

Internet Engineering Task Force  
Internet-Draft  
Intended status: Standards Track  
Expires: November 22, 2020

C. Yang, Ed.  
SY. Pan, Ed.  
South China University of Technology  
HB. Sun  
Inspur  
KM. Qu  
NetEase, Inc  
GQ. Han  
South China University of Technology  
May 21, 2020

The Standards on a Cloud Service Framework and Protocol for  
Construction, Migration, Deployment, and Publishing of Internet-Oriented  
Scalable Web Software Systems in Non-Programming Mode  
draft-yangcan-core-web-software-built-in-cloud-03

## Abstract

This draft mainly focuses on the scalable architecture and publishing protocol standard of REST-based SAAS cloud model Web software in non-programming mode, stipulates the data structure pattern and data exchange protocol for the construction and release of REST-based scalable Web cloud service software systems. Using the standardized framework and protocol, users can easily and quickly design their own software systems in the cloud, transfer and release data, which may make conventional software development so ease to improve the efficiency of complex database construction and server management. Without having to write codes under the standard framework, users can get consistent style background to create service, rapidly develop web application systems with the function of standard data management and data maintenance, and directly publish the software system to the end users of the Internet for access and use. And provide RESTful APIs to facilitate external access to required service resources. The framework can thus greatly shorten the software development life cycle, and save a great deal of development cost and maintenance overhead.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 22, 2020.

## Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1.	Introduction . . . . .	3
2.	Definitions and Terminology . . . . .	4
3.	Technical Specific Implementation Standards . . . . .	6
3.1.	System Architecture Diagram . . . . .	6
3.2.	SAAS Cloud Platform Related Implementation Standards . . . . .	8
3.2.1.	Application Management Module . . . . .	8
3.2.2.	End User Management Module . . . . .	8
3.2.3.	Relationship Management Module between the End User and the Application . . . . .	8
3.2.4.	Resource Management Module . . . . .	9
3.2.5.	Fine-grained Access Rights Management Module . . . . .	9
3.2.6.	Ways of Accessing Cloud Storage Resources . . . . .	10
3.2.7.	Format of Describing the Structured Data URI . . . . .	10
3.2.8.	Contents of the Ways of Accessing Cloud Storage Resources . . . . .	10
3.3.	Web Software Framework Related Implementation Standards . . . . .	11
3.3.1.	System Administrator View . . . . .	11
3.3.2.	User View . . . . .	12
3.4.	Implementation Standards on Construction Methods of Web Software . . . . .	12
3.4.1.	System Requirements Table and Data Record Exchange Table . . . . .	13
3.4.1.1.	System Requirements Table . . . . .	13
3.4.1.2.	Data Record Exchange Table . . . . .	14

3.4.2.	Steps of Reading System Requirements Table . . . . .	14
3.4.3.	Steps of Analyzing System Requirements Table . . . . .	15
3.4.4.	Contents of Modules for Users . . . . .	15
3.4.5.	Steps of Principle of Constructing Web Software System . . . . .	17
3.4.6.	Steps of Injecting Data Records into Web Software System . . . . .	19
3.4.7.	Submission of System Requirements Table . . . . .	19
3.4.8.	Submission of Data Exchange Table . . . . .	19
3.5.	Implementation Standards on Web Software Deployment . . .	20
3.5.1.	System Deployment . . . . .	20
3.5.2.	Model Deployment . . . . .	20
3.5.3.	User Deployment . . . . .	20
3.5.4.	User Group Deployment . . . . .	21
3.6.	Implementation Standards on Web Software Migration . . .	21
3.6.1.	Model Migration . . . . .	21
3.6.2.	User Migration . . . . .	21
3.6.3.	User Group Migration . . . . .	22
3.6.4.	User Group Deployment . . . . .	22
4.	Security Considerations . . . . .	22
5.	IANA Considerations . . . . .	23
6.	References . . . . .	23
6.1.	Normative References . . . . .	23
6.2.	Informative References . . . . .	23
6.3.	URL References . . . . .	24
	Authors' Addresses . . . . .	24

## 1. Introduction

With the rise of the Internet (especially the mobile Internet), Internet-oriented Web services become the mainstream of software architecture in the information age, but sophisticated web database construction and system management bring great increase of cost of software development, operation and maintenance. Whenever a system needs to be upgraded or updated, a large amount of coding workload is inevitable. With the rapid development of cloud computing and the growing popularity of SAAS, more and more users intend to migrate their software systems and deploy them to the cloud to get the cloud service resources they need. At the same time, there exists a common abstract model in the web information system, the system takes data management and maintenance as the core content, we can therefore extract its common features and provide a series of standards and requirements for the flexible construction and agile release of web software. In conclusion, cloud-oriented software migration and deployment technology has very important theoretical and practical significance. This draft focuses on the implementation standard of a cloud-oriented software migration and deployment technology, through which users can generate their own software system and migrate and

deploy to the cloud by simple operation. At the same time, it also provides an open RESTful API for users to obtain the required service resources through requests. From the development point of view, this kind of technology can transfer and deploy the software system conveniently and quickly, subtract the tedious and large number of coding work, and avoid repeated development. From the perspective of use, the convenient operation of this technology reduces the learning cost of users, has a strong usability, and has a unified front-end style. Users can access their own proprietary software system through the Internet, which is easy to promote. It also provides an expressive RESTful API that gives users the flexibility to adapt to complex and changing needs.

## 2. Definitions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

The following definitions are for terms used in the context of this document.

- o "SAAS": SAAS is an abbreviation of Software-as-a-Service. It is a model for providing software over the Internet. The manufacturer deploys the application software on its own server, and customers can purchase the required application services according to their actual needs through the Internet, pay the cost to the manufacturers according to the quantity of service and the length of the order, and obtain the services provided by the manufacturers through the Internet.
- o "Cloud Computing": Cloud computing is a pay-per-use model that provides available, convenient, on-demand network access, the user enters a configurable computing resource sharing pool (resources include network, server, storage, application software, services), these resources can be provided quickly, with little administrative effort or little interaction with service providers.
- o "Cloud Services": Cloud services are models of the addition, use, and delivery of Internet-based related services, often involving the provision of dynamically scalable and often virtualized resources over the Internet.
- o "Cloud Storage": Cloud storage is a network computer data storage model. Data is stored on multiple virtual hosts and is generally held by a third party rather than on a dedicated server.

- o "Cloud Migration": Cloud migration refers to the migration of enterprises from traditional platforms to cloud platforms. Compared with traditional application platforms, cloud computing platforms have the advantages of powerful computing power, storage capacity, diverse services and high cost performance.
- o "Software Migration": This draft is aimed at migrating existing software systems that meet user requirements to a lightweight information management software migration cloud service platform. Systems migrated to the platform can be accessed over the Internet.
- o "Software Deployment": This draft is aimed at rapidly deploying a new software system that meets user requirements on the lightweight information management software migration cloud service platform. The software system can be accessed through the Internet.
- o "Content Management": Using advanced information technology to help users create, store, transmit, publish, apply and analyze content, and use computer technology to evolve and extract it to create more valuable content for enterprises and individuals.
- o "NoSQL": Do not use SQL database. Such databases refer to other types of databases other than traditional relational databases. This type of database is more consistent and can handle very large and high concurrent data. The techniques in this draft use NoSQL to provide storage.
- o "Distributed DataBase": It is a distributed file storage-based database designed to provide a scalable, high-performance data storage solution for web applications.
- o "Web API": Web Application Programming Interface. An application programming interface is an interface that allows users to access information from another service and integrate the service into its own application.
- o "REST": Representational State Transfer. A software architecture style, the service is abstracted into a set of discrete resources. Through the HTTP protocol, different request methods (POST, DELETE, PUT, GET) add, delete, modify, and query the resources specified by the URI.
- o "Restlet Framework": The Restlet project provides a lightweight and comprehensive framework for "building a mapping between REST concepts and Java classes". It can be used to implement any kind of RESTful system, not just RESTful web services.

- o "Multi-Tenant Technology": A software architecture technology, whose key goal is to share the same system or program components in a multi-user environment and to ensure logical isolation of data between users.
- o "Privilege Control": The permission control used in this draft refers to adopting the role-based access control principle and the group configuration permission template method to achieve the separation of the authority control from the business logic.
- o "Certification Security": Authentication security is part of the security of the system. The authentication security system used in this draft adopts the Cookie mechanism on the client side and the Session mechanism on the server side, and uses the MD5 message digest algorithm to achieve the authentication security of the entire session.
- o "Non-Programming Mode": Non-Programming Mode is defined as a kind of software development mode in this draft, referring to develop a software without any codes.

### 3. Technical Specific Implementation Standards

The main goal of this software migration deployment technology is to help users quickly deploy a specific software system of their own on a designated lightweight information management software development and migration cloud service platform, or to migrate the user's traditional software systems to the designated lightweight information management software migration cloud service platform. We propose an implementation standard for this technology so that users who need it can use it for software development, migration, and deployment.

Note: The cloud platforms described in the draft are all designated lightweight information management software development and migration cloud service platforms. The cloud platform administrator refers to the administrator of the cloud service platform for the designated lightweight information management software. The system administrator is the administrator of the software system on the cloud platform, also known as the tenant. The user is a normal user under each software system, and is lower than the system administrator level user.

#### 3.1. System Architecture Diagram

Figure 1 shows the working diagram of the framework.

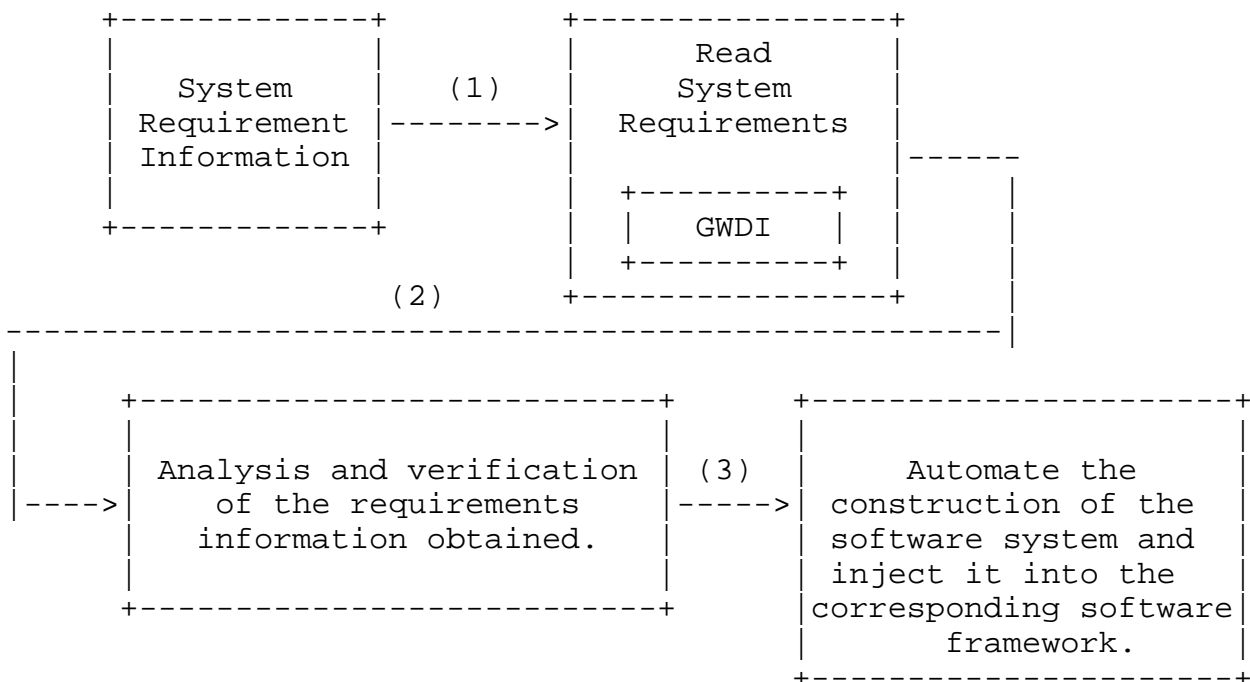


Figure 1:Reference Architecture

The standard requires a general Web Data Interface (GWDI). The system workflow is as follows:

**Step (1) Requirements Submission:** The user fills in the software system requirements information in the software system requirements table, and submits the request to GWDI through the browser.

**Step (2) Demand Acquisition:** GWDI obtains the software system requirements table submitted by the user, and reads the relevant information related to the requirements in the software system requirements table.

**Step (3) Information Verification:** GWDI analyzes and verifies the relevant information about the system requirements obtained.

**Step (4) System Generation:** Automatically construct a Web software system based on the verified requirements related information, and inject the system into the cloud platform.

### 3.2. SAAS Cloud Platform Related Implementation Standards

The technology MUST build a SAAS cloud platform that provides content management and data storage capabilities that can be rented to multiple tenants who are application developers. The platform includes the following modules:

- o Application Management Module
- o End User Management Module
- o Relationship Management Module between the End User and the Application
- o Resource Management Module
- o Fine-grained Access Rights Management Module

#### 3.2.1. Application Management Module

The application management module of the platform MUST support applications developed by management tenants, including:

- o Adding application
- o Deleting application

#### 3.2.2. End User Management Module

The end-user management module of the platform MUST support the management of end-user information for all applications of the tenant, including:

- o End-user registration
- o The end user changes the password
- o End-user logout

#### 3.2.3. Relationship Management Module between the End User and the Application

The relationship management module between the end user of the platform and the application MUST support the management of the use relationship between the end user and the application, including:

- o Establishment of user relationship



- o Elimination of the use relationship

#### 3.2.4. Resource Management Module

The resource management module of the platform MUST include at least:

- o Structured data resource management module

The structured data resource management module is used to manage the tenant's structured data in the cloud platform, including management of structured data table information, management of structured data table data resources. Each row of structured data resources will belong to a unique table, and will also belong to a unique user.

- o File resource management module

The file resource management module is used to manage the file resources of the tenant in the cloud platform. Each file resource will belong to the only application and the only user.

#### 3.2.5. Fine-grained Access Rights Management Module

The fine-grained access authority management module of the platform MUST support the management of end-user access rights to fine-grained resources within the tenant, including:

- o Structured form row data resource authority management module

The structured form row data resource authority management module includes:

- + For each structured table, configure the application of the structured table;
- + For each structured table, configure the access rights of the end user of the application to which the structured table belongs;
- + For each structured table, configure the access rights of the end user of the application that does not use the structured table;
- + For each structured table, configure the access rights of the end user to the single-row data resources that do not belong to the structured table.

- o Single file resource authority management module

The single file resource authority management module includes:

- + For each application, configure the access rights of the end user using the application to file resources within the application;
- + For each application, configure the access rights of the end user who does not use the application to the file resources within the application;
- + For each application, configure end-user access to files that are not part of the application.

### 3.2.6. Ways of Accessing Cloud Storage Resources

The platform SHOULD be able to access cloud storage resources in the following ways:

- o Through the POST, DELETE, PUT, GET requests initiated by REST API, the structured data resources specified by structured data URI are added, deleted, modified and queried.
- o Through the POST, DELETE and GET requests initiated by REST API, the file resources specified by file URI are uploaded, deleted and downloaded.

### 3.2.7. Format of Describing the Structured Data URI

The platform adopts the access method of the resources of the cloud storage platform described in 3.2.6, and it SHOULD be able to describe the structured data URI in the following format:

- o `/{structure table name}[/{filtered fields + combined form}/{conditions for filtered fields}]`;

Where, the structure table name is equivalent to the table name to be operated in the SQL statement. The contents in the brackets ([ ]) are optional: "NONE", which means all resources in the collection are manipulated. "YES", it represents an operation on a particular resource in the collection, equivalent to the WHERE clause in SQL.

### 3.2.8. Contents of the Ways of Accessing Cloud Storage Resources

The platform adopts a cloud storage platform resource access method described in 3.2.6, which SHOULD include:

- o Initiate PUT requests for structured data resources to modify the data resources specified by the URI, including: overwrite and

modify the specified fields, incremental changes to specified fields.

- o Initiate GET requests for structured data resources to query the data resources specified by the URI. Specific return formats can be set for the data resources that are queried, including: return by the specified field, return by the specified information page, and return by the specified field filtering.

### 3.3. Web Software Framework Related Implementation Standards

The standard MUST support a Web software framework that supports the software-as-a-service SAAS cloud pattern, where the software systems generated by the Web software framework are automatically deployed and distributed.

The Web software framework should specifically include the following views:

#### 3.3.1. System Administrator View

- o User Management

This view SHOULD be able to manage the user information of the Web software system, including the functions of adding users, removing users, adding user group information for users, and deleting user group information for users.

- o User Group Management

This view SHOULD be able to manage the user group information of the Web software system, including adding user groups, deleting user groups, adding user information for user groups, and deleting user information for user groups.

- o Model Management

This view SHOULD be able to manage the model information of the Web software system, including adding model information, deleting model information, adding model field information, deleting model field information, modifying model field information, and assigning different access rights to different user groups for the model.

### 3.3.2. User View

- o Data Management

This view SHOULD be able to manage the data table records of the Web software system, including adding data records, deleting data records, modifying data records, searching data records, and obtaining all data records.

- o Data Statistics

This view SHOULD be able to make statistics on the data table records of the Web software system, including functions such as record statistics, maximum statistics, minimum statistics, and average statistics for data table integral fields.

- o Import Excel and Other Format Tables

This view SHOULD be able to import data from Excel tables (sheets) and other format tables into the corresponding tables of the Web software system.

- o Export Excel and Other Formats

This view SHOULD be able to export data from the table of the Web software system to Excel tables (sheets) and tables in other formats.

- o Advanced Search

This view SHOULD be able to perform multiple logical combinatorial searches for multiple fields of the table to query the record information that satisfies the search criteria.

### 3.4. Implementation Standards on Construction Methods of Web Software

This standard SHALL support a table-driven software system automatic construction method within cloud mode, which is characterized by table submission, reading and verification, a Web software generation framework and a set of Web software generation workflow, in which the Web software generation framework should be the framework described in requirement 3.3.

### 3.4.1. System Requirements Table and Data Record Exchange Table

The table described in this method MUST include the system requirements table and the data record exchange table:

#### 3.4.1.1. System Requirements Table

The system requirements table MUST include the following parts:

- o System Administrator Information Section
  - \* System administrator ID
  - \* System administrator password
  - \* System title
- o User Group Information Section (multiple user groups can be customized in this table according to requirements)
  - \* User group ID
- o User Information Section (multiple users can be customized in this table according to requirements)
  - \* User ID
  - \* The user name
  - \* User password
  - \* The user belongs to the group (if filled in, it must be the item contained in the user group ID of the user group information section)
- o Model Information Section (multiple models can be customized in this table according to requirements)
  - \* Read and write permission of the user group to which it belongs
    - Permissions include: readable and writable, readable and unwritable, unreadable and unwritable
  - \* Other users' read and write rights
    - Permissions include: readable and writable, readable and unwritable, unreadable and unwritable

- \* Model name
- \* Subscriber ID (must be the item contained in the user ID in the user information section)
- \* Group ID (must be the item contained in the user group ID of the user group information section)
- \* Field type

Field types include: Text, Float, Integer, Link, Date, Datetime, Boolean.

- \* Field name
- \* Whether empty is allowed

Values include: TRUE and FALSE. Where TRUE means allowed to be empty;

#### 3.4.1.2. Data Record Exchange Table

The data record exchange table MUST include the following parts:

- o Field information section (all field names of the model to be exchanged for data records can be set according to requirements)
- o Data records section (all data records that need to be injected into the model can be added as required)

#### 3.4.2. Steps of Reading System Requirements Table

In the workflow of the Web software generation principle of this method, it MUST include the following steps:

1. Using the traversal unit of system requirements table to read through each cell of the system requirements table.
2. Using the reading unit of system administrator information to read system administrator ID, system administrator password, system title and other information when traversing system administrator information.
3. Using the reading unit of user group information to read user group ID information when traversing user group information.
4. Using the reading unit of user information to read user ID, user name, user password, user group ID (the user group ID must be

included in user group information) when traversing user information.

5. Using the reading unit of model information to read the information such as the reading and writing rights of the user group to which it belongs, the reading and writing rights of other users, the model name, the inputting ID, the group ID to which it belongs, the field type, the field name and whether it is allowed to be empty, when traversing the model information.

#### 3.4.3. Steps of Analyzing System Requirements Table

In the workflow of Web software generation of this method, it MUST include the analysis and verification flow of the relevant information about the system requirements acquired by the Web software framework, including the following parts:

1. Data type and format validation for information related to system requirements obtained.
2. As for the user information part of the system requirements table, if the user's group item is not empty, the framework should judge whether the filled value is included in the user group information part of the system requirements table.

#### 3.4.4. Contents of Modules for Users

The workflow of the Web software generation principle of this method MUST include the module for users to use in the cloud environment where the corresponding Web software framework is located, which must include the following contents:

1. The framework should use the API of dealing with system administrator information in the Read\_Demand class to read the system administrator section in the software system requirement table, each cell of which will be read successively, and then register the software system using the super administrator privileges with its ID and password and the system name, and further create relevant database entities of the software system in the underlying database of the Web software framework. At the same time, such four collections should be create in the database as user group, user, schema, and data.
2. The framework should use the API of dealing with user group information in the Read\_Demand class to read the user group information in the software system requirement table, each cell of which be read successfully, and generate json-formatted strings based on the group ID information obtained, and then

insert them into the user group set of database entities created in 1.

3. The framework should use the API of dealing with user information in the `Read_Demand` class to read the user information in the software system requirement table, each cell of each row of which be read successfully, and generate json-formatted strings according to the user information obtained, and then insert them into the user set of database entities created in 1.
4. The framework should read the model information section of the system requirements table through the API used to process model information in the requirements table reading class (`Read_Demand` class), read through the filled content in the model information cells, generate json-formatted strings based on model information, and insert them into the schema set of database entities created in 1.
5. The framework should operate the user group data in the group set of database entities in the underlying database through the user group management class (`Manage_Group`) and return the results to the interface of the Web software framework. The `Manage_Group` class implements the `GManage_Group` interface defined in the Web software framework.
6. The framework should operate the user data in the user set of the database entity in the underlying database through the user management class (`Manage_User`) and return the results to the interface of the Web software framework. The `Manage_User` class implements the `GManage_User` interface defined in the Web software framework.
7. The framework should operate the model data in the schema set of database entities in the underlying database through the model management class (`Manage_Schema`) and return the results to the interface of the Web software framework. The `Manage_Schema` class implements the `GManage_Schema` interface defined in the Web software framework.
8. The framework should operate the data records in the data set of database entities in the underlying database through the data management class (`Manage_Data`) and return the results to the interface of the Web software framework. The `Manage_Data` class implements the `GManage_Data` interface defined in the Web software framework.



### 3.4.5. Steps of Principle of Constructing Web Software System

This method MUST satisfy the working principle of constructing Web software system automatically according to the system demand table. The principle includes the following steps:

1. The customer shall fill in the system requirements form described in 3.4.1.1 according to the requirements.
2. The Web software framework obtains the system requirements table uploaded by the customer in 1, and verifies whether the system requirements table conforms to the verification standard. If it does not conform to the verification standard, it prompts the user to have the wrong format, please re-upload it.
3. The Web software framework reads the system administrator information in the system requirements table.
4. Create a new system in the corresponding Web software framework according to the information read in 3, in which the administrator of the system is the system administrator in 3, and the system name is the system name in 3.
5. The Web software framework determines whether there is any part of user group information in the system requirements table. If not, it jumps to 8; if there is, it goes 6.
6. The Web software framework reads information about user groups in the system requirements table, including user group IDs.
7. According to the user group information read in 6, the user group is generated in the system created in 4, and the relevant information of the user group is the relevant information read in 6.
8. The Web software framework determines whether there is any relevant information about users in the system requirements table. If there is no information about users, it will jump to 11; if there is, it will go to 9.
9. The Web software framework reads relevant information of users in the system requirements table, including user ID, user name, user password, and user group.
10. The Web software framework generates users in the system created in 4 according to the user information read in 9, and the relevant information of the users is the information read in 9.

11. The Web software framework determines whether the relevant information of the model part exists in the system requirements table. If it does not, it will jump to 20, and if it does, it will go to 12.
12. The Web software framework reads the number of models in the system requirements table and assigns a current count of 0, then go to 13.
13. The Web software framework reads the relevant information of the model in the system requirements table, including the model name, model entry person, model group ID, group permissions, other user rights and other information.
14. According to the information related to the model read in 13, the model is generated in the system created in 4. The model name, model entry person, group ID of the model, group permission and other user rights are the information read in 13.
15. The Web software framework determines whether there is any information related to the structural fields in the model in the system requirements table. If there is no information, it will jump to 18; if there is, it will go to 16.
16. The Web software framework reads the relevant information of structural fields in the model in the system requirements table, including obtaining the name of structural fields of the model, field types, whether to allow null identification, default values and alternative values, etc.
17. The Web software framework creates model fields in the specified model according to the relevant information of structural fields in the model read in 16, and the field information is the information read in 16.
18. Current model number +1.
19. The Web software framework determines whether the current count is less than the number of models read in 12; if it is less than, it will jump back to 13; if it is not less than, it will go to 20.
20. The Web software framework generates the target Web software system.

#### 3.4.6. Steps of Injecting Data Records into Web Software System

The workflow of the Web software generation principle of this method MUST include the workflow of automatically injecting data records into the Web software system according to the data record exchange table, including the following steps:

1. The customer shall fill in the data record exchange form as required.
2. The Web software framework gets the data record exchange table uploaded by the customer, and verifies whether the data record exchange form meets the requirements of the data record exchange form. If it does not meet the requirements, it prompts the user to have the wrong format, please re-upload it.
3. The Web software framework reads the model field information in the data record exchange table.
4. The Web software framework determines the required data exchange model according to the model field information read in 3.
5. The Web software framework reads the data record information in the data record exchange table line by line and injects it into the model determined in 4 successively.

#### 3.4.7. Submission of System Requirements Table

The submission of the system requirements table SHALL include:

1. Online submission: according to the Web form format provided by the Web software generation framework, users fill in the system requirements information online and submit it to the Web software generation framework;
2. Offline submission: according to the offline form format template agreed by the Web software generation framework, users fill in the required software system demand information offline and submit it to the Web software generation framework by file upload.

#### 3.4.8. Submission of Data Exchange Table

The submission of the data exchange table SHALL include:

1. Online submission: after the user constructs a new Web software system according to the Web software generation framework, the user enters data into the Web page through the newly generated

Web software system, enters data information online and delivers it to the generated software system;

2. Offline submission: after the user constructs a new Web software system according to the Web software generation framework, the user enters the data record information offline and submits it to the generated software system through the data record exchange table template format provided by the newly generated Web software system by means of file upload.

### 3.5. Implementation Standards on Web Software Deployment

This framework MUST support the rapid deployment of a new set of software systems that meet the requirements on the cloud platform, where the deployment of software systems should support the deployment of the following modules:

#### 3.5.1. System Deployment

This framework SHOULD support two ways to deploy a system:

- o Supports registering users' own systems through cloud platform administrators to create system administrators belonging to users.
- o Supports users to register their own system through cloud platform and register their own system administrator.

#### 3.5.2. Model Deployment

This framework SHOULD enable system administrator users to enter their own software system and deploy the created models on the cloud platform, where the created models are configured as follows:

- o Model name, input personnel, and user group read and write permission are necessary when the model is created.
- o Field name, field type, whether the field is allowed to be empty, field default value, and field alternative value in the model are optional fields for creating the model.

#### 3.5.3. User Deployment

This framework SHOULD support to create new users affiliated to the software system created by a system administrator user on the cloud platform, where the creation of users is configured with user ID, user nickname, and user password.

#### 3.5.4. User Group Deployment

This framework SHOULD support to create new users affiliated to the software system created by a system administrator user on the cloud platform, where the creation of user groups is configured as follows:

- o The user group ID is necessary.
- o It is optional to select many users into the user group, but one user group must have one use at least.

#### 3.6. Implementation Standards on Web Software Migration

This framework SHOULD support the migration of existing software systems that meet the given requirements to the cloud platform, where the migration of software systems should support the migration of the following modules:

##### 3.6.1. Model Migration

This framework SHOULD support the migration of the specified model from the source system to the target software system, where the source container of the model includes at least one of the following:

###### 1. Relational Database

- \* SQLSERVER
- \* MYSQL
- \* ORACLE

###### 2. Excel File. Here, the model migration may support two modes:

- \* Migrating data from source to specify model structure
- \* Migrating data from source to specify the model structure and migrating data within the model

##### 3.6.2. User Migration

This framework SHOULD support the migration of users in the source system and their access rights to the target software system, where the user's source container includes at least one of the following:

###### 1. Relational Database

- \* SQLSERVER

- \* MYSQL

- \* ORACLE

## 2. Excel File

### 3.6.3. User Group Migration

This framework SHOULD support the migration of user groups in the source system and their access rights to the target software system, where the user group's source container includes at least one of the following:

#### 1. Relational Database

- \* SQLSERVER

- \* MYSQL

- \* ORACLE

#### 2. Excel File

### 3.6.4. User Group Deployment

This framework SHOULD enable system administrator users to enter their own software systems and create user groups on the cloud platform, where the creation of user groups is configured as follows:

1. A user group ID created for identifying the user group is of necessity.

2. It is optional to select many users into a user group, but one user group must have one use at least.

## 4. Security Considerations

This draft proposes an implementation standard for software migration deployment technology for cloud environments, and does not make special requirements for the security of the technology itself. However, the security of the cloud platform and the security between different users in the software system are required. The security of the cloud platform is mainly authentication security, and can also be considered as session security to ensure the security of the user during using software. The security of different users in the system is called permission control. Data isolation between different systems, different user groups in the same system, and different resource access rights between different users should be considered.

## 5. IANA Considerations

This memo includes no request to IANA.

## 6. References

### 6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", RFC 6690, DOI 10.17487/RFC6690, August 2012, <<https://www.rfc-editor.org/info/rfc6690>>.
- [RFC7705] George, W. and S. Amante, "Autonomous System Migration Mechanisms and Their Effects on the BGP AS\_PATH Attribute", RFC 7705, DOI 10.17487/RFC7705, November 2015, <<https://www.rfc-editor.org/info/rfc7705>>.
- [RFC8206] George, W. and S. Murphy, "BGPsec Considerations for Autonomous System (AS) Migration", RFC 8206, DOI 10.17487/RFC8206, September 2017, <<https://www.rfc-editor.org/info/rfc8206>>.

### 6.2. Informative References

- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <<https://www.rfc-editor.org/info/rfc2629>>.
- [RFC3347] Krueger, M. and R. Haagens, "Small Computer Systems Interface protocol over the Internet (iSCSI) Requirements and Design Considerations", RFC 3347, DOI 10.17487/RFC3347, July 2002, <<https://www.rfc-editor.org/info/rfc3347>>.
- [RFC6208] Sankar, K., Ed. and A. Jones, "Cloud Data Management Interface (CDMI) Media Types", RFC 6208, DOI 10.17487/RFC6208, April 2011, <<https://www.rfc-editor.org/info/rfc6208>>.
- [RFC7322] Flanagan, H. and S. Ginoza, "RFC Style Guide", RFC 7322, DOI 10.17487/RFC7322, September 2014, <<https://www.rfc-editor.org/info/rfc7322>>.

[RFC7491] King, D. and A. Farrel, "A PCE-Based Architecture for Application-Based Network Operations", RFC 7491, DOI 10.17487/RFC7491, March 2015, <<https://www.rfc-editor.org/info/rfc7491>>.

### 6.3. URL References

[idguidelines] IETF Internet Drafts editor, "<http://www.ietf.org/ietf/lid-guidelines.txt>".

[idnits] IETF Internet Drafts editor, "<http://www.ietf.org/ID-Checklist.html>".

[ietf] IETF Tools Team, "<http://tools.ietf.org>".

[ops] the IETF OPS Area, "<http://www.ops.ietf.org>".

[xml2rfc] XML2RFC tools and documentation, "<http://xml.resource.org>".

### Authors' Addresses

Can Yang (editor)  
South China University of Technology  
382 Zhonghuan Road East  
Guangzhou Higher Education Mega Centre  
Guangzhou, Panyu District  
P.R.China

Phone: +86 18602029601  
Email: [cscyang@scut.edu.cn](mailto:cscyang@scut.edu.cn)

Shiying Pan (editor)  
South China University of Technology  
382 Zhonghuan Road East  
Guangzhou Higher Education Mega Centre  
Guangzhou, Panyu District  
P.R.China

Email: [201721041376@scut.edu.cn](mailto:201721041376@scut.edu.cn)



Haibo Sun  
Inspur  
163 Pingyun Road  
Guangzhou, Tianhe District  
P.R.China

Email: sunhb@inspur.com

Kemin Qu  
NetEase, Inc  
Netease Building, Building E, Guangzhou Information Port  
16 Keyun Road  
Guangzhou, Tianhe District  
P.R.China

Email: gzqkemin@corp.netease.com

Guoqiang Han  
South China University of Technology  
382 Zhonghuan Road East  
Guangzhou Higher Education Mega Centre  
Guangzhou, Panyu District  
P.R.China

Email: csgqhan@scut.edu.cn