团　体　标　准

无锡市银行领域应用程序接口共享平台技术规范

Technical Specification of the Application Programming Interface（API）Sharing Platform in the Banking Sector of Wuxi City

2025-4-1 发布　　2025-4-1 实施

全国团体标准信息平台

目 录

全国团体标准信息平台

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》给出的规则起草。

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。

本文件由中国人民银行无锡市分行提出并归口。

本文件主要起草人：龚汇丰、纪兴哲、宋成成、付晓东、奚祯、程磊、张雪飞、陆鼎。

本文件起草单位：中国人民银行无锡市分行、无锡锡商银行股份有限公司。

全国团体标准信息平台

无锡市银行领域应用程序接口共享平台技术规范

1　范围

本文件规定了无锡市银行领域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。

本文件适用于无锡市银行建设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—2020.2　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

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

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

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

开发者的应用的使用者。

3.5　应用程序接口

一组预先定义好的函数，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

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

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

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 安全传输协议

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

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：超文本标记语言5.0（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

无锡市银行领域应用程序接口共享平台（以下简称“应用程序接口共享平台”）是基于一套标准化协议及接口对第三方提供技术支持，借助互联网技术，通过开放金融服务，提供标准化的SDK与个性化的API(应用程序接口)，满足不同对接方的需求(包括但不限于企业、

商户及第三方合作伙伴)。开发者通过银行提供的开放API(应用程序接口)或SDK进行对接，银行对外提供的API和SDK接口规范需遵循JR/T 0185—2020要求。

银行、开发者是应用程序接口的主要参与方，通过应用程序接口共享平台实现对接，开发者向应用程序接口共享平台发送相关请求，应用程序接口共享平台将请求转发至银行业务系统，将结果返回。

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。

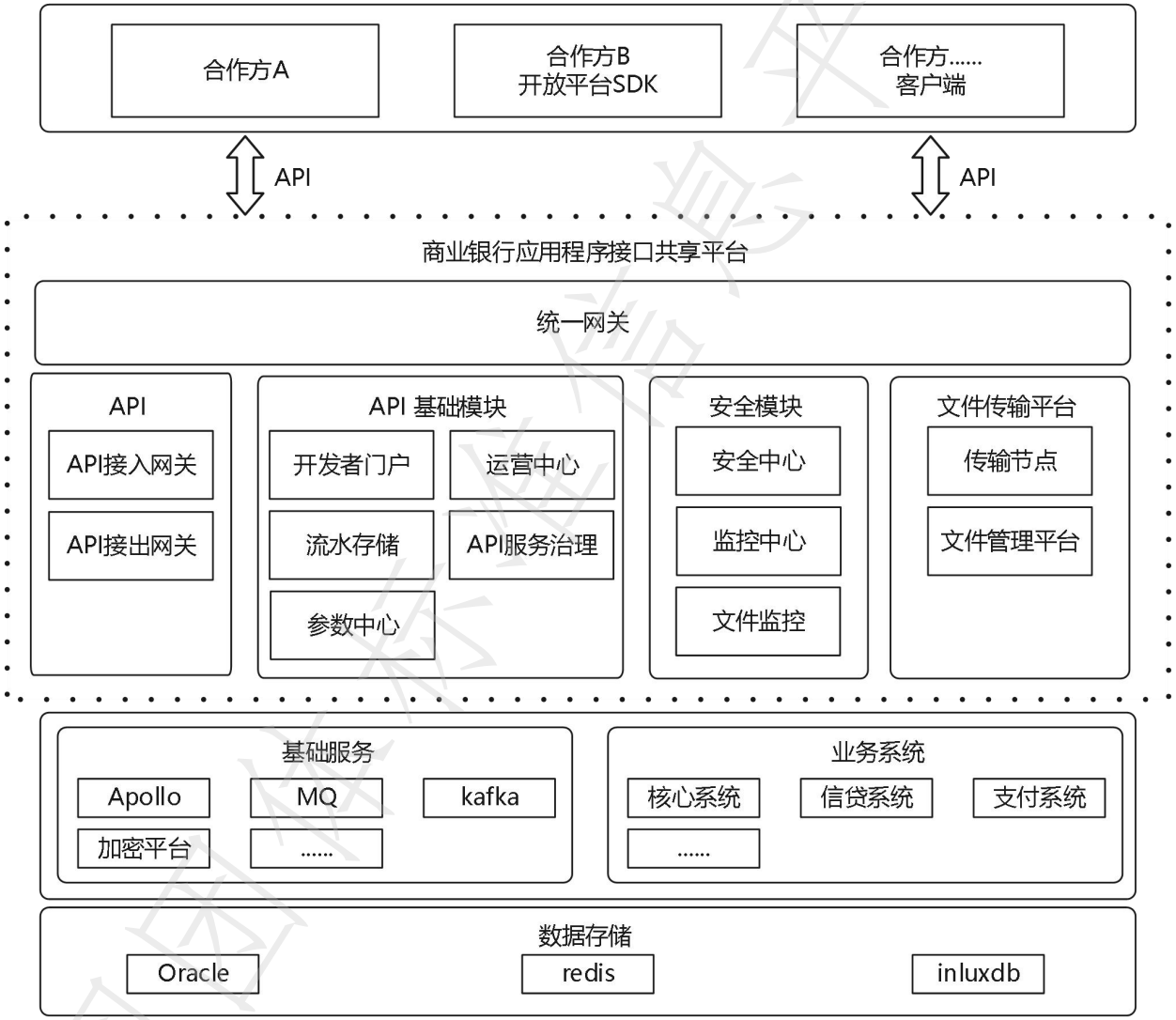

图 1 应用程序接口共享平台逻辑结构图

6 功能架构

6.1 整体架构

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、文件传输平台、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。

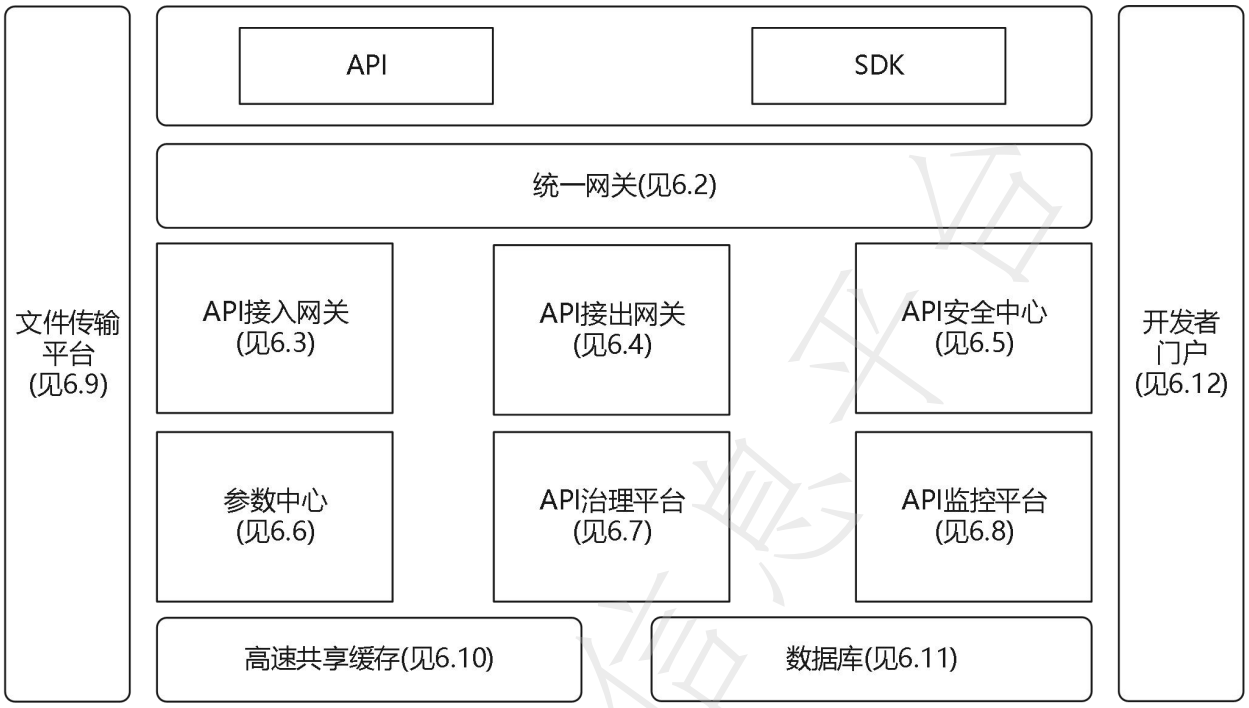

图2　系统功能架构

6.2　统一网关

银行的统一网关是微服务架构体系中的重要组成单元。属于系统中的一个提供外部访问Rest API服务，主要起到服务隔离作用。它是微服务系统中的门面，所有外部（跨系统）的服务请求调用都要经过服务网关，由它来进行服务的调度和过滤。提供了服务路由、服务发现、负载均衡、权限认证、服务限流、服务熔断、服务降级、服务鉴权等功能。网关中包含灵活的支持热部署的过滤器框架，以满足对统一的处理逻辑在网关上进行实现，还提供易用的配置能力便于维护。

6.3　API 接入网关

API接入网关是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，根据服务提供方情况，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） 交易路由策略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；

b） 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；

c） 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；

d） API接入网关应当支持大量API迁移功能。

6.4　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。

6.5 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、OAuth2.0 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，支持开发者对应密钥的查询与查看，密钥的生成、重置、销毁，支持密钥对接加密平台的管理;
   3) 开发者权限管理，提供开发者的权限控制管理，开发者接口权限申请的审核与复核、开发者接口权限的查询与查看，开发者接入权限的冻结与解冻。

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) 令牌（token）管理，提供 token 的生成、存储、更新管理，支持 token 有效期的配置化，提供 token 的自动续期;
   2) 授权，提供基于 token 的授权管理;
   3) 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，包括开发者、业务、接口维度的流量控制;
   2) 限流算法，宜具备多种限流算法，比如令牌桶算法等，根据不同场景选择合适的限流算法;
   3) 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) 国密摘要算法，提供基于国密算法的摘要算法 SM3;
   5) 国际摘要算法，提供基于国际算法的摘要算法 Hash;
   6) 数据脱敏，提供敏感数据的脱敏传输;
   7) 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；
h）服务隔离：支持将异常服务节点从服务注册中心下线，避免某个节点异常导致整个系统性能下降，从而保障系统的稳定性；
i）流量配置：支持对接入、接出网关应用接口、业务及开发者配置令牌数，实现流量控制，防止系统因瞬时流量过载而崩溃，确保网关服务的可用性和稳定性。

6.6.2 参数中心基本功能要求

参数中心应该有以下基本功能：

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。
h) 脱敏还原配置：对接口请求或响应中的敏感字段信息进行脱敏和还原规则的配置，比如对身份证号、电话号码、银行卡号等字段的脱敏还原规则配置，提高接口交互中信息的安全性。
i) 服务配置管理：通过对服务参数的配置和服务配置的导出导入，实现服务的快速配置，提高服务配置效率。
j) 配置导出：支持导出平台已配置的服务定义、服务协议、基础服务、服务流程等配置信息，不同环境导入该压缩包，实现服务的快速配置化管理。
k) 配置导入：支持导入服务配置压缩包，提高服务配置效率。

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 监控平台应对应用系统的资源、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) API 监控平台提供系统资源运行情况，包括应用服务器 CPU、内存、磁盘、网络 IO 的监控、健康监测与预警；
b) API 监控平台提供服务运行情况，包括 JVM 内存、应用 CPU、线程状态、句柄数及应用 GC 情况的实时监控与预警；
c) API 监控平台提供实时流量统计分析功能，支持多种维度、多种指标查看交易运行情况，提供丰富的图表展示；
d) API 监控平台提供各时段多维度的交易统计报表，包括服务提供方、服务消费方、分路流量、异常交易、成功率等详细的报表数据；
e) API 流水查询功能包括对所有 API 网关进行的交易流水进行查询，支持多种查询条件，如流水号、时间段、时间点、交易状态、交易返回码等；
f) API 监控平台提供实时告警功能，支持短信、邮件等告警通知类型，并支持告警指标、告警阈值及告警用户的自定义设置；
g) API 监控平台提供运营报告的查看与导出功能，汇总与统计每月交易运行情况，提供丰富的数据展示；
h) 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 文件传输平台

通过文件传输平台的建设，实现开发者与银行之间的文件的安全、高效、可靠传递，在网络故障、主机宕机等各种意外情况下，应做到断点续传，保证数据“一次传递、可靠到达”；使文件的传输可见、可控、可管；并提供对文件的格式化、压缩、重命名以及其它定制化的处理功能。

安全中继：包括对开发者上传的文件暂时落地处理、银行对外提供文件的传输处理及文件的病毒扫描。开发者通过集成银行提供的SDK进行文件上传或下载的请求，SDK接收到请求通过互联网转发至安全中继，调用数据节点进行服务安全认证后，安全中继对请求数据进行处理，提供文件的暂时落地和文件病毒扫描及风险处置。

数据节点:实现文件传输过程中文件的上传和下载，其主要功能包括流量控制、安全认证、用户权限、实时文件上传/下载、批量文件上传/下载等功能。文件上传和下载需支持分片传输，并进行文件md5完整性校验，从而提升文件传输速度和安全性。数据节点将文件上传和下载记录发送到kafka消息中心，最终实现传输数据记录的持久化。

文件管理平台:实现数据节点的界面化管理服务，通过对接配置中心，实现对各文件传输数据节点配置统一管理，其主要功能包含权限管理(运营人员功能菜单和操作权限等)、数据节点信息管理(数据节点名称、IP地址等)、服务管理（上传、下载等接口流程的配置）、用户管理(开发者信息的查询与查看，开发者接入信息的维护等)、定时任务(定时任务的查询与查看，修改、删除等)、配置下发等功能。

传输监控平台:实现系统的统一监控，包括监控指标维护、告警设置、应用服务器监控、流水收集监控、文件的传输统计功能、文件上传记录、文件下载记录、节点告警统计、节点告警查询、资源监控日志查询、故障与告警等功能，需支持短信和邮件两种告警方式。

6.10 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，应采用 redis 高速共享缓存方案，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，系统应确保能从其他数据源恢复，保证对用户的持续服务不中断；
c) 高速缓存需要具备相应的内存、状态等的监控告警，确保缓存健康运行，并及时响应潜在的性能瓶颈或故障。

6.11 数据库

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，应支持一种或多种高可用架构的数据库类型，包括分布式数据库、读写分离架构、共享集群等。数据库部署方案应具备高可用性、可扩展性，并能承受高并发访问；
b) 面向互联网的服务宜支持读写分离功能，以提高读操作的性能，并确保写操作的可靠性。同时，针对高频操作的数据表（如流水表），宜支持分表或分库操作，优化查询性能并分散数据库压力；
c) 面向互联网提供入口服务功能的应用，宜能够脱离数据库提供服务，借助缓存、消息队列或其他中间件技术，提高系统的响应速度和可用性。

6.12 开发者门户

开发者门户应包括以下核心功能模块：

a) 产品中心：集中展示平台产品分类，简介，功能以及特色；
b) 帮助中心：提供开发工具、SDK、客户端及其他相关资源的下载服务，提供新手引导，帮助开发者解决在开发和接入过程中遇到的问题，还应包含 API 文档、开发指南、接口规范等资源；
c) 新闻公告：发布平台最新动态、技术更新、版本发布等公告信息；
d) 开发者注册与实名认证：门户应支持开发者注册与实名认证，确保开发者的合法与合规性；
e) 应用中心：支持开发者查看、申请及管理其应用，提供应用的注册、审核、上线、下线功能；
f) 应用服务申请与管理：支持开发者申请并管理平台上的各类业务服务、产品及 API 接口，并提供产品上线、下线、更新等功能；
g) 密钥与证书管理：门户应支持开发者上传证书与密钥，确保 API 交互过程中的数据安全。

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，应采用 https 作为应用层协议，只支持 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，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状态码的表述或返回相对友好的响应码和响应信息，告知客户问题原因。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：https://openapi.baidu.com/rest/2.0/passport/users/getLoggedInUser

请求：https://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设计基本要求

应用程序接口安全设计基本要求如下：

a) 使用的密码算法、技术及产品应符合国家密码管理部门及行业主管部门要求；

b) 应制定安全编码规范；

c) 应对开发人员进行安全编码培训，并依照安全编码规范进行开发；

d) 开发中如需使用第三方应用组件，应对组件进行安全性验证，并持续关注相关平台的信息披露 和更新情况，适时更新相关组件；

e) 应对应用程序接口进行代码安全专项审计,审计工作可通过人工或工具自动化方 式开展；

f) 应制定源代码和应用程序接口版本管理与控制规程，规范源代码和应用程序接口版本管理，并就接口废止、变更等情况与开发者保持信息同步；

g) 银行向开发者提供的异常与调试信息，不应泄漏服务器、中间件、数据库等软硬件信 息或内部网络信息。

7.3 API接入/接出网关

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。

7.4 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、文件传输以及 API 等融为一体提供给开发者，大大减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与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、Nodejs、C#、Python、PHP 等主流开发语言。

7.5 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。

7.6 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。

7.7 API 治理

API 治理是基于元数据的服务定义、描述、映射、检索、治理系统，以提升组织业务能力为导向的工具，提供对所有现有以及将来新增服务所进行的整理、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行内标准服务。

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应符合JR/T 0185—2020中7.1的要求；

b） 统和网络等安全应符合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应符合国家密码管理部门有关要求。

8.2 API 安全

8.2.1 身份认证安全

a） 接口身份认证安全要求如下：

1） 应使用App_ID、公私钥对的方式进行双向身份认证；

2） 应支持App_ID、公私钥对、数字证书组合的方式进行双向身份认证。

b) 用户身份认证安全要求如下：

1） 银行应结合金融服务场景，对不同安全级别的应用程序接口设计不同级别的用户身份认证机制；

2）用户身份认证应在银行执行，对于资金交易类服务，用户登录身份认证应至少使用双因子认证的方式来保护账户财产安全。

8.2.2 接口交互安全

接口交互安全要求如下：

a）应对连通有效性进行验证，如接口版本、参数格式等要素是否与平台设计保持一致；

b）应对通过应用程序接口进行交互的数据进行完整性保护，对于A2级别的接口，银行和开发者应使用数字签名来保证数据的完整性和不可抵赖性；

c）对于支付敏感信息等个人金融信息，应采取以下措施进行安全交互：

1）登录口令、支付密码等支付敏感信息在数据交互过程中应使用包括但不限于替换输入框原 文、自定义软键盘、防键盘窃听、防截屏等安全防护措施，保证无法获取支付敏感信息明 文

2）账号、卡号、卡有效期、姓名、证件号码、手机号码等个人金融信息在传输过程中应使用 集成在SDK中的加密组件进行加密，或对相关报文进行整体加密处理；若确需使用应用程序接口将账号、卡号、姓名向开发者进行反馈，应脱敏或去标识化处理，因 清分与清算、差错对账等需求，确需将卡号等支付账号传输至开发者时，应使用加密通道 进行传输，并采取措施保证信息的完整性；

3）对于金融产品持有份额、用户积分等A2类只读信息查询，可使用API直接连接方式进行 查询请求对接，应采取加密等措施保证查询信息的完整性与保密性，查询结果在开发者本 地不得保存。

d）应在交易认证结束后及时清除用户支付敏感信息，防范攻击者通过读取临时文件、内存数据等方式获得全部或部分用户信息。

8.2.3 密钥管理安全

密钥安全要求如下：

a）加密和签名应分配不同的密钥，且相互分离；

b）不应以编码的方式将私钥明文（或密文）编写在应用程序相关代码中，私钥不应存储 于银行与应用方本地配置文件中，防止因代码泄露引发密钥泄露；

应依据应用程序接口等级设置不同的密钥有效期，并对密钥进行定期更新。

8.3 网络安全

建立安全的网络通道，应包括：

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，通过 TCP4 层信息进行网络数据包的过滤，生产系统的数据流量必须提供源 IP、目标 IP、目标端口,才能开通策略，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，银行要求开发者应使用 TLS1.2 及以上版本通信协议进行接入；

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) 网络监控和防护：采用 IPS（基于 IPS 特征库来进行防御）、WAF(针对每个对外发布的 HTTP 应用,基于 WAF 特征库进行防御)、风险检测(通过风险流量态势感知产品实现风险数据的识别、告警、回溯)等防护策略,并针对网站挂马、黑链、违规内容等进行监控；

f) 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。

8.4 系统安全

8.4.1 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。

8.4.2 流量控制

流量控制应提供多维度控制，应包括下列维度：

a) 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。

8.4.3 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。

8.4.4 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。

8.4.5 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。

文件传输过程中，应支持对服务访问IP的黑白名单配置，进行访问权限及开发者访问目录权限控制，通过配置实现控制文件服务器的访问权限。

8.5 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。相应敏感数据的传输、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。

开发者在数据安全保护方面的安全要求如下：

a) 数据完整性保护：应对数据完整性进行校验，并在检测到完整性错误时停止执行请求；

b) 数据机密性保护：

1) 不应采集、存储用户个人金融信息或支付敏感信息；

2) 对于需要用户输入支付敏感信息或身份鉴别信息的场景，开发者仅可作为信息的采集与传输通道，应部署银行SDK、采取报文加密等措施，保证采集与传输信息的机密性与完整性，支付敏感信息与身份鉴别信息在开发者不得留存。

c) 数据抗抵赖性保护：应使用数字签名等技术确保A2类数据的不可抵赖性；

d) 数据删除与销毁：在合作终止后，应依据与银行约定的方式删除（或销毁）通过应用程序接口获取的银行及其用户的相关数据；

e) 应针对接口处理的数据，建立数据备份管理机制和应急灾备机制，并纳入机构灾备体系。在合作终止后，应依据行业主管部门有关要求，履行反洗钱、反欺诈等义务。

应支持运营人员对不同用户访问的数据进行控制。开放平台运营中心具备菜单权限控制，针对不通运营人员，支持分配不同菜单权限，当设置为不可见时，用户访问该数据，应不显示。业务应用系统要支持数据访问权限控制，开放平台门户针对产品进行权限控制，在开发者申请产品时，只允许开发者选择分配的产品进行接口权限的申请，防止出现开发者跨权限访问业务数据。

8.6 业务安全

建立智能风控体系，加强银行合规监管能力。根据合作企业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：

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：

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本人发起,核实用户本人意愿，其中，token应设定有效期且有 效期可配置，应具备token自动续期机制；

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，开放平台提供开发者门户，用于开发者进行实名注册，在实名认证后，可根据开放平台运营中心分配的产品权限，进行接口权限的申请；平台运营人员应在充分评估的基础上，通过运营中心对申请实名信息进行审核，对开发者权限申请进行审核及复核；

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。

8.7 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；

银行开放平台门户开发者登录密码必须包含大小字母、数字，并采用加密键盘进行密码输入，应设置密码有效期，到期需进行更换，银行开放平台运营中心采取双重授权模式，分配审核和复核员，保证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。银行应完整记录应用程序接口访问日志，日志记录应至少包括交易渠道来源、交易流水号、应用唯一标识、接口唯一 标识、调用耗时、时间戳、返回结果（成功或失败）等,应保留自投产以来的日志，并对日志根据接口维度进行备份。依据商业服务需求和风险控制要求，遵循最少够用原则适当保留开发者上送报文（全部或部分信息）,应对日志记录进行完整性保护，确保日志不被篡改、删除、覆盖。开发者应完整记录应用程序接口访问日志，日志记录应至少包括交易渠道来源、交易流水号、应用唯一标识、接口唯一标识、调用耗时、时间戳、返回结果（成功或失败）等,应对日志记录进行完整性保护，确保日志不被篡改、删除、覆盖,应提供查询开发者用户应用程序接口相关登录、授权、交易等历史操作日志功能。

全国团体标准信息平台