in the inter-cloud relies on a two-level management framework which is presented in Figure 7-1:

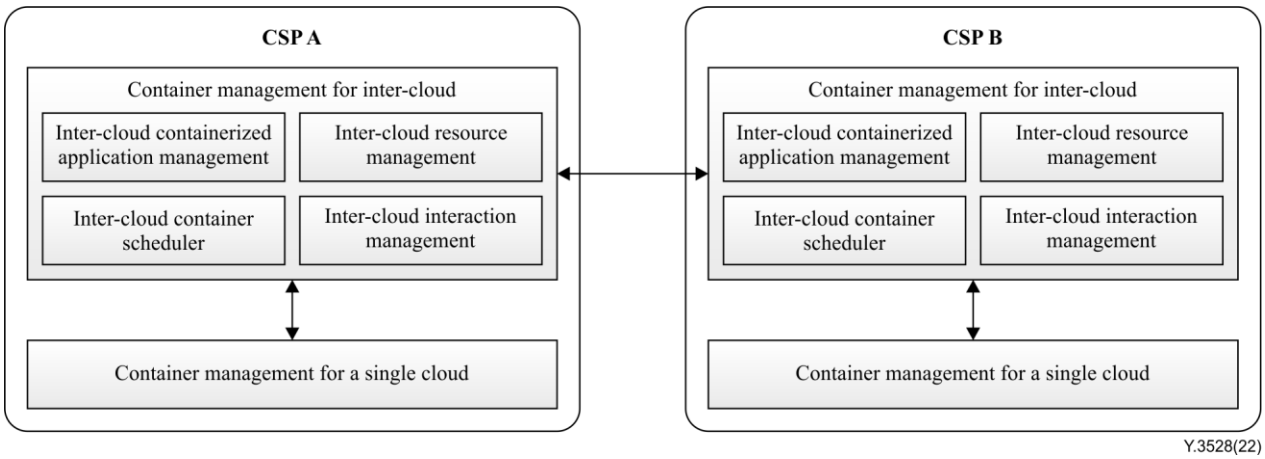


Figure 7-1 – The two-level management framework of container management in the inter-cloud

The container management for an inter-cloud functional component is responsible for container management across different CSPs. The container management for a single cloud functional component is responsible for the container management in each cloud, including selecting an appropriate cluster and node to deploy containers. The container management for an inter-cloud functional component is built upon the following functional components:

– **Inter-cloud containerized application management**: This is responsible for accepting the request for deploying containerized applications in the inter-cloud. The required resources (CPU, memory and storage, etc.) for each container are specified in the request. The container deployment constraints are also specified in the request, which could include the total number of containers and the container scheduling preference for this application. The container scheduling preference specifies how to distribute containers in the inter-cloud, which could be even distribution in all CSPs or weighted distribution for different CSPs, etc. This functional component is also responsible for accepting the request for updating or removing the deployed containerized application in the inter-cloud, and manages all of the containerized application deployment results. When a CSC requests to access the application, this functional component acts as a DNS server and a load balancer to provide the IP addresses of the application load-balanced in different CSPs to the CSC.

– **Inter-cloud resource management**: This is responsible for managing the entering and leaving of the CSP in the inter-cloud. It is also responsible for monitoring the status of resources in the inter-cloud. The status changes of the resources may result in a reschedule for containers. For example, if the resource utilization of a CSP breaks the threshold, the new request for resources will not be assigned to this CSP and some containers in this CSP will be migrated to other CSPs.

– **Inter-cloud container scheduler**: This is responsible for assigning the containers to appropriate CSPs based on container deployment constraints and the dynamic monitoring results for the resources in the inter-cloud. The further scheduling to select a specific node in the cluster to deploy the container is a responsibility of container management for a single cloud functional component.

– **Inter-cloud interaction management**: This is responsible for managing the interactions with other CSPs in the inter-cloud and using the capabilities of container management for a single cloud to realize the container deployment tasks. The interchanged information between CSPs includes container deployment tasks and results, and resources' status of different CSPs in the inter-cloud.

The relationships between particular functional components of container management in the inter-cloud are presented in Figure 7-2.

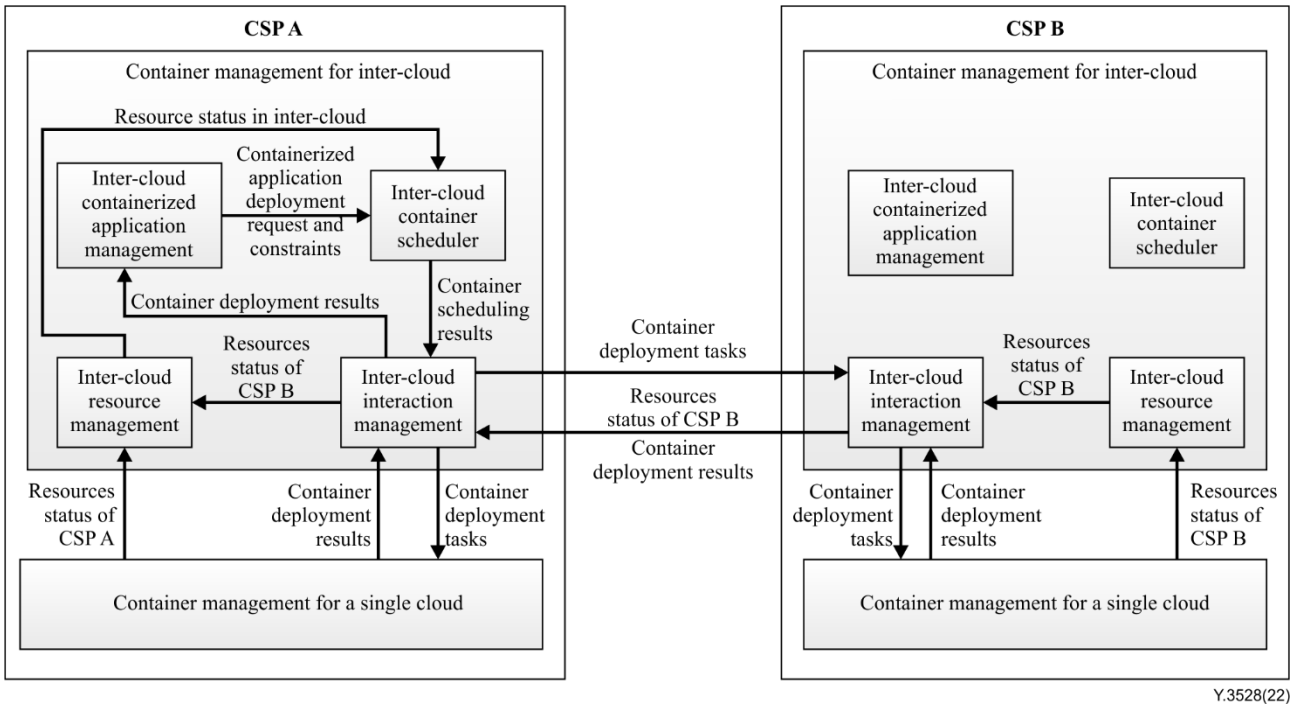


Figure 7-2 – The relationships between particular functional components of container management in the inter-cloud

In Figure 7-2, CSP A deploys a containerized application in CSP A and CSP B. When a new containerized application is requested to be deployed in the inter-cloud, the following steps are executed:

1) The inter-cloud containerized application management functional component of CSP A sends the required resources for each container and the container deployment constraints including the container scheduling preference to the inter-cloud container scheduler functional component of CSP A.

2) The inter-cloud resource management functional component of CSP A provides the resources status in the inter-cloud to inter-cloud container scheduler functional component of CSP A. The resources status in the inter-cloud includes the resources status from different CSPs' container management for a single cloud functional component.

3) The inter-cloud container scheduler functional component of CSP A schedules the containers to appropriate CSPs based on the container deployment constraints and the dynamic monitoring results for resources in the inter-cloud.

4) The inter-cloud container scheduler functional component of CSP A sends the scheduling results to the inter-cloud interaction management functional component of CSP A. If the containers should be deployed in CSP B, the inter-cloud interaction management functional component of CSP A sends the container deployment tasks to the inter-cloud interaction management functional component of CSP B. Otherwise, it sends the container deployment tasks to the container management for a single cloud functional component of CSP A to implement the actual deployment for each container.

5) When the container deployment work is done, the container management for a single cloud functional component sends the container deployment results information to the inter-cloud interaction management functional component of the same CSP. If the container deployment work is done in CSP B, the inter-cloud interaction management functional component of

CSP B sends the container deployment results information to the inter-cloud interaction management functional component of CSP A. The inter-cloud interaction management functional component of CSP A sends this information to the inter-cloud containerized application management functional component of CSP A for service provision and load balance.

7.2 The position of container management functional components in inter-cloud

The cloud computing reference architecture [ITU-T Y.3502] provides the functional architecture and functional components of cloud computing. It also defines the functional components for supporting inter-cloud computing, e.g., peer service integration and peer service management. The inter-cloud computing functional architecture [ITU-T Y.3516] is built on the functional view of the cloud computing reference architecture and makes extensions to functional components with inter-cloud functions (see clause 8 of [ITU-T Y.3516]). But neither of them specify the functional components for container management in the inter-cloud.

This Recommendation is based on the functions defined in [ITU-T Y.3502] and [ITU-T Y.3516], and identifies container management in the inter-cloud specific extensions to functional components. It focuses on the container management framework and requirements in an inter-cloud environment based on the patterns between multiple peer CSPs, including peering, federation and intermediary which are defined in [ITU-T Y.3511].

Container management in an inter-cloud environment is considered in cross-CSP and cross-layer dimensions. The container management functionalities are supported by the "peer service management" functional component and "platform and virtualization management" functional component within the multilayer functions of the cloud computing reference architecture defined in [ITU-T Y.3502]. The position of a container management functional component across the CSPs which provide inter-cloud services is presented in Figure 7-3.

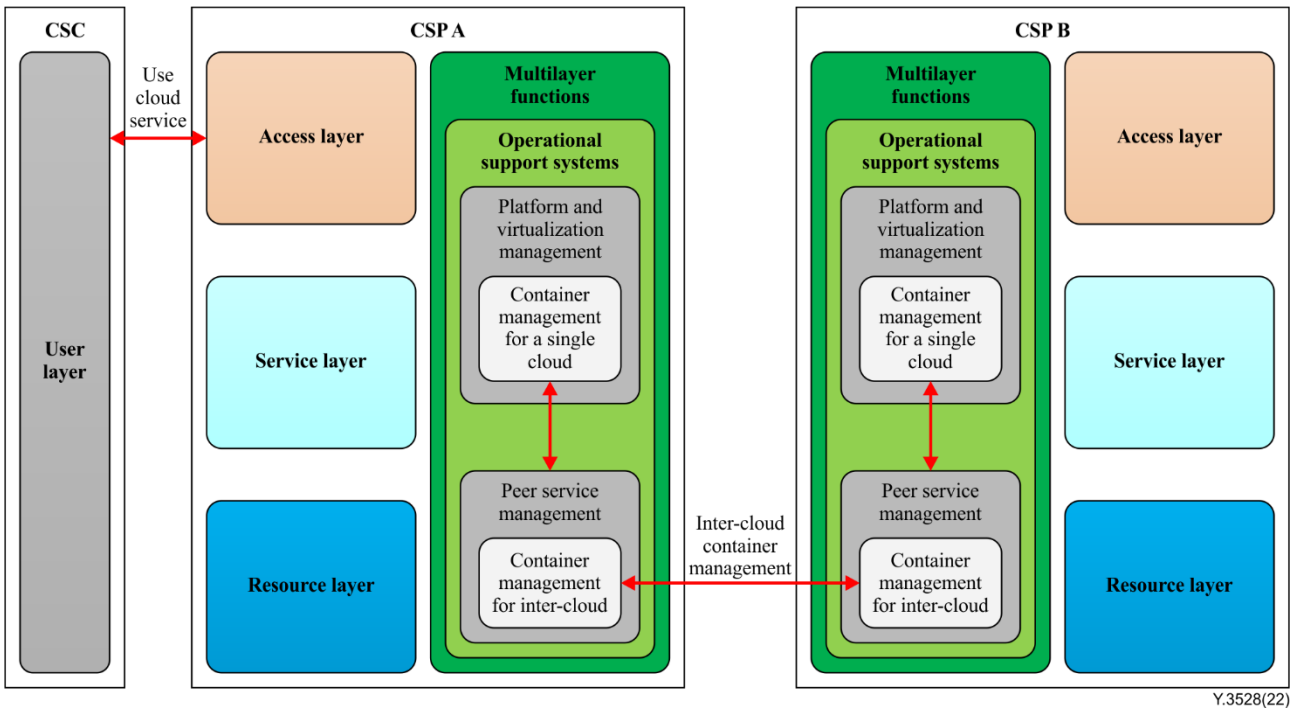


Figure 7-3 – The position of container management functional components over CSP in the inter-cloud

The inter-cloud container management is realised over container management for inter-cloud functional components of CSP A and CSP B located in the "peer service management" functional components.

7.3 Container management in different inter-cloud patterns

There are three patterns of inter-cloud for describing relations and interactions involving multiple CSPs, i.e., inter-cloud peering, inter-cloud federation and inter-cloud intermediary [ITU-T Y.3511]. The description of container management for each pattern is illustrated as follows.

7.3.1 Container management in inter-cloud peering

In inter-cloud peering, two CSPs interwork directly with each other in order to use the services provided by the peer CSP. As shown in Figure 7-4, CSP A is the primary CSP when using the resource of CSP B to deploy containers for providing service X and service Y to its own CSCs. Both CSP A and CSP B should support container management for inter-cloud and container management for single cloud functional components. The APIs between CSP A and CSP B interchange the container deployment tasks and results for CSP B and resources status of CSP B.

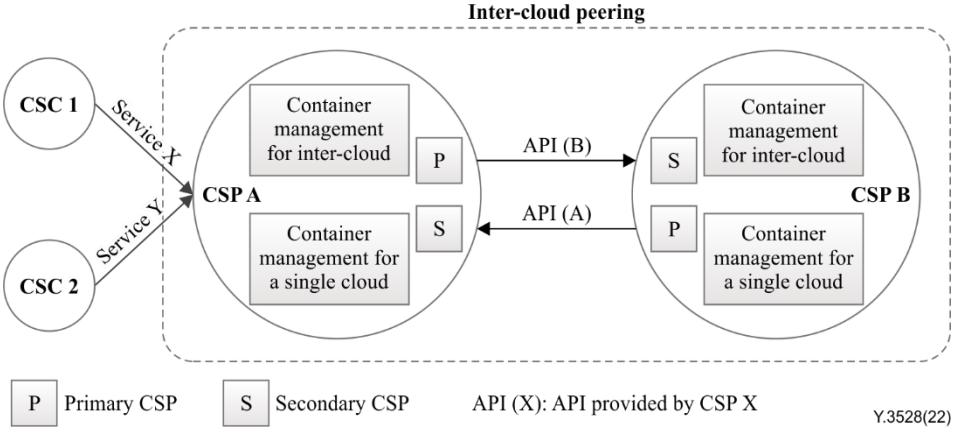


Figure 7-4 – Container management in inter-cloud peering

7.3.2 Container management in inter-cloud federation

In inter-cloud federation, a number of CSPs provide services to CSCs. When needed (e.g., in the event of a serious shortage in the resource pool), the CSP within the inter-cloud federation utilizes other CSPs' resources to provide services to their CSCs. As shown in Figure 7-5, CSC 1 and CSC 2 use services X and Y provided by CSP A, but the containers supporting services X and Y may actually be deployed in CSP B, CSP C, CSP D and CSP E. CSC 3 uses services Z provided by CSP E but the containers supporting services Z may actually be deployed in CSP A, CSP B, CSP C and CSP D. All of the CSPs in the inter-cloud should support both container management for inter-cloud and container management for single cloud functional components. A common API for cloud interworking is defined in the inter-cloud federation, each CSP interworks with the other CSPs in the inter-cloud federation through this common API to interchange the container deployment tasks and results and resources status of each CSP.

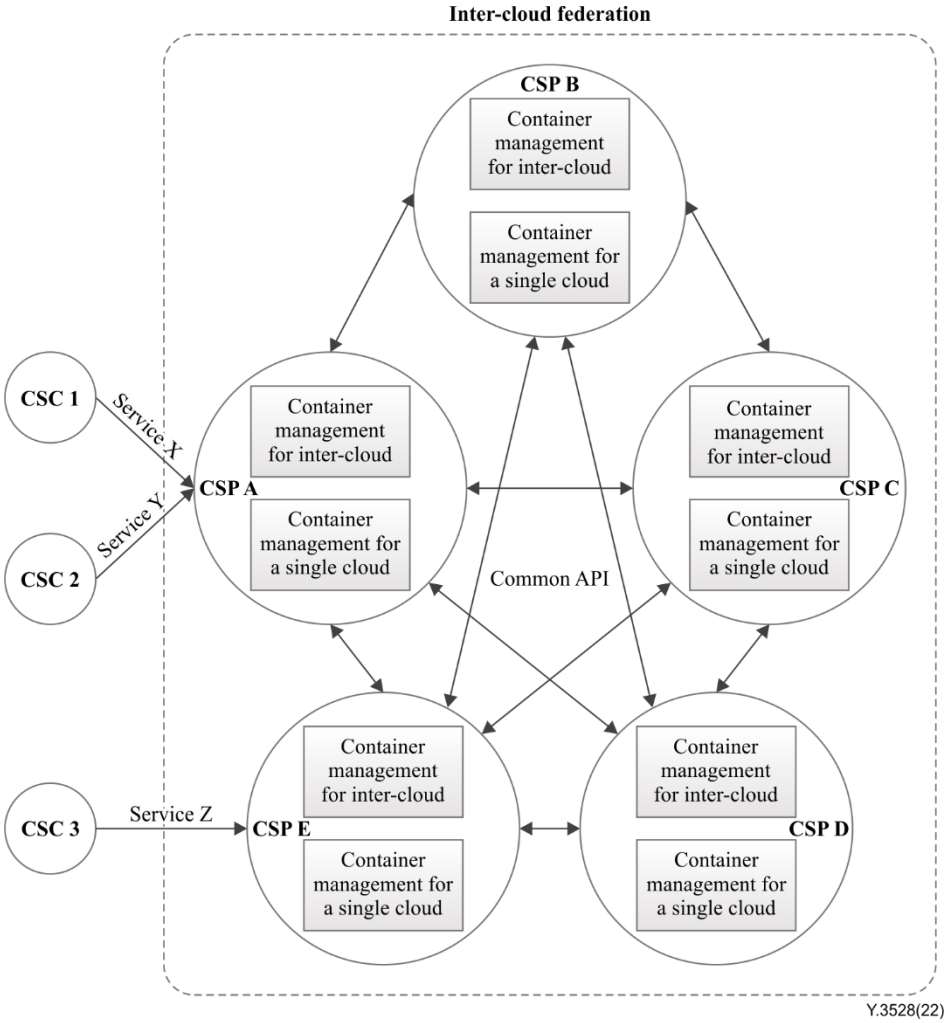


Figure 7-5 – Container management in inter-cloud federation

7.3.3 Container management in inter-cloud intermediary

In the inter-cloud intermediary pattern, the CSP interworks with one or more peer CSPs and provides intermediation, aggregation and arbitrage of services provided by these CSPs. As shown in Figure 7-6, CSP A is a primary CSP when using the resource of CSP B, CSP C, CSP D and CSP E to deploy containers for providing services to its own CSCs. All of the CSPs in the inter-cloud should support both container management for the inter-cloud and container management for single cloud functional components. The APIs between CSP A and other CSPs interchange the container deployment tasks and results and resources status of other CSPs.

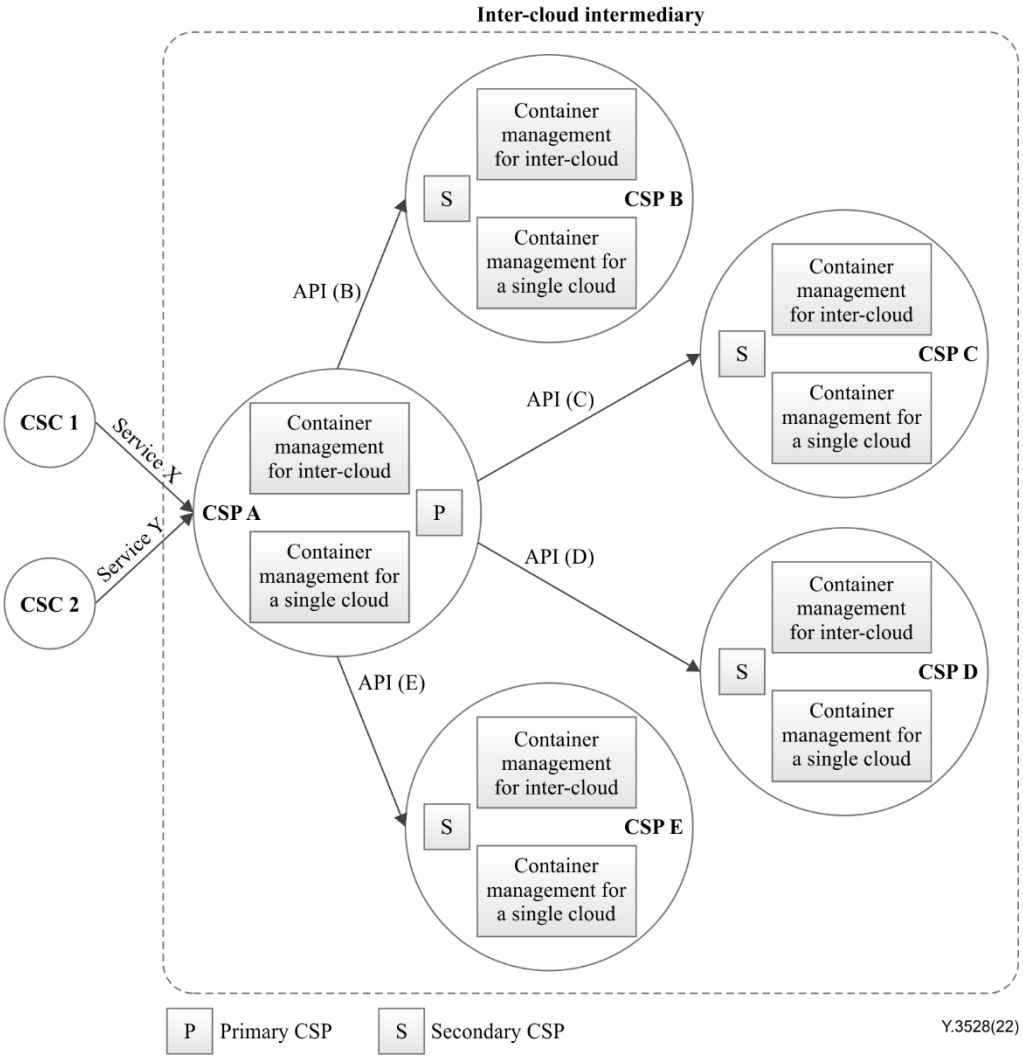


Figure 7-6 – Container management in inter-cloud intermediary

8 Functional requirements of container management in inter-cloud

8.1 Inter-cloud interaction management

It is required that a CSP supports interchanging the container deployment tasks and results and resources status with other CSPs in the inter-cloud.

8.2 Inter-cloud resource monitoring

It is recommended that the primary CSP supports the continuous monitoring of the status of resources of the secondary CSP(s) in the inter-cloud.

8.3 Inter-cloud container scheduling

It is required that the primary CSP supports distributing containers in the inter-cloud based on container deployment constraints.

NOTE – The container deployment constraints include the total number of containers and the container scheduling preference for the application carried by the containers in the inter-cloud. The container scheduling preference specifies how to distribute the containers in the inter-cloud, which could be evenly distributed in all CSPs or weighted distributed for different CSPs, etc.

It is required that the primary CSP supports rescheduling containers for handling fluctuant workloads in the inter-cloud based on container deployment constraints.

8.4 Container migration

It is recommended that the primary CSP supports dynamical container instance adjustment of the application across different CSPs in the inter-cloud, based on predefined conditions and rules.

It is recommended that the CSP supports containerized application data synchronization and replication across different CSPs in the inter-cloud.

8.5 CSP changing management in inter-cloud

It is required that the primary CSP supports reconfiguring the container deployment constraints when there are CSPs joining or quitting the inter-cloud.

It is required that the primary CSP supports rescheduling the containers in the inter-cloud based on the reconfigured container deployment constraints when there are CSPs joining or quitting the inter-cloud.

8.6 Inter-cloud container deployment constraints management

It is required that the primary CSP supports container deployment constraints management, including adding, deleting and updating the container deployment constraints, according to application request and the status of the resources in the inter-cloud.

9 Security considerations

Security aspects for consideration within the cloud computing environment are described in [ITU-T X.1601] which analyses security threats and challenges, and describes security capabilities that could mitigate these threats and meet the security challenges.

Appendix I

Use case of container management in inter-cloud

(This appendix does not form an integral part of this Recommendation.)

I.1 Use case template

The use cases developed in Appendix I should adopt the following unified format for better readability and convenient material organization.

Table I.1 – Use case template

Title	The title of the use case
Description	Scenario description of the use case
Roles	Roles involved in the use case
Figure (optional)	Figure to explain the use case, but not mandatory
Preconditions (optional)	The necessary preconditions that should be achieved before starting the use case.
Post-conditions (optional)	The post-condition that will be carried out after the termination of the current use case.
Derived requirements	Requirements derived from the use cases, whose detailed description is presented in the dedicated chapter

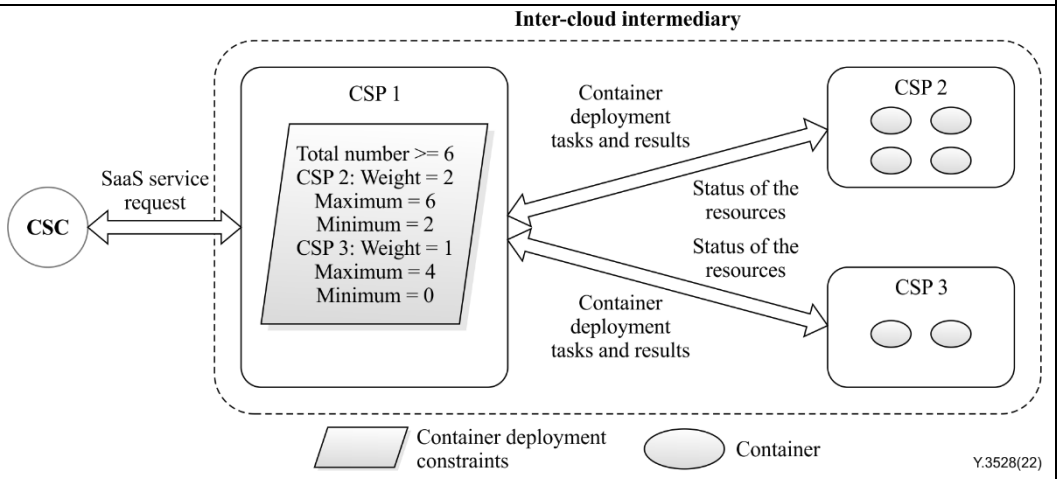
I.2 Use case of deploying a containerized application in inter-cloud

This use case illustrates the process of deploying a containerized application in the inter-cloud.

Table I.2 – Deploying a containerized application in the inter-cloud

Title	Deploying a containerized application in the inter-cloud
Description	– CSP1 is the primary CSP in an inter-cloud intermediary pattern. It provides a software as a service (SaaS) implemented through a containerized application to the CSC, and the containers for the containerized application could be placed in CSP2 and CSP3 within the inter-cloud. – To guarantee the quality and availability of the SaaS, CSP1 needs to control the total number and the distribution of the containers for this application in the inter-cloud automatically through setting container deployment constraints. – The container deployment constraints include the total number of containers and the container scheduling preference for this application. The container scheduling preference could be configured flexibly to accommodate different requests. Its distribution could be even distribution in all CSPs or weighted distribution in different CSPs, etc. It could also limit the maximum and minimum of the containers for this application in each CSP. For example, considering the location and the amount of resources in CSP2 and CSP3, CSP1 decides to maintain no less than six containers for deploying this application and distribute these containers with weighted distribution that CSP2's weight is 2 and CSP3's weight is 1. The maximum and minimum of the containers for this application in CSP2 are six and two, the maximum and minimum of the containers for this application in CSP3 are four and zero. – Based on this container scheduling preference, when this application is deployed, initially there should be four containers in CSP2 and two containers in CSP3. With

Table I.2 – Deploying a containerized application in the inter-cloud

	the increasing of requests for this application, new containers will be deployed in CSP2 and CSP3 with this ratio. – The resource utilization and the status of the resources of CSP2 and CSP3 are continuously monitored by CSP1. CSP1 needs to know how many containers for this application are running in each CSP and their health condition. – If there are some changes for the resources, it may result in a reschedule for containers in the inter-cloud. For example, if the resource utilization of CSP3 breaks the threshold, some of the containers in CSP3 will migrate to CSP2.
Roles	CSC, CSP
Figure (optional)	
Preconditions (optional)	– CSP1(SaaS) deploys its containerized application on CSP2(IaaS) and CSP3(IaaS).
Post-conditions (optional)	– CSP1(SaaS) deployed its containerized application on CSP2(IaaS) and CSP3(IaaS) based on the container scheduling preference for this application.
Derived requirements	– Inter-cloud interaction management (refer to clause 8.1) – Inter-cloud resource monitoring (refer to clause 8.2) – Inter-cloud container scheduling (refer to clause 8.3)

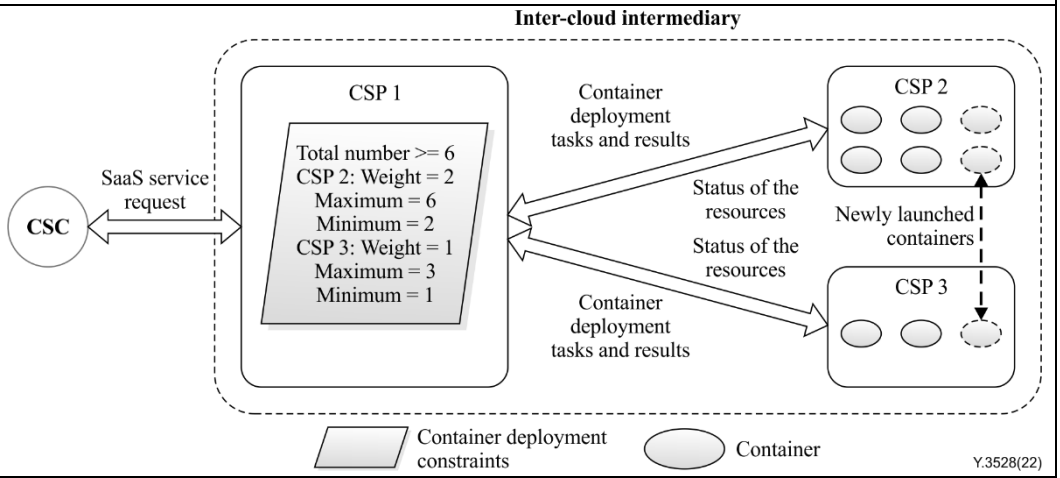
I.3 Use case of rescheduling containers for handling fluctuant workloads in inter-cloud

This use case illustrates the process of rescheduling containers for fluctuant workloads in the inter-cloud.

Table I.3 – Rescheduling containers for handling fluctuant workloads in the inter-cloud

Title	Rescheduling containers for handling fluctuant workloads in the inter-cloud
Description	– CSP1 is the primary CSP in an inter-cloud intermediary pattern. It provides a SaaS implemented through a containerized application to the CSC, and the containers for the containerized application could be placed in CSP2 and CSP3 in the inter-cloud. – Based on the container deployment constraints, when this application is deployed, initially there are four containers in CSP2 and two containers in CSP3. – With the increasing requests for this SaaS, the service response time is getting longer and longer. To guarantee the quality of the service, CSP1 should launch more containers to share the workload of this service.

Table I.3 – Rescheduling containers for handling fluctuant workloads in the inter-cloud

	– Based on the container deployment constraints, two containers are newly launched in CSP2 and one container is newly launched in CSP3 to improve the performance of the SaaS.
Roles	CSC, CSP
Figure (optional)	
Preconditions (optional)	– CSP1(SaaS) deploys its containerized application in CSP2(IaaS) and CSP3(IaaS).
Post-conditions (optional)	– 3 containers are newly launched in CSP2(IaaS) and CSP3(IaaS) based on the container deployment constraints to share the workload of this service.
Derived requirements	– Inter-cloud container scheduling (refer to clause 8.3)

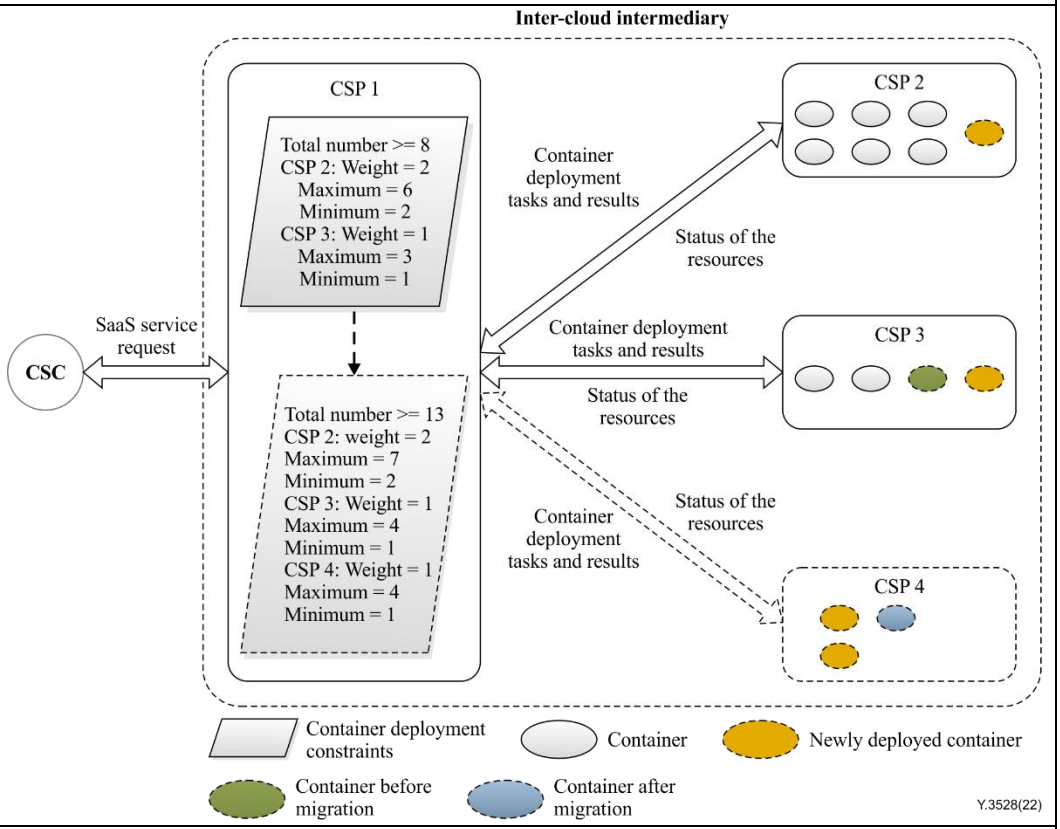
I.4 Use case of managing CSPs changing in inter-cloud

This use case illustrates the process of adding a CSP into an existing inter-cloud relationship.

Table I.4 – Managing CSPs changing in the inter-cloud

Title	Managing CSPs changing in the inter-cloud
Description	– CSP1 is the primary CSP in an inter-cloud intermediary pattern. It provides a SaaS implemented through a containerized application to the CSC, and the containers for the containerized application could be placed in CSP2 and CSP3 in the inter-cloud. – CSP2 has launched six containers and CSP3 has launched three containers for this application, which has reached the upper limit according to the container deployment constraints. – With the increasing requests for this SaaS, CSP1 has to deploy more containers in the existing CSP or new CSP to guarantee the quality of the service. – CSP1 tries to deploy more containers in CSP2 and CSP3. CSP1 should have the capability to manage the container deployment constraints of CSP2 and CSP3. The deployment constraints could be reconfigured by either a SaaS service request or monitoring results of the resources status of CSP2 and CSP3. – According to the latest deployment constraints reconfigured by the SaaS service request, the upper limit of CSP2 and CSP3 have been adjusted to seven and four respectively. – Based on the updated container deployment constraints, a new container has been scheduled to CSP2 and CSP3, respectively.

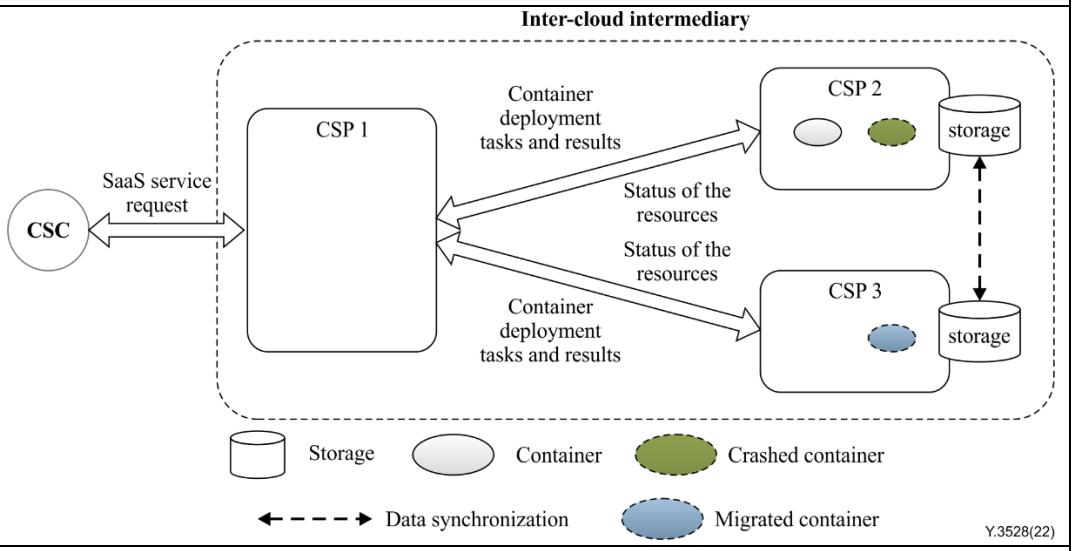
Table I.4 – Managing CSPs changing in the inter-cloud

	– In addition, CSP1 requests more resource to deploy a SaaS. A new CSP (here CSP4) is added into the inter-cloud relationship. The containerized application is also placed in CSP4. – To take full advantage of the resources, CSP1 should have the capability to monitor the status of the resources of CSP4 and reconfigure the container deployment constraints for the application to include the container scheduling preference for CSP4. – Based on the reconfigured container deployment constraints, initially there are three containers launched in CSP4. Two of the three containers are newly deployed to guarantee the quality of the SaaS service, and the other container migrates from CSP3 considering the equal weight of CSP3 and CSP4.
Roles	CSC, CSP
Figure (optional)	
Preconditions (optional)	– CSP1(SaaS) deploys its containerized application in CSP2(IaaS) and CSP3(IaaS).
Post-conditions (optional)	– CSP1(SaaS) deploys its containerized application in CSP2(IaaS), CSP3(IaaS) and CSP4(IaaS) based on the reconfigured container scheduling preference.
Derived requirements	– CSP changing management in the inter-cloud (refer to clause 8.5) – Inter-cloud container deployment constraints management (refer to clause 8.6)

I.5 Use case of container migration in inter-cloud

This use case illustrates container migration in the inter-cloud.

Table I.5 – Container migration in the inter-cloud

Title	Container migration in the inter-cloud
Description	CSP1 is the primary CSP in an inter-cloud intermediary pattern. It provides a SaaS implemented through a containerized application to the CSC, and the containers for the containerized application could be placed in CSP2 and CSP3 in the inter-cloud. When CSP2 crashes half of the cluster nodes for containers, the workload of all left nodes exceeds its predefined threshold. In this case, the crashed container has to be migrated to CSP3 and half of the workload of CSP2 is migrated to CSP3. At the same time, the data of the service also has to be migrated or replicated to CSP3. CSP1 synchronizes the data of the service with CSP2 and CSP3. The remaining data in CSP2 should keep real-time synchronization with CSP3.
Roles	CSP, CSC
Figure (optional)	
Preconditions (optional)	CSP1(SaaS) deploys its containerized application on CSP2(IaaS) and CSP3(IaaS).
Post-conditions (optional)	The crashed container has been migrated from CSP2 to CSP3.
Derived requirements	– Container migration (refer to clause 8.4)

Bibliography

[b-ITU-T Y.3500] Recommendation ITU-T Y.3500 (2014) | ISO/IEC 17788:2014, *Cloud computing – Overview and Vocabulary.*

[b-ITU-T Y.3535] Recommendation ITU-T Y.3535 (2022), *Cloud computing – Functional requirements for container*

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	**Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2022