ICS 35.240.40
CCS A 11

团　体　标　准

T/BFIA 016—2022

商业银行应用程序接口共享平台技术要求

Technical requirements for sharing platform of commercial bank application programming interface

2022－08－16 发布　　2022－08－16 实施

北京金融科技产业联盟　发布

版权保护文件

目　　次

前　言

本文件按照GB/T 1.1—2020《标准化工作导则　第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由北京金融科技产业联盟归口。

本文件起草单位：神州数码信息服务股份有限公司、北京金融科技产业联盟、中国人民银行清算总中心、中国银联股份有限公司、交通银行股份有限公司、招商银行股份有限公司、广发银行股份有限公司、中电金信软件有限公司、赞同科技股份有限公司、杭州趣链科技有限公司、恒生电子股份有限公司、武汉达梦数据库股份有限公司、广州顺德农村商业银行股份有限公司、无锡锡商银行股份有限公司、平安银行股份有限公司、杭州云链趣链数字科技有限公司、成都虚谷伟业科技有限公司、北京东方通科技股份有限公司、秦皇岛银行股份有限公司、兰州银行股份有限公司、珠海华润银行股份有限公司。

本文件主要起草人：李敏、史博文、杨宗智、马洪杰、于宏志、沈伟、施媛、潘紫娟、聂丽琴、杨毅夫、王艺颖、汤洋、向洁敏、方鹤鸣、张美庆、刘慕寒、连宾雄、吴守钰、魏晓云、闵勇博、张璐、严佳栋、黄海明、张寒、陈金元、付晓东、卢锐、洪旭东、刘洁、姜勇、明玉琢、黄元霞、林静、项海南、李建庆、王凌云、许泽敏。

商业银行应用程序接口共享平台技术要求

1 范围

本文件规定了商业银行应用程序接口共享平台逻辑结构、功能架构、技术要求和安全要求。

本文件适用于商业银行应用程序接口共享平台的建设及应用。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件，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。

GB 17859—1999 计算机信息系统安全等级保护划分准则

GB/T 22239—2019 信息安全技术 网络安全等级保护基本要求

GB/T 35273—2020 信息安全技术 个人信息安全规范

JR/T 0071.2—2020 金融行业网络安全等级保护实施指引 第2部分：基本要求

JR/T 0171—2020 个人金融信息保护技术规范

JR/T 0185—2020 商业银行应用程序接口安全管理规范

3 术语和定义

下列术语和定义适用于本文件。

3.1

应用程序接口共享平台 sharing platform of commercial bank application programming interface

通过应用程序接口调用的方式，实现后台功能的共享的应用系统。

3.2

网络安全 cybersecurity

通过采取必要措施，防范对网络的攻击、侵入、干扰、破坏和非法使用以及意外事故，使网络处于稳定可靠运行的状态，以及保障网络数据的完整性、保密性、可用性的能力。

[来源：GB/T 22239—2019,3.1]

3.3

开发者 developer

应用程序接口共享平台的使用用户，即接口使用方，包括但不限于企业、商户、第三方合作伙伴，使用该平台对外发布的应用程序接口进行自身应用的开发。

3.4

用户 user

开发者的应用的使用者。

3.5

应用程序接口 application programming interface

一组预先定义好的功能，开发者可通过该功能（或功能的组合）便捷地访问相关服务，而无需关注服务的设计与实现。

[来源：JR/T 0185—2020,3.1]

3.6

应用软件开发工具包 software development kit

基于特定的软件包、软件框架、硬件平台、操作系统等建立应用时所使用的软件开发工具集合。

[来源：JR/T 0185—2020,3.5]

3.7

移动客户端应用软件 mobile client application software

在移动终端上为用户提供服务的应用软件，包括但不限于可执行文件、组件等。

3.8

安全套接层协议 secure sockets layer

一种处于网络层与应用层之间，提供客户端和服务器的鉴别及保密性和完整性服务的协议。

3.9

安全传输层协议 transport layer security

用于在两个通信应用程序之间提供保密性和数据完整性。

3.10

每秒查询率 query per second

对一个特定的查询服务器在规定时间内所处理流量多少的衡量标准，在因特网上，作为域名系统服务器的机器的性能经常用每秒查询率来衡量。

4 缩略语

下列缩略语适用于本文件。

API：应用程序接口（Application Programming Interface）

App：应用（Application）

DMZ：（网络）隔离区（Demilitarized Zone）

HTTPS：超文本传输安全协议(Hypertext Transfer Protocol Secure)

H5：超文本标记语言5.0（Hyper Text Markup Language 5.0）

QPS：每秒查询率（Query Per Second）

SDK：应用软件开发工具包（Software Development Kit）

SSL：安全套接层协议（Secure Sockets Security）

SFTP：安全文件传送协议（Secret File Transfer Protocol）
URL：统一资源定位符（Uniform Resource Locator）

5 概述

商业银行应用程序接口共享平台是商业银行将自身的服务通过应用程序接口对接的共享方式，提供给开发者（包括但不限于企业、商户及第三方合作伙伴）而搭建的应用平台。开发者可以使用商业银行的API或者SDK等方式进行对接，其逻辑结构见图1。

——商业银行应用程序接口共享平台的参与方主要包括开发者、商业银行。商业银行通过API直接连接或者SDK间接连接方式向开发者提供共享服务。商业银行对外提供的API和SDK接口规范应遵循JR/T 0185—2020要求。

——开发者向商业银行应用程序接口共享平台发送相关请求，接收返回结果。

——商业银行提供应用程序接口，通过商业银行应用程序接口共享平台实现对接，将请求转发至银行业务系统，将结果返回。商业银行应用程序接口共享平台重点实现对接过程中的基本功能，包含安全认证、流量控制、故障隔离、报文转换、服务组合等功能。

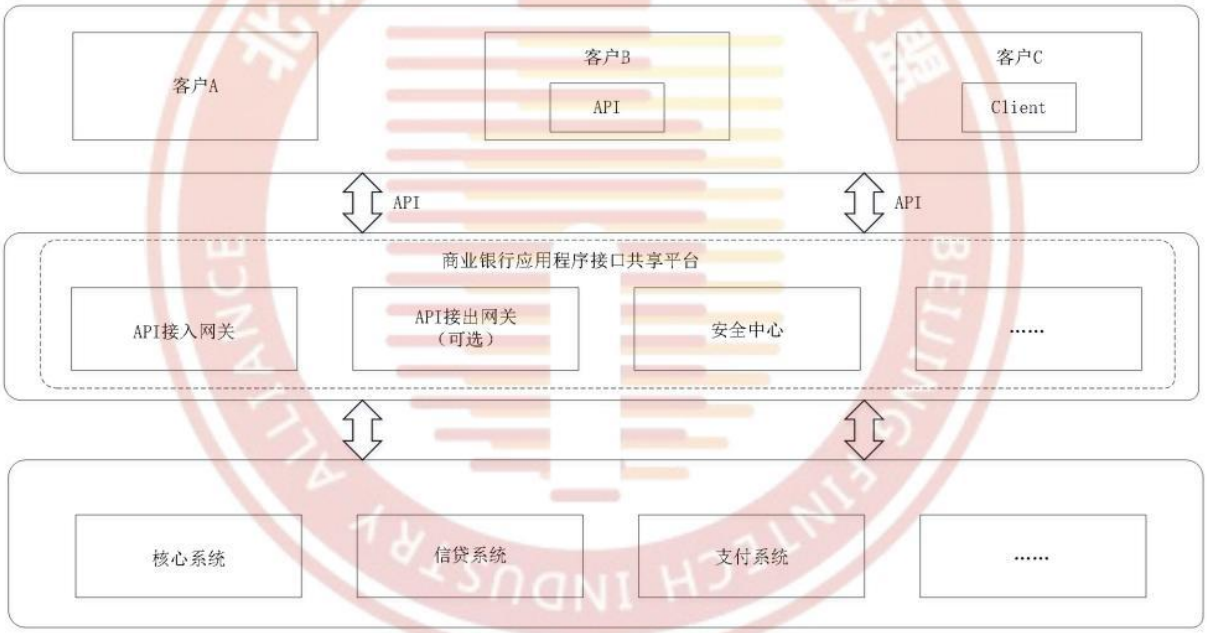

图 1　商业银行应用程序接口共享平台逻辑结构

6 功能架构

6.1 整体架构

商业银行应用程序接口共享平台功能整体上分为：开发者门户、安全中心、接入网关、接出网关、文件网关、安全管理、监控管理、参数中心，为更好管理共享平台服务，还宜包括服务治理、服务组合功能模块，系统功能架构见图 2。

图 2　系统功能架构

6.2　API 接入网关

API接入网关是商业银行应用程序接口共享平台中核心和基础的运行态功能实现平台。主要包含服务发现、安全控制、服务路由，根据服务使用方情况，还宜包括协议转换、报文转换、动态发布等技术集成功能，为API的调度提供安全、稳定、可靠的运行环境。

API接入网关的核心控制就是安全接入，通过用户、应用、系统和服务多维度组合完成系统的安全和访问权限多维度控制，接入网关要求如下：

a）　应具备服务熔断与降级的功能；

b）　交易路由策略应对于不同业务场景分配不同的路由策略，保证网关的高扩展性；

c）　动态发布的能力，使得更新和发布不会影响网关的基本功能，应保证网关的高可用性；

d）　API接入网关的服务适配宜支持服务路由和报文转换的功能，其中报文转换应该支持动态加载的功能，便于后台服务系统的服务发布和变更；

e）　API接入网关宜支持大量API迁移、流量切换功能。

6.3　API 接出网关

API接出网关为银行内部业务系统提供安全的外联接出服务，实现第三方服务的整合和第三方系统差异的屏蔽，包括不限于异步交易的结果通知、外部数据查询等业务场景。

API接出网关应具备API接入网关的基本功能，包括服务发布、服务路由、安全控制、为更好的屏蔽第三方系统的差异，还宜包括协议转换、报文转换等。

6.4　文件网关

文件网关用于满足商业银行与合作方进行文件传输需求，应提供安全可控、稳定可靠的文件传输服务，主要包括文件上传、下载、文件通知、传输校验、文件加解密、身份认证、访问控制、流量控制、分区／目录隔离、用户管理、文件清理、文件日志审计等功能，为保证共享平台更稳定、可靠、安全运行，系统还宜具备文件传输流量控制、文件病毒扫描等系统安全保护措施。

商业银行跟外部第三方系统间的文件交互宜采用HTTPS、SFTP等安全通讯协议，HTTPS协议版本宜使用TLS 1.2及以上版本，并持续及时更新到安全稳定版，取消对存在重大安全隐患版本的协议支持。若

选用SFTP协议，宜以客户端身份接入第三方系统。文件传输校验内容包括不限于文件大小、传输时间、文件后缀名、黑白名单、完整性等。

6.5 API 安全中心

API 安全中心应为服务的发布、开放、运行提供一整套安全解决方案。包括安全管理控制台功能、SDK 安全控制功能、开放授权（OAuth）2.0 功能、限流管理功能、安全数据服务功能。

a) 安全管理控制台功能应包括：
   1) 证书管理，提供对合作方数字证书的管理，证书签发宜通过可信 CA 签发；
   2) 密钥管理，密钥的生成、上传、重置；
   3) 开发者权限管理，提供开发者的权限控制管理。

b) SDK 安全控制功能应包括：
   1) SDK 授权认证，提供 SDK 的授权认证；
   2) SDK 信息管理，提供 SDK 签名、版本管理。

c) OAuth2.0 功能应包括：
   1) 授权码，提供授权码的获取和超时管理；
   2) 令牌（token）管理，提供 token 的生成、存储、更新管理；
   3) 授权，提供基于 token 的授权管理；
   4) 认证，提供基于 token 的认证管理。

d) 限流控制功能应包括：
   1) 维度控制，提供多种维度的限流控制；
   2) 限流算法，宜具备多种限流算法，比如令牌桶算法等；
   3) 统计分析，提供在线流量实时统计分析功能；
   4) 参数调整，提供在线限流控制参数调整并实时生效。

e) 安全数据服务功能应包括：
   1) 国密算法加解密，提供基于国密算法的对称算法 SM4、非对称算法 SM2 的加密解密；
   2) 国际算法加解密，提供基于国际算法的对称算法 AES、非对称算法 RSA 的加密解密；
   3) 数据加签验签，提供基于 RSA 算法、SM2 算法的加签验签；
   4) 数据脱敏，提供敏感数据的脱敏传输；
   5) 数据还原，提供敏感数据的脱敏还原。

6.6 参数中心

6.6.1 参数中心能力要求

共享平台系统建设可遵循微服务架构、分布式系统特性，为保障系统高可用、扩展性、先进性，还宜具备参数中心，参数中心宜具备以下能力：

a) 可以动态地对共享平台的服务运行参数配置，便于共享平台运营网关等模块在线扩容；
b) 确保参数中心的高可用保障，避免由单点故障导致业务中断；
c) 参数配置是可以多个服务共享的；
d) 支持权限管理，只有被授予权限的用户才能查看和修改参数配置；
e) 参数配置可以回滚，当遇到配置出现问题的时候可以像回滚服务一样回滚配置；
f) 支持共享平台服务发布灰度发布，发布服务可以部分发布更新，试运行正常后，全量发布；
g) 参数中心自身性能能够支撑业务系统所需的 QPS，在微服务架构下，能为多达几十至上百台服务器能提供稳定的服务支撑。

6.6.2 参数中心基本功能要求

参数中心应具有以下基本功能：

a) 命名空间：和注册中心一样，命名空间属于顶层的结构，用于进行租户级别的隔离，最常用的就是不同环境按照不同命名空间分开，比如测试环境和线上环境进行隔离；
b) 参数配置管理：系统参数配置的编辑、存储、分发、变更管理、历史版本管理、变更审计等所有与配置相关的活动；
c) 配置项：一个具体的可配置的参数与其值域，通常以“param-key=param-value”的形式存在；
d) 配置集：一组相关或者不相关的配置项的集合称为配置集，在系统中，一个配置文件通常就是一个配置集，包含了系统各个方面的配置（例如，一个配置集可能包含了数据源、线程池、日志级别等配置项）；
e) 配置集 ID：某个配置集的 ID。配置集 ID 是组织划分配置的维度之一；
f) 配置分组：参数中心中的一组配置集，是组织配置的维度之一；
g) 配置快照：参数中心的客户端 SDK（如 jar 组件）会在本地生成配置的快照。当客户端无法连接到参数中心时，可以使用配置快照显示系统的整体容灾能力。

6.7 API 治理平台

API 治理平台提供对外发布 API 服务资产的管理。API 治理平台应包括 API 的定义、发布、审批，对 API 生命周期的管理，对元数据的管理，对服务配置的管理和控制，宜具备 API 自动化导入导出功能，将整合后的 API 下发到 API 网关运行态，使得服务的治理、导入、下发一体化、自动化，支持与开发者门户联动，API 治理正式发布的服务可以通过开发者门户进行查询。

6.8 API 监控平台

API 监控平台提供对应用系统的资源、应用服务器、数据库服务器、系统运行、服务和产品运行、业务系统、开发者的第三方应用进行监控，并提供报表查询和下载。API 监控平台监控和提供的内容应包括：

a) 提供系统资源运行情况；
b) 提供服务运行监控功能情况；
c) 提供各时段服务运行情况的统计报表；
d) 提供多维度服务查询，支持以流水号、时间段、时间点、交易状态、交易返回码等作为查询条件进行查询；
e) 支持图形、邮件、短信的告警功能，并支持告警阈值的自定义设置；
f) 如银行已建立统一的集中告警平台，则 API 监控平台应具有推送监控明细数据及告警数据到第三方集中监控平台的能力。

6.9 API 组合

为了进一步整合银行内部资源、减少银行内外的交互次数、提高交互效率，商业银行应用程序接口共享平台宜对原始服务（接口）进行一定程度的组合。宜支持多种 API 的组合方式，典型的类型如下：

a) 串行组合：原始服务（接口）顺序执行；
b) 并行组合：原始服务（接口）并发执行；
c) 服务分发：原始服务（接口）根据条件选择执行，多个原始服务（接口）选择其中一个执行；
d) 条件组合：原始服务（接口）根据条件选择执行，多个原始服务（接口）可能执行、不执行、执行一部分；

e) 并行转串行：两个（或多个）的原始服务（接口）存在依赖关系，在不同时参与组合时，可以与其他原始服务（接口）并行执行；但是这两个（或多个）原始服务（接口），同时参与组合时，顺序执行；
f) 业务数据查询补填：原始服务（接口）的执行需要依赖于其他查询类原始服务（接口）的结果；
g) 批量转单笔多次：存在依赖关系的原始服务（接口）之间，一个是多条数据一次发送，一个一次仅能处理单条数据；
h) 自动冲正：原始服务（接口）应搭配相应的冲正服务，有组合服务在异常时调用，保障数据一致性；
i) 存储转发，定时调用：需要存储相关数据，定时调用原始服务（接口）。

6.10 工具集

商业银行应用程序接口共享平台在建设过程中，提供配套工具集，支持系统快速开发，快速集成的管理工具集。工具集宜包括下列内容：

a) 后台服务挡板；
b) 配置三方挡板；
c) 发送报文模拟器，提供在线测试的发送报文模拟器；
d) 回归测试工具，在准备好测试报文的情况，通过工具进行自动化测试。支持单个服务和多个服务的自动化测试；
e) 自动化部署工具，提供在线版本发布与部署功能，保证在集群部署的架构模式下，简化版本部署流程，提升版本部署效率；
f) 沙箱环境，提供第三方开发者接入前的联调测试，沙箱环境测试通过后接入正式生产环境。

6.11 高速共享缓存

为保证服务的更新频次、产品的灵活性以及零信任的安全要求，必要的验证数据需具备随机性和临时性，平台宜具备高速共享缓存，功能包括：

a) 集群或者分布式的部署方式，避免单点故障后，导致服务停止；
b) 即便缓存中的数据丢失，仍需保证系统依然能正常对用户提供服务；
c) 高速缓存需要具备相应的内存、状态等的监控告警。

6.12 数据库

商业银行应用程序接口共享平台的数据持久化要求包括：

a) 支持的数据库，应包括分布式、读写分离、共享集群等高可用架构中的一种或者多种；
b) 面向互联网的服务宜支持读写分离功能，宜支持对流水表进行分表或分库操作；
c) 面向互联网提供入口服务功能的应用，宜支持脱离数据库提供服务。

6.13 开发者门户

开发者门户应包括门户首页、应用中心、文档及 API 中心、下载中心、帮助服务中心等功能模块，实现开发者注册、认证、应用服务申请和产品、服务以及相关文档的在线查看和发布等功能，提供机构接入者身份，使其可以自管理以本机构名义接入的开发者，满足接入机构身份安全控制要求。

7 技术要求

7.1 API 架构风格

API 应遵循开放简洁的设计风格，体现出面向资源、面向服务的思想。出于安全性考虑，采用 HTTP 或 HTTPS 作为应用层协议，支持 GET、POST、HEAD（可选）、OPTIONS（可选）接口方法。一个 URI 由以下内容组成：

a) 协议，例如“http”或“https”；
b) 主机，例如 api.cloud.cn；
c) 端口号，当使用默认值时，可以不出现；
d) 一段或多段路径，例如“/user/1234”；
e) 查询字符串。

在 URI 的路径部分使用斜杠分隔符“/”来表示资源之间的层次关系，将 API 的版本号放入 URI 中，而不是放入 http 报文的 HEAD 部分中。宜支持驼峰、下划线和大小写，并且大小写敏感。

格式宜支持 XML、JSON、HTML，推荐使用 JSON。

对于那些由客户端输入所造成的错误，宜返回带 4xx 状态码的表述。对于那些由服务器实现造成的错误，宜返回带 5xx 状态码的表述。

示例： 使用 RESTFul 风格。

HTTPS URL：http://openapi.baidu.com/rest/2.0/passport/users/getLoggedInUser

请求：http://openapi.baidu.com/rest/2.0/passport/users/getLoggedInUser?access token=xxx&

应答：JSON 格式：

```
{
    "uid":2346677,
    "uname":"liupc24"
    "portrait":"e2c1776c31393837313031319605"
}
```

7.2 API 网关

API 网关主要包含服务发布、安全控制、服务路由等技术集成功能，应为 API 的调度提供安全、稳定、可靠的运行环境，在不同的服务提供方环境下，还宜具备协议转换、报文转换等功能。

API 网关应支持松耦合的连接架构，基于工业标准，并支持各种协议。

API 网关应支持服务化技术，包括 Restful 服务、Web 服务、代理服务，提供服务发布、注册、调用、转换、编排、监控等工具，提供安全防护的措施。

API 网关应具备通信管理能力，包括服务管理和调度、流量控制、负载均衡、版本管理、故障转移以及熔断功能、SSL 协议支持。

API 网关应支持 API 的全生命周期的管理，包括 API 创建、API 发布、API 订阅等。通过 API 管理设置 API 的访问控制信息，仅订阅该 API 的用户有权访问该 API。

7.3 SDK 集成技术

SDK 的设计主要基于 HTTPS 通讯协议，将通讯、安全、页面以及 API 等融为一体提供给开发者，降低轻开发者的开发难度，提高接入效率。

应用程序的集成方式分为服务端对服务端方式和移动终端对服务端集成两种方式，应提供对应方式的集成开发工具包，服务器端 SDK 支持常用的 Java、C#、Python 、PHP 等开发语言，客户端 SDK 应提供在不同操作系统和使用条件下的对应安装包，包括 Android-SDK、IOS-SDK、H5-SDK。

7.4 数据描述规则

平台应制定通俗易懂、统一规范的API、SDK、元数据命名规则，便于开发者理解使用相关资源。

7.5 版本管理技术

平台应采用灰度发布管理模式，提供API的多版本共存能力，如分支管理与合并控制，版本变更记录和追踪，版本前后兼容等，支撑特殊情况的平滑过渡。

7.6 沙箱环境

为开发者提供沙箱测试环境，提升开发者的使用体验，应保障正式环境的稳定可靠。

7.7 API 治理

API治理是基于元数据，进行API服务的定义、描述、映射、检索，以提升API服务资产管理为导向的工具，提供对现有以及将来新增服务进行整理、归纳、定义、组装。治理应按照所定义的规范进行约束，满足规范的治理称为标准化治理，反之为非标准化治理（即个性化要求），非标准API需通过适配手段转换为标准服务。

7.8 前后端分离技术及前端框架技术

面向开发者/运营人员的系统宜使用前后端分离技术。通过前后端分离设计，实现前端与后端的解耦，提高前端的灵活性和后端的稳定性。前端框架应采用HTML5标准实现的框架技术。

8 安全要求

8.1 基本要求

应满足以下要求：

a) 应用程序接口安全设计符合 JR/T 0185—2020 中 7.1 的要求；
b) 系统和网络等安全符合 GB 17859—1999、GB/T 22239—2019 和 JR/T 0071—2020 的要求；
c) 密码算法应用及密钥管理实施符合国家密码管理部门有关要求。

8.2 网络安全

建立安全的网络通道，包括以下要求：

a) 网络防火墙：宜采用双层网络防火墙进行保护，即互联网区和 DMZ 区设置一道防火墙，DMZ 区和应用区设置第二道防火墙，并同时设置防火墙网络安全策略，确保网络流量的合法性；
b) 网络隔离区：宜使用 DMZ 网络隔离区，作为外部互联网区与内部安全系统网络区域之间的缓冲区，可以有效保护内部安全系统网络区域的应用安全；
c) 安全协议：应使用 TLS 等安全传输协议来保护数据的安全传输，安全传输协议应采用被广泛应用的较新版本协议，同时禁用弱加密强度算法的选项，如使用 TLS 协议，TLS 协议宜使用 1.2 及以上版本，推荐使用 TLS 1.3 版本协议；
d) 网络安全设备：借助 SSL 加速器和防火墙等网络安全设备，实现 SSL 加速、应用攻击过滤以及拒绝服务(DoS)攻击等安全防御功能；
e) 转发代理：基于资源访问路径的负载路由，可以选择性的将内网应用的接口对外进行暴露。其他要求遵守 GB/T 22239—2019 的规定。

8.3 系统安全

8.3.1 故障隔离

故障隔离应支持分路隔离、第三方隔离、服务隔离、服务系统隔离。

a) 分路隔离是指系统在多路部署的情况下，当其中某一分路发生故障或需要重启时，可对该分路进行隔离，且隔离不影响其他分路的交易；
b) 第三方隔离是指可以对服务请求方进行隔离，一般隔离的维度是在 APP 维度将某一个接入的应用进行隔离；
c) 服务隔离是在 API 维度，针对某一个服务接口进行隔离；
d) 服务系统隔离是在服务提供系统维度，针对某一个服务提供系统进行隔离。

8.3.2 流量控制

流量控制应提供多维度控制，各维度可以自由组合，划分为以下维度：

a) 商业银行应用程序接口共享平台的总流量控制；
b) 开发者流量控制，指针对某一个开发者的接入的总并发数和调用频度进行控制；
c) 服务流量控制，指针对某一个 API 接口接出调用的并发数和调用频度进行控制；
d) 服务系统流量控制，指针对某一个服务提供系统接出调用的并发数和调用频度进行控制。

8.3.3 服务降级

API监控过程中，应对服务质量较低的API进行服务降级。在一段时间内判断服务的处理成功率，并给予降级或恢复。此时间段的长度应可配置。服务质量按照服务SLA的约定。

8.3.4 动态扩容

API监控过程中，应具备动态增加系统处理能力，增加部署资源，以应对API的高峰使用情况。

8.3.5 服务熔断

API监控过程中，当后台API提供者的成功率达到临界阈值以下时，根据配置规则应启动熔断机制，自动隔离该服务，避免扩大影响范围。

8.3.6 IP 黑白名单控制

API服务请求访问中，应支持对服务访问IP的黑白名单配置和控制，通过配置实现控制服务请求者的访问权限。

8.4 数据安全

接口数据安全部分，遵守 JR/T 0185—2020中9.3.3的规定。

应识别应用中涉及的敏感数据。敏感数据包括但不限于GB/T 35273—2020中规定的个人敏感数据、JR/T 0171—2020中规定的C3/C2类数据、银行规定的敏感业务数据。敏感数据的传输、存储应符合GB/T 35273—2020、JR/T 0171—2020以及银行规定的机密性和完整性保护要求。

应支持运营人员对不同用户访问的数据进行控制。当设置为不可见时，用户访问该数据时，应不显示。业务应用系统要支持数据访问权限控制，防止出现开发者跨权限访问业务数据。

8.5 业务安全

建立智能风控体系，加强银行合规监管能力。根据合作方建立准入准出机制防范风险。针对银行业务流程，业务安全包括以下要求。

a) 电子账户管理：应遵循行政主管部门对二三类账户的使用和管理要求。

b) 业务风险监控：应对不同业务建立风控模型，降低业务风险；同时对资金交易的风险进行监控，满足反洗钱和反欺诈的交易管理要求，对大额资金变动进行监控。

c) API鉴权：应对接入的开发者及其应用进行身份核验，包括接入申请阶段业务上的核验以及API调用的身份认证：

1) 核验方法包括但不限于数字证书、token、OAuth2.0等技术；

2) API调用应提供基于APPKEY的服务访问权限控制；

3) 资金交易应充分识别是否由用户本人发起，核实用户本人意愿，其中，token应设定有效期；

4) 相关密钥或证书应具有主动失效、更换更新机制。

d) API授权：平台应提供合理的流程机制，允许外部用户申请订阅使用API；平台运营人员应在充分评估的基础上，对订阅申请进行审批。

e) 不可抵赖性：平台应对不同类型的业务进行分级管理，涉及资金等重要类型的业务应使用数字签名进行验证。

8.6 安全审计

通过安全审计，保证系统安全：

a) 建立安全策略，密码应具备一定强度且应定期更换，账号权限应符合最小授权要求等；

b) 日志审计，应采集、分析系统日志，包括操作系统、数据库等的操作日志进行日志审计，并对日志进行保护。