

CFDAB

国家食品药品监管信息化标准

CFDAB/T-0302-2013

第 8 部分：应用支撑平台通用技术规范

Part 8: General technical specifications for application support platform

(征求意见稿 V1.0)

2013 - xx-xx 发布

20xx - xx-xx 实施

国家食品药品监督管理总局信息办 发布

目 次

前 言.....	III
1. 范围.....	1
2. 规范性引用文件.....	1
3. 术语和定义.....	1
4. 总体要求.....	3
5. 应用支撑平台总体架构.....	4
6. 基于应用支撑平台建设基本要求.....	5
7. 统一接入认证规范.....	5
7.1 概述.....	5
7.2 单点登录规范.....	6
7.3 统一用户管理规范.....	6
7.4 权限管理规范.....	7
7.4.1 权限基本要求.....	7
7.4.2 系统访问权限.....	7
7.4.3 功能访问权限.....	8
8. 应用支撑安全规范.....	8
8.1 概述.....	8
8.2 应用系统日志记录接口.....	8
8.3 系统管理记录接口.....	9
8.4 用户行为记录接口.....	9
9. 服务建设规范.....	10
9.1 服务建设基本要求.....	10
9.2 服务设计规范.....	10
9.2.1 概述.....	10
9.2.2 服务识别.....	10
9.2.3 服务定义.....	10
9.3 服务封装规范.....	12
9.3.1 服务封装原则.....	12
9.3.2 服务封装方式.....	13

- 9.4 服务开发规范 13
 - 9.4.1 基本要求 13
 - 9.4.2 Web Service 命名规范 13
 - 9.4.3 WSDL 编写说明 14
 - 9.4.4 适配器开发规范 15
- 9.5 服务集成规范 16
 - 9.5.1 企业服务总线（ESB） 16
 - 9.5.2 服务描述 17
 - 9.5.3 服务注册/发布 17
 - 9.5.4 服务发现/调用 17
 - 9.5.5 服务编排 17

前 言

《国家食品药品监管信息化标准》发布以下几个部分：

- 第 1 部分：标准化指南；
- 第 2 部分：基本术语；
- 第 3 部分：标准编写规则；
- 第 4 部分：信息分类与代码；
- 第 5 部分：数据元与代码集；
- 第 6 部分：数据集元数据；
- 第 7 部分：软件开发过程规范；
- 第 8 部分：应用支撑平台通用技术规范；
- 第 9 部分：数据共享与交换接口；
- 第 10 部分：数据库设计规范；
- 第 11 部分：网络及通信；
- 第 12 部分：信息安全；
- 第 13 部分：工程管理。

本标准第 8 部分。

本标准依据 GB/T1.1—2009 给出的规则起草。

本标准由国家食品药品监督管理总局信息中心提出。

本标准由国家食品药品监督管理总局信息办归口。

本标准起草单位：国家食品药品监督管理总局信息中心、广东省食品药品监督管理局、中科软科技股份有限公司。

第 8 部分：应用支撑平台通用技术规范

1. 范围

本规范规定了国家食品药品监管信息化工程中使用应用支撑平台的技术规范要求。本规范从应用支撑平台总体架构、基于应用支撑平台建设基本要求、统一接入认证规范、应用支撑安全规范、服务设计规范、服务开发规范、服务集成规范等多方面对工程的开发提出要求。

本规范适用于国家食品药品监管信息化工程的设计、开发、建设实施和管理维护。

2. 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

IEC 61968 Enterprise Service Bus Implementation Profile Integration

GB/T 20271-2006 信息系统通用安全技术要求

GB/T 22239-2008 信息系统安全等级保护基本要求

GB/T 25070-2010 信息系统等级保护安全设计技术要求

3. 术语和定义

CFDAB/T-0102-2013 收录的部分术语和定义适用于本标准。为了便于使用，以下重复列出了 CFDAB/T-0102-2013 中的部分术语和定义。

3.1

面向服务的体系结构 Service-Oriented Architecture (SOA)

面向服务架构，它可以根据需求通过网络对松散耦合的粗粒度应用组件进行分布式部署、组合和使用。服务层是 SOA 的基础，可以直接被应用调用，从而有效控制系统中与软件代理交互的人为依赖性。

SOA 是一种粗粒度、松耦合服务架构，服务之间通过简单、精确定义接口进行通讯，不涉及底层编程接口和通讯模型。SOA 可以看作是 B/S 模型、XML/Web Service 技术之后的自然延伸。

[CFDAB/T 0102-2013 定义 3.2.8.1]

3.2

企业服务总线 Enterprise Service Bus (ESB)

企业服务总线是传统中间件技术与 XML、Web 服务等技术结合的产物。ESB 提供了网络中最基本的连接中枢，是构筑企业神经系统的必要元素。ESB 的出现改变了传统的软件架构，可以提供比传统中间件产品更为廉价的解决方案，同时它还可以消除不同应用之间的技术差异，让不同的应用服务器协调运

CFDAB/T-0302-2013

作，实现了不同服务之间的通信与整合。从功能上看，ESB 提供了事件驱动和文档导向的处理模式，以及分布式的运行管理机制，它支持基于内容的路由和过滤，具备了复杂数据的传输能力，并可以提供一系列的标准接口。

[CFDAB/T 0102-2013 定义 3.2.8.2]

3.3

Web Service

Web service 是一个平台独立的、松耦合的、自包含的、基于可编程的 web 应用程序，可使用开放的 XML 标准来描述、发布、发现、协调和配置这些应用程序，用于开发分布式的互操作的应用程序。

[CFDAB/T 0102-2013 定义 3.2.8.3]

3.4

Portal

portal 是一个基于 web 的应用，它能提供个性化定制、单点登录、不同源内容的聚合和信息系统的表示层集中。聚合是整合不同 web 页面数据的过程。为了提供用户定制的内容，portal 可能包含复杂的个性化特征，为不同用户创建内容的 portal 页，可能包含不同的 portlet 集。

[CFDAB/T 0102-2013 定义 3.2.8.4]

3.5

Portlet

portlet 是基于 java 的 web 组件，处理 request 并产生动态内容。portlet 产生的内容称为片段，即一段遵守中心规则的标记（比如 html, xhtml, wml(无线标记语言)），多个片段可以聚合成一个完整的文档。多个 portlet 聚合在一起，组成 portal 页。portlet 容器控制 portlet 的生命周期。

[CFDAB/T 0102-2013 定义 3.2.8.5]

3.6

JSR-168

portlet 规范 Java Standardization Request 168

JSR-168 是为了实现 portal 和 portlet 的互操作。它定义了 portlet 和 portlet 容器之间的和约，让 portlet 实现个性化、表示和安全的 api 集。规范还定义了怎样在 portlets 应用中打包 portlets。

[CFDAB/T 0102-2013 定义 3.2.8.6]

3.7

访问控制列表 Access Control List (ACL)

国家食品药品监管信息化工程 ACL 中的内容是菜单列表，通过对菜单列表的控制，实现功能模块的权限控制。

[CFDAB/T 0102-2013 定义 3.2.8.7]

3.8

基于角色的访问控制 Role-Based Access Control (RBAC)

在 RBAC 中, 权限与角色相关联, 用户通过成为适当角色的成员而得到这些角色的权限, 这就极大地简化了权限的管理。在一个组织中, 角色是为了完成各种工作而创建, 用户则依据它的责任和资格来被指派相应的角色, 用户可以很容易地从一个角色被指派到另一个角色。角色可依新的需求和系统的合并而赋予新的权限, 而权限也可根据需要而从某角色中回收。角色与角色的关系可以建立起来以囊括更广泛的客观情况。

[CFDAB/T 0102-2013 定义 3.2.8.8]

3.9

应用程序接口 Application Programming Interface (API)

API 是一个应用系统留给应用程序的一个调用接口, 应用程序通过调用应用系统的 API 而使应用系统去执行应用程序的特定功能。

[CFDAB/T 0102-2013 定义 3.2.8.9]

3.10

Web 服务的描述语言 Web Services Description Language (WSDL)

WSDL 是 Web Service 的描述语言, 是一种接口定义语言, 用于描述 Web Service 的接口信息等, WSDL 文档可以分为顶部和底部两部分, 顶部分由抽象定义组成, 而底部分则由具体描述组成。

[CFDAB/T 0102-2013 定义 3.2.8.10]

4. 总体要求

应用支撑平台为国家食品药品监管信息化工程建设的基础支撑平台, 国家食品药品监管信息化工程各应用系统建设应遵循应用支撑平台的要求进行建设, 对于无法使用应用支撑平台的单位, 可遵循本规范进行应用支撑平台建设。



图 1：应用支撑平台总体架构图

如图 1 所示，应用支撑平台按照支撑的能力细分，基本划分为三层，各层之间通过一系列集成手段有机的集成在一起，各功能组件独立发挥自身的作用，又通过无缝的集成手段相互配合，为国家食品药品监管信息化工程提供层次清晰、功能全面、扩展良好的支撑。

- a) 平台运行基础支撑环境主要为应用支撑平台以及应用系统提供底层的运行环境，包括基础网络环境、操作系统、数据库管理系统、J2EE 应用服务器等。
- b) 信息资源服务平台层，存储应用系统的所有数据。
- c) 应用支撑平台是国家药品监管信息系统的基础设施，包括应用系统开发、集成、部署和管理的环境及相应规范。提供的应用基础开发环境由一系列组件组成，包括相应的开发环境、开发组件、开发规范等，通过配置形成对应用系统的支撑；使基于应用支撑平台之间的应用系统能够在一个严格定义的、

统一的环境中运行，从而提高系统互连、互通、互操作和信息共享的能力。

SOA 应用支撑平台按功能类型分为应用集成类组件、数据集成类组件、基础工具类组件和服务类组件。应用支撑平台应具有统一性、安全性、灵活性等特点，即必须建立在安全支撑平台基础之上，通过统一的接入认证、安全 WEB 门户、可信信息交换和可信应用传输以及系统间的接口规范，来为应用系统提供安全服务和可信应用环境，实现安全互联互通和安全信息共享，安全运行业务应用。

6. 基于应用支撑平台建设基本要求

应用支撑平台应提供应用集成类、基础工具类、数据集成类、系统服务类等组件，应用系统建设过程中应使用这些组件进行系统的建设，其中：

a) 应用集成类

负责应用系统的集成，实现统一系统界面、统一系统权限、统一用户管理的功能，并提供服务的管理与维护功能这类组件需要各应用系统根据组件要求进行使用。

b) 数据集成类

负责数据的抽取、传输、交换、共享等服务支撑，完成数据的备份等功能。数据集成类组件可包括：实时数据同步、ETL 工具、LDAP 服务组件、消息服务组件等。

c) 基础工具类

负责向应用系统提供基础功能的支撑，辅助完成系统功能的开发。基础工具类组件可包括：内容管理、全文检索、报表管理、工作流管理、智能表单、短信平台等等。

d) 系统服务类

系统服务类的组件是由各个应用系统开发的基础数据服务组成，主要实现应用系统之间的数据服务。服务类组件以 XML 数据格式和 HTTP 页面两种方式进行数据的展现，并由应用集成类组件的服务管理与维护功能统一管理。

7. 统一接入认证规范

7.1 概述

统一接入认证基于国家食品药品监督管理局建立的 CA 系统进行，地方单位可基于国家食品药品监督管理局的 CA 系统建立地方 RA 系统或二级 CA 系统，实现统一接入认证。

应用支撑平台应提供认证管理组件，实现应用系统的单点登录和统一用户管理功能，结合安全认证网关提供系统级别的统一权限管理。各应用系统应遵循统一接入认证规范的要求，使用单点登录实现应用系统的登录认证，并通过 API 获取登录用户信息、使用统一用户管理提供的用户和机构信息、使用权限管理维护应用系统登录权限。

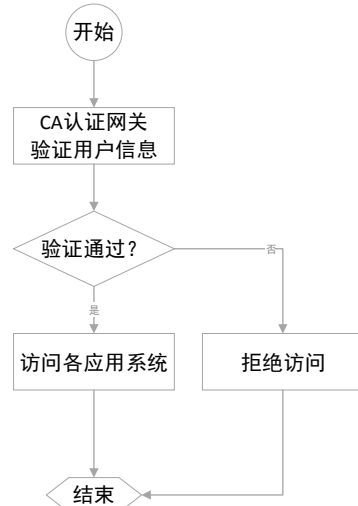


图 2：统一接入认证流程图

如图 2 所示，用户发起服务请求后，首先由 CA 安全认证网关进行用户信息验证，如果用户信息验证通过，则可访问具有权限的各应用系统，如果验证不通过则拒绝用户的访问。

7.2 单点登录规范

单点登录通过 CA 证书认证方式实现。

国家食品药品监管信息化工程建设基于 CA 认证体系，所有需要用户身份认证的应用系统应基于国家食品药品监管信息化工程 CA 认证体系进行用户身份认证。

应用支撑平台应提供安全认证网关实现单点登录。用户单点登录成功后，可使用统一的用户证书访问不同应用，不需要多次登录。

单点登录对应用系统建设有以下要求：

- a) 各应用系统的用户信息应与安全认证网关使用的 LDAP 用户信息同步，且用户关键标识一致。
- b) 各应用系统根据安全认证网关的认证 API 获取用户信息。

7.3 统一用户管理规范

对于已建设应用系统，可根据实际情况设计适宜的方式实现统一用户管理。

对于新建应用系统，应用支撑平台应提供统一用户管理功能，各应用系统用户的管理和维护由应用支撑平台统一实现，包括组织机构管理、用户管理、系统账号管理等功能。各应用系统在开发时，可以通过 API 调用和同步数据表的方式获取用户信息，各应用系统的用户信息和机构信息需要与应用支撑平台保持一致。设计思路如图 3 所示：

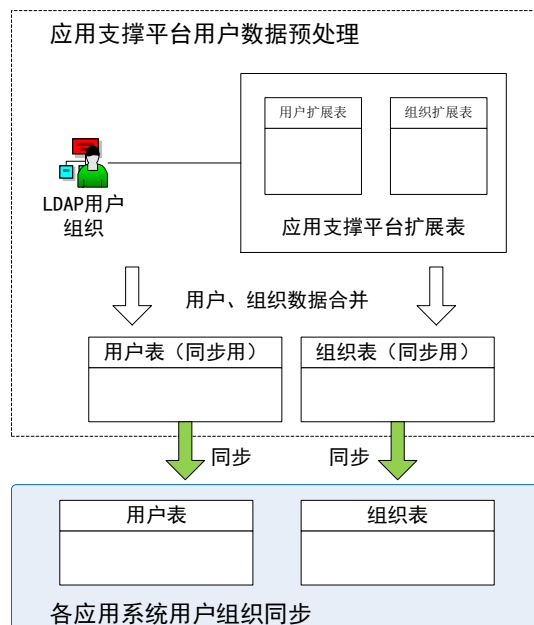


图 3：统一用户管理规范设计思路图

用户和组织数据由应用支撑平台先做预处理，各应用系统与预处理后的用户表和组织表做信息的同步。

7.4 权限管理规范

7.4.1 权限基本要求

应用支撑平台应提供统一权限管理功能，负责系统级别的权限控制，即用户能够访问哪个系统，应用系统内部的权限控制，由应用系统独立完成。如图 4 所示：

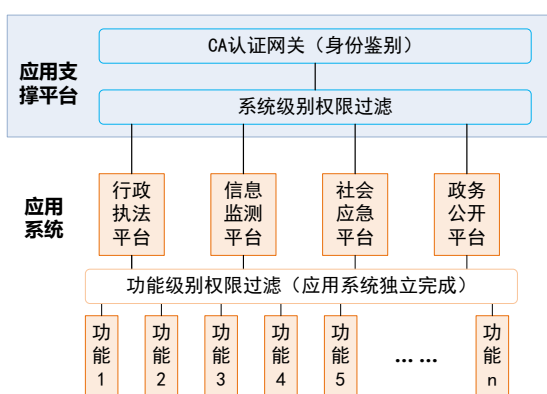


图 4：权限管理功能基本要求图

7.4.2 系统访问权限

应用支撑平台实现用户的系统访问权限，各应用系统开发时如需要判断用户是否具有指定系统的权限，可调用 API 获取。系统访问权限的 API 包括：

- 根据用户登录名称获取该用户所具有访问权限的应用列表。
- 根据用户登录名称和应用标识，验证该用户是否具有此应用的访问权限。

7.4.3 功能访问权限

功能访问权限由各应用系统根据本系统的需求进行控制，建议包括以下权限控制：

a) 功能权限控制

基于 ACL、RBAC 等多种方式实现权限控制，权限控制粒度应到每一个功能点，基于角色可自由分配权限。

b) 菜单权限控制

基于角色、菜单实现用户菜单级的授权管理，实现菜单权限可自由分配。

c) 数据权限控制

根据业务需求，对数据进行分类，做到数据权限的控制。

8. 应用支撑安全规范

8.1 概述

根据建设需要，国家食品药品监管信息化工程首先进行信息系统安全等级保护定级工作，然后根据所定级别进行应用系统安全相关的开发，应用系统信息安全开发应根据需求确定功能，并抽象出安全相关的基础功能在应用支撑平台中建设，且至少包含行为审计功能。

行为审计功能包括应用系统日志记录接口、系统管理记录接口、用户行为记录接口，各应用系统应按要求进行行为审计的记录。

8.2 应用系统日志记录接口

应用系统日志记录用户登录应用系统的信息，日志记录内容包括：

a) 应用门户首页调用 API 记录操作日志。

b) 各子系统日志。

应用系统日志涉及的日志类包括：

a) LogFactory：提供了获得 APPLog 实例的方法。

b) APPLog：提供了写入系统日志的方法。

c) LogConfigurationException：日志配置异常类。

日志写入的方法为：APPLog.log()。

其中 log 方法的参数包括：

——String actorIp：客户端 IP 地址；

——String actorId：登录用户 ID（可选）；

——String actorName：登录用户中文名称（可选）；

——String actorAccount：登录用户账号；

——String actType：行为分类，包括：用户行为、系统行为；

——String actAction：动作名称，包括：登录、退出；

- String actObjType: 操作对象分类, 此处指应用系统名称;
- String actObj: 操作对象的名称 (可选);
- String actResult: 操作结果, 包括: 成功、失败;
- String actLevel: 操作级别, 登录成功级别为: 1; 登录失败级别为: 10; 退出级别为: 1;
- String memo: 行为记录的描述 (可选)。

8.3 系统管理记录接口

系统管理日志记录管理员的操作, 应用系统中菜单注册、角色维护、角色授权、人员授权等操作均记录系统管理日志中。

应用支撑平台提供日志记录 API, 各应用系统在开发管理功能时, 应调用 API 记录管理操作日志:

- a) LogFactory: 提供了获得 SYSLog 实例的方法。
- b) SYSLog: 提供了写入系统日志的方法。
- c) LogConfigurationException: 日志配置异常类。

日志写入的方法为: SYSLog.log()。

其中 log 方法的参数包括:

- String actorIp: 客户端 IP 地址;
- String actType: 行为分类, 包括: 用户行为、系统行为;
- String actAction: 动作名称, 包括: 增加、删除、修改、启动、停止等等操作, 由应用系统自定义;

- String actObjType: 操作对象分类, 此处指应用系统名称;
- String actObj: 操作对象的名称 (可选);
- String actResult: 操作结果, 包括: 成功、失败;
- String actLevel: 操作级别, 操作级别分为 1-10 级, 越严重级别数字越高。

8.4 用户行为记录接口

应用系统开发时, 业务操作以及关键数据的查询等功能在开发时需要调用用户行为记录接口, 对数据的增加、删除、修改以及重要数据的查询进行记录。

应用支撑平台提供日志记录 API, 各应用系统在开发业务功能时, 应调用 API 记录用户行为操作日志:

- a) LogFactory: 提供了获得 USERLog 实例的方法。
- b) USERLog: 提供了写入系统日志的方法。
- c) LogConfigurationException: 日志配置异常类。

日志写入的方法为: USERLog.log()。

其中 log 方法的参数包括:

——String actorIp: 客户端 IP 地址;

——String actType: 行为分类, 包括: 用户行为、系统行为;

——String actAction: 动作名称, 包括: 增加、删除、修改、启动、停止等等操作, 由应用系统自定义;

——String actObjType: 操作对象分类, 此处指应用系统名称;

——String actObj: 操作对象的名称 (可选);

——String actResult: 操作结果, 包括: 成功、失败;

——String actLevel: 操作级别, 操作级别分为 1-10 级, 越严重级别数字越高。

9. 服务建设规范

9.1 服务建设基本要求

服务建设宜基于企业服务总线 (ESB) 进行, 如果项目中无企业服务总线, 应按照服务建设规范中的服务设计规范、服务封装规范、服务开发规范进行服务的建设, 并实现服务的统一管理。如果项目中存在企业服务总线, 同时应遵循服务集成规范。

9.2 服务设计规范

9.2.1 概述

Web 服务作为应用系统之间的交互方式, 应按照统一的标准进行开发, 服务的设计包括服务识别与服务定义。

9.2.2 服务识别

a) 服务识别是指对业务进行分析和梳理, 抽象出业务实现所需的服务, 并按服务分类要求进行合理划分。

b) 服务识别应分析与业务功能或业务数据相关的接口以及约束该接口的契约, 接口和契约采用中立、基于标准的方式进行定义, 它独立于实现服务的硬件平台、操作系统和编程语言。

c) 服务识别应从业务的角度出发, 包括但不限于下述几点:

1) 业务流程切入点: 通过梳理、优化业务流程, 将业务流程转化为可重用、更具有灵活性的流程服务。

2) 用户体验切入点: 关注用户体验需求, 为终端用户提供增值、个性化、多渠道的服务, 并据此来优化整合内部的应用和流程。

9.2.3 服务定义

服务定义是在服务识别的基础上定义服务的各项属性, 描述服务的信息。

服务的属性包括: 基本属性、技术属性、安全属性、配置属性, 服务的各项属性定义应分阶段进行、逐步细化。服务识别阶段定义服务的基本属性, 服务设计阶段定义服务的技术属性与安全属性, 服务的

部署阶段定义服务的配置属性。

a) 服务的基本属性包括但不限于表 1 所述信息：

表 1

序号	属性	说明	取值说明
1	服务编码	标识服务的唯一编码	
2	服务英文名称	服务的英文概要名称，描述应简洁准确	
3	服务中文名称	服务的中文概要名称，描述应简洁准确	
4	服务性质编码	描述服务的特性	如：关键任务服务、高容量服务、高水平服务、标准服务
5	服务功能描述	对服务功能规格的详细描述。	
6	服务开发单位	实现服务的开发厂家	如：中科软科技股份有限公司

b) 服务的技术属性包括但不限于表 2 所述信息：

表 2

序号	属性	说明	取值说明
1	版本号	服务的版本号	如：V1.0
2	注册时间	服务的正式注册时间	如：2013-01-01 10:00
3	依赖的服务	本服务需要调用的其它服务的编号列表	如：服务编码 1；服务编码 2；
4	实现方式	具体技术实现方式	如：JAVA
5	服务类型	属于服务体系中的哪种类型	如：访问服务、数据服务、业务服务、流程服务、综合服务、展现服务
6	交互属性	是否需要人工交互	是/否
7	服务调用方式	客户端调用服务的具体方式	如：服务同步调用、服务异步无返回调用、服务异步有返回调用
8	接口方法	服务提供的接口方法列表	如：Create()
9	接口协议	调用服务的通讯协议	如：http
10	服务启用时间	服务的正式启动时间	2013-01-10 8:00
11	服务停用时间	服务的正式停用时间	2013-12-31 18:00

c) 服务的安全属性包括但不限于表 3 所述信息：

表 3

序号	属性	说明	取值说明
1	安全要求	调用服务时，是否需要进行安全认证	是/否
2	允许调用的角色	允许调用该服务的角色列表	如：Operator；Manager
3	服务自行安全认证	服务被调用时，是否还进行自身的安全认证	是/否

d) 服务的配置属性包括但不限于表 4 所述信息：

表 4

序号	属性	说明	取值说明
1	服务部署 IP 地址	提供服务功能的网络 IP 地址	如：127.0.0.1
2	服务接口定义文件	描述服务接口定义的文件路径	如：http://192.168.1.1/test.wsdl
3	可以使用的时	可以使用该服务的时间段	如：0：00—24：00
4	是否支持重试	服务调用失败后，是否支持重发调用	是/否

9.3 服务封装规范

9.3.1 服务封装原则

a) 服务封装是 Web 服务实现的手段，服务封装将应用系统可重用的功能或数据“剥离”出来，对外以相关的接口方式以及约束这个接口的契约提供给消费者调用。接口和契约的定义是中立的且基于标准的，并独立于实现服务的硬件平台、操作系统与编程语言。

b) 服务封装应遵循包括但不限于下述原则：

1) 无状态原则：最大限度减少服务管理的状态信息以及状态期限。

2) 单一实例原则：避免功能冗余。

3) 接口定义原则：使用 WSDL 定义服务接口，使用 WS-Policy 描述服务契约，使用 XML 模式(Schema) 定义服务交换的消息格式（即服务的公共数据）。服务消费者依赖服务契约调用服务，服务定义应相对稳定，修改应通过审核批准。

4) 自包含和模块化原则：服务封装的是那些在业务上稳定的、重复出现的活动和组件，组成服务的功能实体是完全独立自主的，可以独立进行部署、版本控制、自管理和恢复。

5) 粗粒度原则：服务粒度指抽象级别或者服务包含的功能。确定服务粒度时需要考虑性能需求，

以及未来可能进行的更改对服务实现的影响，应尽可能使用粗粒度模式隐藏其中的细粒度服务，这样有利于将服务与其实现的更改隔离开来。服务数量太多会带来服务管理的复杂性。

6) 松耦合性原则：服务消费者使用服务接口调用服务，服务的位置、实现技术、当前状态以及服务的私有数据对服务消费者是透明的。

7) 可重用原则：服务是可重复使用的。

8) 策略声明原则：应利用策略声明描述对服务的期望，例如：安全性方面的要求、与业务有关的语义方面的要求以及服务级别方面的要求等。

9.3.2 服务封装方式

a) 对于技术实现方式和接口不能满足或难以满足本规范服务定义与服务封装要求的现有应用系统，建议通过适配器对现有应用系统进行集成，利用适配器对外提供的各类接口的方式实现服务封装。

b) 对于采用 J2EE 支持 Web 服务开发的现有应用系统，可以在不改变现有应用系统的技术实现方式与现有接口的前提下，通过增加对外接口以支持标准的服务接口协议的方式，直接实现现有应用系统功能模块的服务封装。

c) 对于新建的应用系统，应按本规范的服务定义与服务封装的要求，直接采用支持 SOA 的开发工具进行服务封装。

9.4 服务开发规范

9.4.1 基本要求

国家食品药品监管信息化工程 Web Service 服务要求统一基于 AXIS2 进行开发和部署，并在企业服务总线中注册。

9.4.2 Web Service 命名规范

9.4.2.1 服务命名

Web Service 服务名称应遵循以下要求：

a) 服务的方法名称为大小写混合的形式，以小写字母开头，名字中其他单词的首字母以大写字母开头，所有其它的单词都应小写字母，不宜使用下划线分隔单词。服务的命名应该能描绘出方法的作用和功能，服务的名字宜使用祈使动词或者动词短语。

b) 服务的方法参数命名应用一个小写字母开头，后面的单词应用大写字母开头。

c) 所有的服务方法应有完整的注释，注释内容包含方法功能介绍、参数说明、返回类型说明、例外类型，例如：

示例：

```
/**
```

```
* 方法的功能描述。
```

<pre>* @param name 名称 * @return 返回值类型 (INT); 返回-1 表示传入的参数存在类型错误或参数不足, * 返回 0 表示操作失败, 返回 1 表示操作成功。 */</pre>
解释: @param 格式如下: @param 参数名 说明 其中,“参数名”是指参数列表内的标识符,而“说明”代表一些可延续到后续行内的说明文字。一旦遇到一个新文档标记,就认为前一个说明结束。可使用任意数量的说明,每个参数一个。 @return 格式如下: @return 说明 其中,“说明”是指返回值的含义。它可延续到后面的行内。

9.4.2.2 targetNamespace 命名要求

targetNamespace 在国家食品药品监管信息化工程中的命名为: www.cfda.gov.cn。

9.4.3 WSDL 编写说明

9.4.3.1 编写要求

WSDL 文档是利用这些主要的元素来描述某个 web service 的, 如表 5 所示:

表 5

元素	定义
<portType>	web service 执行的操作
<message>	web service 使用的消息
<types>	web service 使用的数据类型
<binding>	web service 使用的通信协议

WSDL 文档可包含其它的元素, 比如 extension 元素, 以及一个 service 元素, 此元素可把若干个 web services 的定义组合在一个单一的 WSDL 文档中。

a) WSDL 端口

<portType>元素是最重要的 WSDL 元素, 它可描述一个 web service、可被执行的操作, 以及相关的消息, 可以把<portType>元素比作传统编程语言中的一个函数库 (或一个模块、或一个类)。

b) WSDL 消息

<message>元素定义一个操作的数据元素。

每个消息均由一个或多个部件组成。可以把这些部件比作传统编程语言中一个函数调用的参数。

c) WSDL types

<types>元素定义 web service 使用的数据类型。

为了最大程度的平台中立性, WSDL 使用 XML Schema 语法来定义数据类型。

d) WSDL Bindings

<binding>元素为每个端口定义消息格式和协议细节。

9.4.3.2 WSDL 实例

以下是 WSDL 文档的简化片段的举例：

示例：

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://www.cfda.gov.cn/SimpleService/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="SimpleService"
targetNamespace="http://www.cfda.gov.cn/SimpleService/">
  <wsdl:types>
    <xsd:schema targetNamespace="http://www.cfda.gov.cn/SimpleService/">
      <xsd:element name="concatRequest">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="s1" type="xsd:string" />
            <xsd:element name="s2" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="concatResponse" type="xsd:string">
      </xsd:element>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="concatRequest">
    <wsdl:part element="tns:concatRequest" name="parameters"/>
  </wsdl:message>
  <wsdl:message name="concatResponse">
    <wsdl:part element="tns:concatResponse" name="parameters"/>
  </wsdl:message>
  <wsdl:portType name="SimpleService">
    <wsdl:operation name="concat">
      <wsdl:input message="tns:concatRequest"/>
      <wsdl:output message="tns:concatResponse"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="SimpleServiceSOAP" type="tns:SimpleService">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="concat">
      <soap:operation soapAction="http://www.cfda.gov.cn/SimpleService/concat"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="SimpleService">
    <wsdl:port binding="tns:SimpleServiceSOAP" name="SimpleServiceSOAP">
      <soap:address location="http://IP:PORT/axis2/services/SimpleService"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

9.4.4 适配器开发规范

基于 TCP/IP 协议，国家食品药品监管信息化工程可以使用的适配器类型包括：

a) Socket 通讯方式适配器。

b) Http 通讯方式适配器。

c) JMS (Java Message Service: JAVA 企业级消息规范) 通讯方式适配器。

适配器分为长连接和短连接, 以及同步通讯方式和异步通讯方式。

通过适配器可以进行:

a) 定义 Web Service 服务: 定义 Web Service 的方法、参数等信息。

b) 转换 Web Service 的输入参数: 把节点的输入参数转换为 Web Service 的方法参数。

c) 调用 Web Service 服务: 调用 Web Service 服务, 并输入 Web Service 服务对应的参数。

d) 打包 Web Service 的返回结果: 把 Web Service 运行的结果打包返回。

9.5 服务集成规范

9.5.1 企业服务总线 (ESB)

a) ESB 提供了一种开放的, 基于标准的消息机制, 通过标准的适配器和接口, 来完成服务之间的互操作。ESB 提供了多协议的服务调用接入、服务路由、服务访问控制和服务适配器等核心功能。ESB 是服务提供者和服务消费者之间的一个中介, 避免了点对点的集成, 是实现 Web 服务松耦合的重要机制。服务消费者可以不关心服务提供者的接口 (消息格式的不同)、地域 (服务部署位置的变化)、调用方式 (同步/异步)、传输协议 (服务提供者和消费者可以使用不同的通讯协议)、技术实现 (编程语言/部署环境) 等。ESB 功能包括但不限于:

1) 服务接入: 服务接入是调用服务的统一入口。包括: 接收服务请求消息和调用者使用的通信协议与 ESB 内部通信协议之间转换功能。

2) 访问控制: 访问控制包括身份鉴别与权限控制。

3) 消息转换: 消息转换提供不同格式的消息之间的转换, 包括输入消息转换和输出消息转换。

4) 服务路由: 根据消息的内容和预先配置好的规则, 将服务请求传递给某个具体的服务处理。

5) 适配转换: 负责 ESB 内部通信协议与被调用的服务使用的通信协议之间的转换, 并调用服务, 获得服务返回结果。

b) ESB 应包括但不限于下述特性:

1) 与操作系统和编程语言无关, 并能在 Java 和 .Net 应用程序之间工作。

2) 使用 XML 作为标准通信语言。

3) 支持 Web 服务标准。

4) 支持多种传输协议, 例如: HTTP(S)、JMS、RMI、FTP 或其他消息中间件。

5) 支持多种消息传递方式, 包括: 同步、异步、点对点、发布-订阅等。

6) 包含基于标准的适配器, 例如: JCA、文件适配器、JDBC 适配器等。

7) 包含基于内容的智能路由功能。

8) 包含标准安全模型, 用于 ESB 的认证、授权和审计。

9) 包含消息转换功能, 使用可视化映射工具定义 XSLT 规则, 在发送应用和接收应用之间能够进行

格式转换、语义转换。

10) 包含基于模式 (schema) 的消息验证。

11) 支持服务管理, 比如服务的注册、维护、报废和版本管理; 监控服务的运行情况, 包括时延、吞吐量、错误率等。

12) 提供对中文的全面支持, 包括对 GB2312/GBK/UTF8 的完善支持。

c) ESB 应具有良好的可扩展性, 能够充分利用硬件系统的资源, 支持垂直扩展和水平扩展, 支持负载均衡, 满足日益增长的服务数量的需求。

9.5.2 服务描述

a) 服务使用 WSDL 描述其使用的抽象消息操作、具体的网络协议和端点地址。

b) 服务使用 XML 模式 (XML Schema) 描述其接收和发送的基于 XML 的消息的结构和内容。

c) 服务使用 Web 服务策略 (Web Services Policy) 规范来描述 Web 服务的能力、需求和一般特征, 包括但不限于安全性策略。

9.5.3 服务注册/发布

a) 基于 SOA 松耦合特性, 需要对各种服务进行注册, 以方便服务提供者发布自己的服务、服务消费查找所需的服务。

b) 服务注册中心需要提供分类管理能力, 实现对服务的搜索。

c) 服务注册应该提供服务审批的功能, 保证对敏感注册数据的任何变动都能够传递到适合的审批流程中。同时, 还应该提供服务变更管理功能, 支持变更的通知和订阅, 能够实现将注册数据的变动主动地通知到管理员或者相应的流程。

9.5.4 服务发现/调用

a) ESB 应具有自动发现服务的功能, ESB 中可以通过图形化界面及参数配置的方式调用服务。

b) Java 客户端应用程序中使用 UDDI API, 以编程方式查找服务目录, 并调用服务。

c) SOA 服务应优先采用 Web 服务方式实现, 并符合 WS-I 国际标准。

d) 服务间互操作的协议为简单对象访问协议 (SOAP)。

e) 服务间数据交换的格式为可扩展标记语言 (XML)。

9.5.5 服务编排

a) 复杂的服务需要通过若干个简单的服务组合而成, 这时候就需要对服务进行编排。

b) 轻量级服务编排在 ESB 中完成, 通过 ESB 的服务路由、消息格式转换功能, 实现多个服务组合成一个更粗粒度的服务。

c) 需要长期运行的业务流程, 可利用业务规则管理组件进行服务编排。