International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.1602

(03/2016)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

Cloud computing security – Cloud computing security design

Security requirements for software as a service application environments

Recommendation ITU-T X.1602

International Telecommunication Union

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	X.1–X.199
OPEN SYSTEMS INTERCONNECTION	X.200–X.299
INTERWORKING BETWEEN NETWORKS	X.300–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600–X.699
OSI MANAGEMENT	X.700–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	X.850–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	
General security aspects	X.1000–X.1029
Network security	X.1030–X.1049
Security management	X.1050–X.1069
Telebiometrics	X.1080–X.1099
SECURE APPLICATIONS AND SERVICES	
Multicast security	X.1100–X.1109
Home network security	X.1110–X.1119
Mobile security	X.1120–X.1139
Web security	X.1140–X.1149
Security protocols	X.1150–X.1159
Peer-to-peer security	X.1160–X.1169
Networked ID security	X.1170–X.1179
IPTV security	X.1180–X.1199
CYBERSPACE SECURITY	
Cybersecurity	X.1200–X.1229
Countering spam	X.1230–X.1249
Identity management	X.1250–X.1279
SECURE APPLICATIONS AND SERVICES	
Emergency communications	X.1300–X.1309
Ubiquitous sensor network security	X.1310–X.1339
PKI related Recommendations	X.1340–X.1349
CYBERSECURITY INFORMATION EXCHANGE	
Overview of cybersecurity	X.1500–X.1519
Vulnerability/state exchange	X.1520–X.1539
Event/incident/heuristics exchange	X.1540–X.1549
Exchange of policies	X.1550–X.1559
Heuristics and information request	X.1560–X.1569
Identification and discovery	X.1570–X.1579
Assured exchange	X.1580–X.1589
CLOUD COMPUTING SECURITY	
Overview of cloud computing security	X.1600–X.1601
Cloud computing security design	**X.1602–X.1639**
Cloud computing security best practices and guidelines	X.1640–X.1659
Cloud computing security implementation	X.1660–X.1679
Other cloud computing security	X.1680–X.1699

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

Recommendation ITU-T X.1602

Security requirements for software as a service application environments

Summary

Recommendation ITU-T X.1602 analyses the maturity levels of software as a service (SaaS) application and proposes security requirements to provide a consistent and secure service execution environment for SaaS applications. These proposed requirements originate from cloud service providers (CSP) and cloud service partners (CSN) as they need a SaaS application environment to meet their demands on security. The requirements are general and independent of any service or scenario specific model (e.g., web services, or representational state transfer (REST)), assumptions or solutions.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T X.1602	2016-03-23	17	11.1002/1000/12615

Keywords

Security requirement, software as a service (SaaS) application environment, SaaS maturity level.

* 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, 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 TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T X.1602

Security requirements for software as a service application environments

1 Scope

This Recommendation focuses mainly on the security requirements of software as a service (SaaS) application environments based on the SaaS application maturity level. The target audiences of this Recommendation are cloud service providers (CSPs) and cloud service partners (CSNs) such as application developers.

2 References

None.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

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

3.1.2 cloud service category [b-ITU-T Y.3500]: Group of cloud services that possess some common set of qualities.

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.

3.1.4 cloud service partner [b-ITU-T Y.3500]: Party which is engaged in support of, or auxiliary to, activities of either the cloud service provider or the cloud service customer, or both.

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

3.1.6 cloud service user [b-ITU-T Y.3500]: Natural person, or entity acting on their behalf, associated with a cloud service customer that uses cloud services.

3.1.7 desktop as a service [b-ITU-T Y.3500]: The capabilities provided to the cloud service customer are the ability to build, configure, manage, store, execute, and deliver users' desktop functions remotely.

3.1.8 infrastructure as a service (IaaS) [b-ITU-T Y.3500]: Cloud service category in which the cloud capabilities type provided to the cloud service customer is an infrastructure capabilities type.

3.1.9 software as a service (SaaS) [b-ITU-T Y.3500]: Cloud service category in which the cloud capabilities type provided to the cloud service customer is an application capabilities type.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ASP Application Service Provider

CaaS Communications as a Service

CRM Customer Relationship Management

CSC Cloud Service Customer

CSN Cloud Service Partner

CSP Cloud Service Provider

DaaS Desktop as a Service

IaaS Infrastructure as a Service

IAM Identity and Access Management

IdM Identity Management

OLAP OnLine Analytical Processing

OS Operating System

PaaS Platform as a Service

PKI Public Key Infrastructure

REST Representational State Transfer

SaaS Software as a Service

SAP Service Access Point

SLA Service Level Agreement

5 Conventions

None.

6 Overview

A software as a service (SaaS) application environment is a service-oriented multi-tenant development, deployment and execution environment in which software and its associated data are hosted centrally and are typically accessed on-demand by users using a client, e.g., a web browser, over the Internet.

While this Recommendation is primarily concerned with SaaS, some of the concepts in this Recommendation may also be applicable to other cloud service categories that also include the application capabilities type, for example communications as a service (CaaS).

Figure 1 depicts a conceptual model of a SaaS application environment. The underlying capabilities from infrastructure as a service (IaaS), platform as a service (PaaS) and desktop as a service (DaaS) will be encapsulated into services and provide consistent secure access using exported service access point (SAP). In this Recommendation, IaaS could provide computing services, storage services and network services; PaaS could provide platform service, and DaaS could provide desktop service for a SaaS application environment. All these services constitute the basic building blocks of an application development.

The environment also provides some necessary service management functions including service registration, service configuration, service orchestration, service dependency checking, service access control, service isolation, service monitoring and other service control functions.

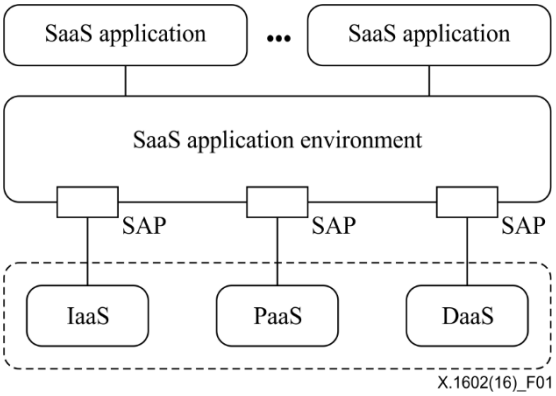
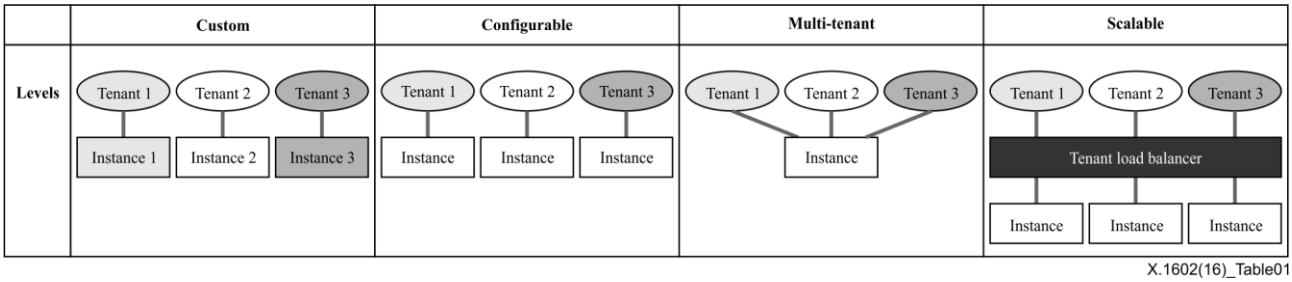


Figure 1 – Conceptual model for the SaaS application environment

7 Maturity levels of SaaS application

In the industry, the maturity of SaaS is classified into four levels which could be shortly named as custom level, configurable level, multi-tenant level, and scalable level. Each level covers characteristics of the previous one and provides extended characteristics. The diagram that represents the characteristics of the different SaaS maturity models is shown in Table 1.

Table 1 – Diagram of SaaS application maturity level



Different maturity levels of the SaaS application have different security requirements to SaaS application environments, and the requirements will be illustrated from the viewpoint of CSPs and CSNs in clause 8.

7.1 Level 1: Custom SaaS application

Custom SaaS application is similar to the traditional application service provider (ASP) model of software delivery. Each customer has its own customized solution for SaaS application and runs its individual application instance on the cloud server. As illustrated in Figure 2, the custom application instance comprises the whole execution environment including the operating system (OS), the data management system and the middleware that are specific to each tenant, and the SaaS environment provider has to maintain multiple instances. This model is difficult to scale in order to satisfy the increasing requirement demands of customers, and it can be costly to operate.

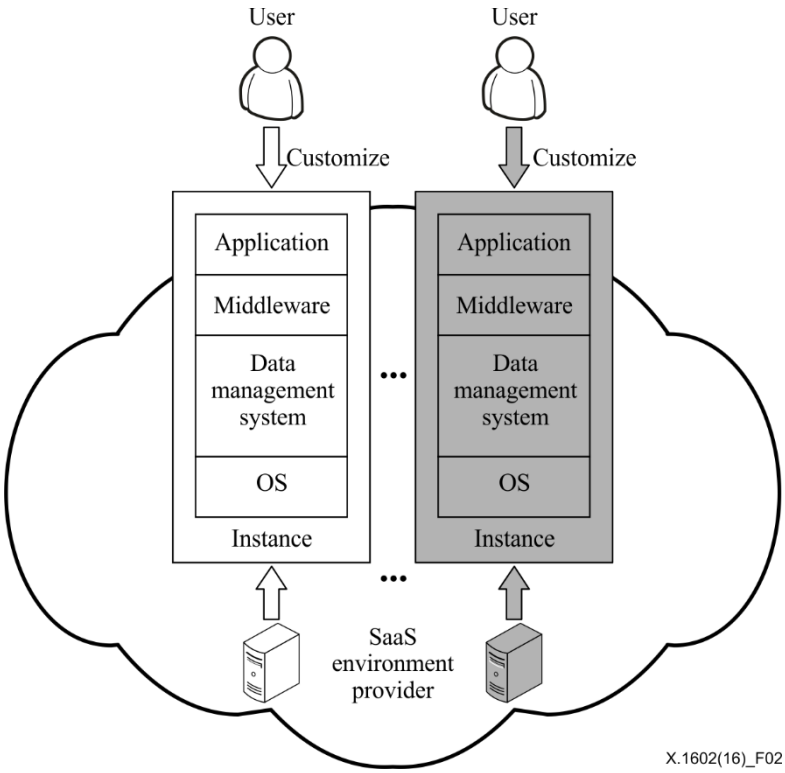


Figure 2 – Architecture of custom SaaS application

The typical client-server model applications can be easily transformed into custom SaaS applications by moving servers to the cloud with relatively little modification. The applications suitable for this scenario are usually developed with special requirements from the enterprise or organization. Top consideration will be given to security in the system itself, thus the usual way is to group a set of physical machines into a private zone and to deploy a data management system (which provides abstracted methods of persistence and operations for different kinds of data) and associated software on it. The system is solely for internal usage with strict access control. The template of application instance is the same for all customers, and it provides limited configuration ability. However, the instance for each customer is totally independent of any other instance.

7.2 Level 2: Configurable SaaS application

For some commonly used applications that are not customized, such as self-service website building system, SaaS application providers offer common templates for these applications and several sets of run-time environment for the instances of these applications. Based on the same template, customers are able to create multiple separated instances of the application by configuring the application's appearance and behaviour, which are deployed and executed on individual virtual or physical machines to meet their customized requirements. Application instances are isolated from each other. The architecture is shown in Figure 3.

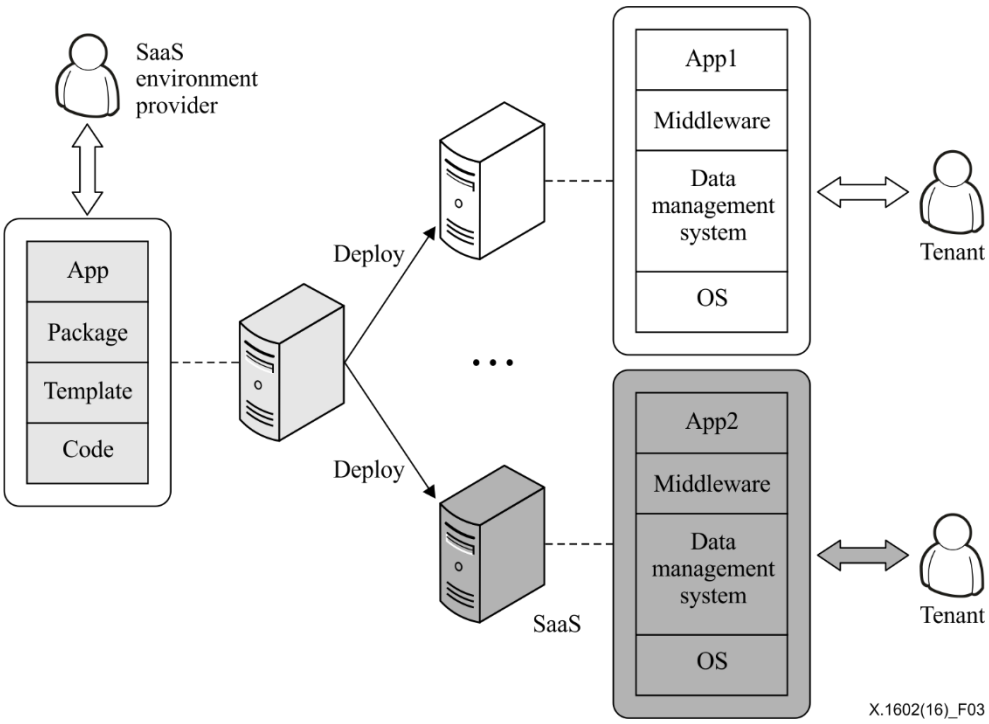


Figure 3 – Architecture of configurable SaaS application

The configurable SaaS application has the following characteristics:

1) Application in the initial deployment is a copy of a standard product, and tenants configure the application to suit their own requirements. However, the configuration options of the product are limited.

2) For SaaS application providers, any modifications to the product codes can be easily applied to all tenants immediately. However, only a little update or optimization to the product codes are suitable for each instance because the forward compatibility problem incurred by the update or optimization may occur.

3) Tenants store data in their own virtual machines or physical machines, which are isolated from each other. As a result, the SaaS environment provider has to provide sufficient resources such as storage to support a potentially large number of application instances running concurrently.

With the development and improvement of software technology, the application will be provided with enough configuration options to meet the users' customized requirements, and the configuration and usage process should be more intelligent and automated. SaaS application providers will divide the products into different versions to match different tenant levels.

7.3 Level 3: Multi-tenant SaaS application

In this level, with the help of configurable metadata, a SaaS application provider is able to provide a single instance that serves multiple tenants concurrently. The multi-tenancy can be enabled at different layers including OS, the data management system, middleware and application. A tenant identifier is introduced to distinguish between the different customers. When a database is used in a data management system, the database schema is extended to include tenant identity parameter for storing all customers' data into the same set of tables. A tenant identity is also needed in database queries in order to retrieve data for a specified customer. Figure 4 illustrates the general architecture of multi-tenant SaaS application.

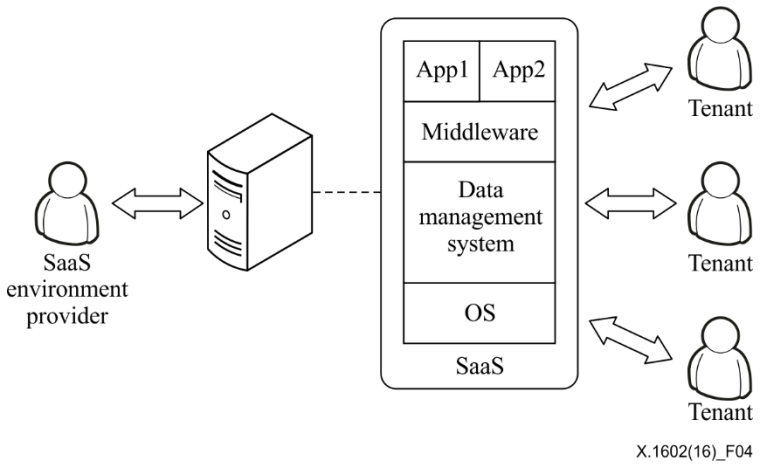


Figure 4 – Architecture of multi-tenant SaaS application

Business intelligence SaaS, for example customer relationship management (CRM), is considered to be a typical implementation of this level. Until now, more efforts have been taken to combine data warehousing and cloud computing with SaaS in order to provide online business intelligence applications. Data warehouse are hosted in the data centre, and business intelligence applications and the data models are predefined to use with very little customization. For the tenants, all they need to do are selecting the data elements required by business intelligence applications and defining the data mapping from data sources to the data warehouse and data model. The system will integrate data from multiple source systems into data warehouse to support the online analytical processing (OLAP) applications by using automatically generated scripts. Usually in the run time, a single instance of business intelligence application serves multiple tenants concurrently by using metadata techniques. Authorizations and security policies ensure that each customer's data and application access are isolated from that of other customers.

This level provides much more efficiency in the use of computing and storage resources, and therefore can be able to accommodate more tenants. It is also possible to achieve comparable performance, scalability and elasticity with the help of data partitioning and parallel techniques.

Configurability and multi-tenant efficiency are the distinctive characteristics for this level of SaaS application.

7.4 Level 4: Scalable SaaS application

Most public web service providers serve an arbitrarily large number of customers as multiple tenants. Consequently, each layer of the underlying platform architecture, from hardware to application, is required to be easily scalable for applications and services as shown in Figure 5. Hence, more tenants and more per-tenant users can be added without requiring additional re-architecting of the applications.

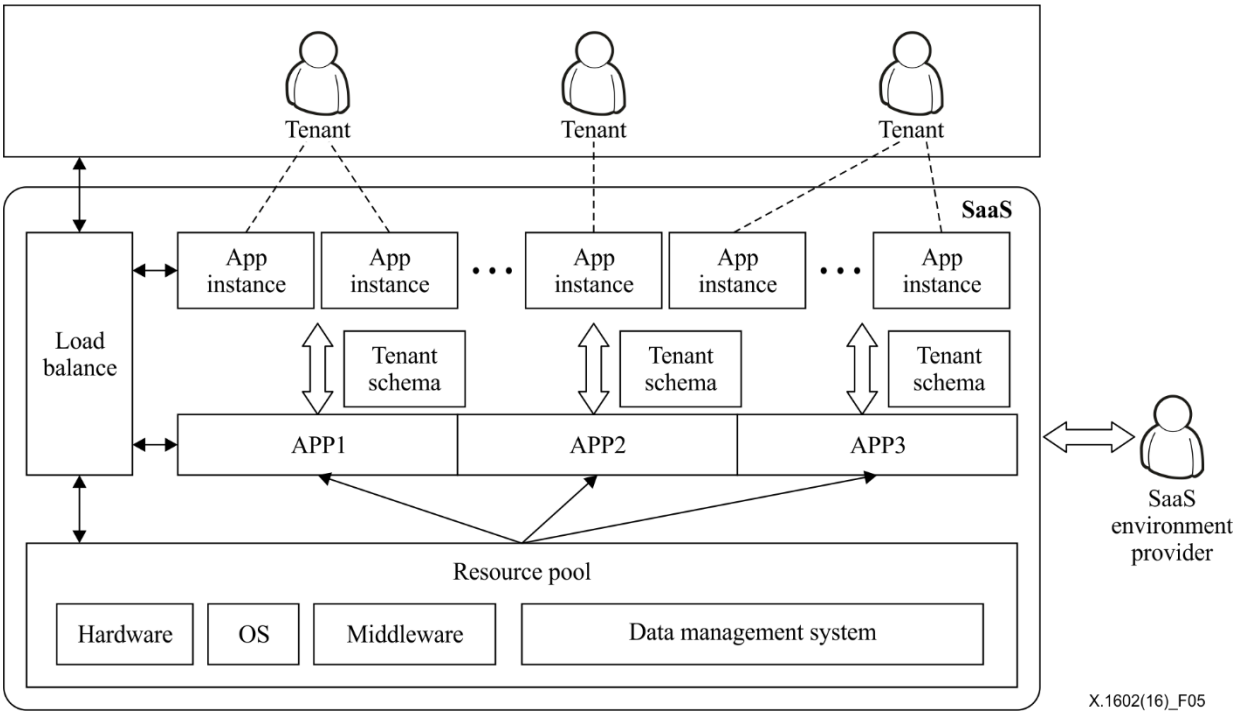


Figure 5 – Architecture of scalable SaaS application

For the application layer, when there is a new tenant, one or more application instances will be generated according to the tenant-specific requirements, or a suitable existing instance will be chosen in accordance with the requirement based on load balance mechanism. All the instances of the applications in such an environment are required to be created dynamically.

The underlying resources of scalable SaaS applications also support elastic scaling. Any hardware, middleware, software, and data are needed to be managed in the resource pool. The applications get all resources they need from the resource pool dynamically. New resources can be added without any recombination or re-architecting when needed.

There are multiple design considerations about the dynamic scaling technologies, including scaling choices, resource allocation, the service level agreement (SLA), etc. A new tenant can be executed as a single instance or can coexist with other tenants on a shared instance. Different instances, which run different types of tenants, can be allocated to varied resources. The SaaS environment provider should consider different SLAs for different tenants when using load balance and shared resources.

8 Security requirements for SaaS application environment

Figure 6 shows the relationship among the cloud service customer (CSC), CSP and CSN with respect to the SaaS application environment, in which CSP and CSN play different roles in performing different functions. CSN can serve CSP as a content provider, software provider, system integrator or auditor, while both CSN and CSP can develop applications for CSC. CSP and CSN have interfaces with the SaaS application environment, while CSC only interacts with applications built upon it. As a result, this Recommendation focuses mainly on the security requirements of the SaaS application environment for CSP and CSN in a different maturity model. The security requirements for the SaaS application environment originate from CSP and CSN as they need a SaaS application environment to have the capability to meet their demands on security.

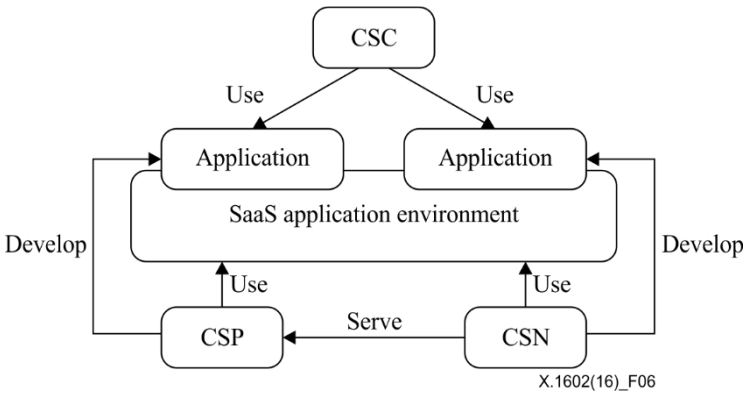


Figure 6 – Relationship among CSC, CSP and CSN

CSP and CSN have their own security requirements about the environment in different levels of SaaS. Table 2 illustrates the security requirements of CSP and CSN in the SaaS application environment. The requirements applicable for both CSP and CSN are the common requirements.

Table 2 – Security requirements of CSP and CSN in SaaS application environment

	SaaS application environment
Common requirements	Identity and access management, data security, security assessment and audit, interface security, security hardening.
CSP	Availability, service interoperability/portability guarantee, software assets protection, legal compliance, security verification for source codes.
CSN	Audit security, software security, software maintainability.

8.1 Common security requirements

For both CSP and CSN, they have several common security requirements in the SaaS application environment.

8.1.1 Identity and access management (IAM)

8.1.1.1 Identity management (IdM)

Multiple administrators and users are involved in the SaaS application environment, which can be accessed to and used internally (CSPs) and externally (CSNs). Identity Management (IdM) is needed not only to protect identities, but also to facilitate access management, authentication, authorization and transaction audit processes in such a dynamic and open SaaS application environment.

For all maturity models, IdM should enable the implementation of single sign-on and/or identity federation for the SaaS application environment using varied authentication mechanisms in different security domains.

8.1.1.2 Trust model

The SaaS application environment is required to incorporate an overall trust model for both multi-tenant level and scalable level. This trust model will enable the creation of islands and/or federations of trusted entities. Consequently, the SaaS application environment management system, the underlying resources, hypervisors, virtual machines and applications built upon the SaaS application environment will be able to authenticate the identities and authorized rights of other entities and components. Each island or federation of trust will be based on one or more trusted authorities (e.g., a public key infrastructure (PKI) certificate authority).

8.1.1.3 Access management

SaaS application environment administrators are required to provide mechanisms, which delegate authorization to tenants' administrators. The tenants' administrators grant access rights to their corresponding resources. The access management of such a SaaS application environment should support multiple access control models, such as identity based model, strategy based model, role based model, task based model, etc.

For custom and configurable level SaaS applications, a role-based access control model is a basic requirement. For instance, CSN, which supports to build a service from CSP, may be in charge of some applications but has no rights to administer the whole cloud service system. Besides, CSN may be allowed to access only a part of the resources with granted access rights. However, CSN can share its resource by providing application interfaces to other CSNs.

For the multi-tenant and scalable level, an integration of access control model for each individual and group is needed. For the role-based access control, shared resources among multiple tenants should be utilized according to task groups in a work flow and rights granted to those tasks. Thus, when these task groups are executed, the SaaS application environment should define the support task-based access control mechanism. This mechanism is used to make sure that access right of tenants to underlying resources could be timely granted and revoked, and underlying resources are prevented from unauthorized utilization.

8.1.2 Interface security

The SaaS application environment is required to secure interfaces open to CSPs or CSNs through which various kinds of cloud computing services are delivered or developed, and it is also required to secure communications based on these interfaces. Mechanisms that are available to ensure interface security include but are not limited to: unilateral/mutual authentication, integrity checksum, digital signature, etc.

8.1.3 Data security

8.1.3.1 Data isolation

Data can be isolated physically or logically. Physical data isolation should be accomplished by the access control of physical storages. It should require the SaaS application environment to store data of different tenants in different areas of physical storage, or implement the data accesses control for different tenants through access permission, data domain or any other methods. Logical data isolation implies that different tenants should be avoided to access others' data by the means of techniques such as virtualization, even if all the data are stored together.

For custom and configurable level SaaS applications, each tenant's data are separately stored and isolated from the others at the physical level.

For multi-tenant and scalable level SaaS applications, all tenant's data are stored in the cloud. Therefore, the SaaS application environment is required to be intelligent enough to segregate data from different tenants, and maintain isolation among different tenants' data at rest, at processing or at transmission. The boundary between each tenant should be ensured at the physical level or at the logical level, which depends on the required isolation granularity and the specific deployment of the cloud computing software and hardware.

8.1.3.2 Data confidentiality

In most cases, the tenant's data is on off-premise storage and utilization, and is subjected to exposure. Therefore, the SaaS application environment is required to support encryption mechanisms to ensure data confidentiality in transmission, during processing or out of occupation, and prevent data leakage due to security vulnerabilities in the application.

Data encryption service is required for all SaaS levels. Critical data is required to be encrypted to prevent exposure.

For multi-tenant and scalable level, as tenants' data should be stored in one database or even one big table, the SaaS application environment is required to provide an appropriate key management mechanism to ensure that the data cannot be cracked by other tenants.

8.1.3.3 Data integrity

Data including system data and user data, such as logs and configuration data, require the SaaS application environment to support integrity mechanisms to prevent them from unauthorized tampering in transmission, during processing or out of occupation.

System log and application log are required not to be modified. In this case, when either fault or misuse occurs, CSP and malicious software are prevented from concealing trace by modifying logs.

SaaS application may require CSCs to configure it on demand. The configuration data, such as configuration file, is also required to not be modified without authorization.

In the SaaS application environment, users' data is stored in the cloud which is managed by CSP. In this case, the verification of data integrity becomes a remarkable security requirement. Moreover, it is required to verify the integrity of massive data.

8.1.3.4 Data reliability

To support data reliability, the SaaS application environment is required to support data backup or redundancy mechanisms to ensure that tenants can access the data even if part of the cloud storage nodes lose efficacy.

Hosted data are required to implement a multiple-site backup; otherwise, the data will be completely ineffective. The SaaS application environment is required to have the ability to fully recover data and restore the data in time as well as keep data synchronism to ensure the consistency of multiple copies.

8.1.3.5 Data traceability and control

The SaaS application environment is required to ensure that physical location of data comply with the applicable law and local regulations, and with any restrictions in the legal agreements. The SaaS application environment is required to provide methods for CSCs to specify their data storage locations and verify that their data are appropriately placed.

Major concerns in a shared and virtualized infrastructure include not only loss of control by users over their data, but also locating data and controlling its whole life cycle. At any given time, the SaaS application environment is required to know exactly where both system data and user data are stored and processed, and provide verification of data location for CSCs. Both during and after usage, it shall not be possible for unauthorized third parties (including other CSPs) to trace the movement of the data.

8.1.4 Security assessment and audit

When underlying resources are changed, cracked or worked improperly, the SaaS application environment is required to be triggered to initiate security assessment procedure to evaluate whether or not specified security services or their applied security policies are affected, and indications or instructions are suggested to provide if they cannot satisfy predetermined conditions. An authorized party should be delegated to verify that the SaaS application environment complies with the applicable security requirements. Security assessment or security audit could be performed by CSC, CSP or a third party (CSN), and security certification could be performed by an authorized third party (CSN).

Independent trusted third parties should be used to provide reliable, independent and neutral security assessments or security audit.

8.1.5 Security hardening

The SaaS application environment aims mainly at offering secure service oriented multi-tenant development, deployment and an execution environment for SaaS applications. Security features of SaaS applications are in some cases insufficient or not well developed. The SaaS application environment is required to retrieve and verify those deficient security features of the SaaS applications, and provide differentiated security hardening mechanisms to enhance SaaS applications according to those deficient security features in order to meet the security requirements of different tenants in different contexts. The security features of applications consist of static security features when applications are in idle status and of dynamic security features when applications are running.

8.2 Security requirements of CSP

Besides common security requirements, CSP has specific security requirements in the SaaS application environment.

8.2.1 Availability

For CSP, the SaaS application environment is required to ensure that CSCs are in service all the time, which requires the handling of hardware/software failures, denial of service attacks, etc. It is essential to ensure the minimal downtime for CSCs.

8.2.2 Service interoperability/portability guarantee

When CSC wants to migrate all or a part of its system to another CSP, the original CSP requires the SaaS application environment to provide service interoperability and portability guarantee to minimize the damage to CSC's business. Besides, the SaaS application environment is required to guarantee that the related data will be deleted permanently on the previous CSP and will not be recovered by any other party.

8.2.3 Software assets protection

Software assets (such as applications, application-internal data, scripts, macros, function code library, software license, etc.) are required to be protected in the SaaS application environment.

CSP requires the SaaS application environment to protect the confidentiality and integrity of any software assets provided by CSP or CSN, which implies that these software assets cannot be copied, misappropriated, tampered with, given away, or otherwise used in an unauthorized manner.

8.2.4 Legal compliance

Though CSP can use data backup and redundancy mechanisms to ensure CSC's data reliability, the SaaS application environment is required to ensure that data copies shall not be retained for longer time than the permitted data retention period under the applicable data protection law.

8.2.5 Security verification for source codes

As in the SaaS application environment, CSN may provide the applications' codes, content or software to CSP, the SaaS application environment is required to provide mechanisms that assist CSP to verify the codes and to prevent malicious codes.

8.3 Security requirements of CSN

In the SaaS application environment, CSN can be an application developer, content provider, software provider, system integrator and auditor. Besides common security requirements, CSN has its own security requirements in the SaaS application environment.

8.3.1 Audit security

When CSN is an auditor, the SaaS application environment is required to provide mechanisms that assist CSN to collect audit events, logging and reporting information at the granularity of tenant and

application. These information are used to assure that CSP's service complies with governmental regulatory requirements and legal agreements contracted with tenants. The SaaS application environment is also required to provide mechanisms that assist CSN to ensure that the information collected and reported by the audit components within the CSP system are correct and not subject to tampering or manipulation.

Besides, the SaaS application environment is required to provide the capability for CSN to record the changes of important data and monitor the data availability online, in order to send a security alarm in time and therefore reduce losses.

8.3.2 Software security

When CSN is a cloud content or software developer, the SaaS application environment is required to provide mechanisms that assist CSN to ensure that their codes or other components supplied to CSP comply with any programming constraints required by the CSP**,** Besides, the codes or components should not contain malware or violate the integrity of CSP's cloud services.

8.3.3 Software maintainability

When CSN is a cloud software developer, the SaaS application environment is required to support mechanisms that assist CSN to provide source codes or other functionality for CSP's system. The source codes or functionality are required to contain versioning and other appropriate methods, in order to ensure that they can be maintained during the lifetime of the service. These methods include but are not limited to providing updates to fix known vulnerabilities, removing dependency on other components with known vulnerabilities, and increasing the overall system security.

Bibliography

[b-ITU-T X.1601] Recommendation ITU-T X.1601 (2014), *Security framework for cloud computing*.

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

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
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	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
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 and next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2016