书

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | \*Status  Change | Historic Modification | Date | Compiler | Remark |
| V0.1 | A | Base version | 2023/03/19 | Kalvan |  |
|  |  |  |  |  |  |

**目录**

[1 整体介绍 10](#_Toc950905928)

[1.1 平台介绍 10](#_Toc302630922)

[1. 产品架构 10](#_Toc1076629958)

[2. 核心功能 11](#_Toc222494484)

[1.2 平台优势 14](#_Toc695763161)

[1. 统一的平台 14](#_Toc642989012)

[2. 卓越的性能 14](#_Toc578612980)

[3. 安全的服务 14](#_Toc942401244)

[4. 专业的指标 14](#_Toc1245811283)

[5. 灵活的发布 14](#_Toc384675131)

[6. 丰富的组件 15](#_Toc1309149247)

[1.3 应用场景 15](#_Toc1901430814)

[1. 移动应用新开发 15](#_Toc643539891)

[2. 已有移动应用 15](#_Toc1247301745)

[2 详细设计 15](#_Toc1812549848)

[2.1 客户端设计 15](#_Toc1469762641)

[2.1.1 技术架构 15](#_Toc1943799493)

[2.1.2 H5容器 16](#_Toc1916840687)

[2.1.2.1 服务概述 17](#_Toc1939237762)

[2.1.2.2 应用场景 17](#_Toc409755415)

[2.1.2.3 产品架构和技术原理 17](#_Toc1926687623)

[2.1.2.3.1 总体架构 17](#_Toc2080450295)

[2.1.2.3.2 H5 容器工作流程 18](#_Toc799367611)

[2.1.2.4 产品优势 19](#_Toc313742445)

[2.1.3 社交分享 20](#_Toc996919730)

[2.1.3.1 服务概述 20](#_Toc562488216)

[2.1.3.2 系统架构及技术原理 20](#_Toc516432218)

[2.1.3.2.1 总体架构 20](#_Toc1694870399)

[2.1.3.2.2 操作流程 21](#_Toc1463702185)

[2.1.3.3 产品优势 22](#_Toc1017446910)

[2.1.4 扫一扫 22](#_Toc1965418956)

[2.1.4.1 服务概述 22](#_Toc202935338)

[2.1.4.2 应用场景 23](#_Toc530194330)

[2.1.4.3 产品优势 23](#_Toc1066452907)

[2.1.5 定位 24](#_Toc975490087)

[2.1.5.1 服务概述 24](#_Toc1171731011)

[2.1.5.2 应用场景 25](#_Toc858058887)

[2.1.5.3 产品优势 26](#_Toc1043024204)

[2.1.6 统一存储 26](#_Toc198786167)

[2.1.6.1 服务概述 26](#_Toc1662037684)

[2.1.6.2 应用场景 27](#_Toc1547558459)

[2.1.6.3 产品优势 27](#_Toc1640571596)

[2.1.6.3.1 数据库存储（WCDB） 27](#_Toc1544270139)

[2.1.6.3.2 键值对存储（MMKV） 30](#_Toc60868531)

[2.1.6.3.3 文件存储（File） 30](#_Toc815184545)

[2.1.7 UI组件库 31](#_Toc2008463602)

[2.1.7.1 服务概述 31](#_Toc2099795268)

[2.1.7.2 应用场景 31](#_Toc1660298125)

[2.1.7.3 系统架构及技术原理 31](#_Toc228077757)

[2.1.7.4 产品优势 32](#_Toc44552004)

[2.2 中台设计 33](#_Toc1461222072)

[2.2.1 技术架构 33](#_Toc136377012)

[2.2.2 应用架构 39](#_Toc723389335)

[2.2.3 移动网关 41](#_Toc1099627678)

[2.2.3.1 服务概述 41](#_Toc198118064)

[2.2.3.2 应用场景 41](#_Toc1170648798)

[2.2.3.3 系统架构及技术原理 41](#_Toc1996657819)

[2.2.3.4 产品优势 43](#_Toc1248495911)

[2.2.4 消息推送 44](#_Toc408061340)

[2.2.4.1 服务概述 44](#_Toc1371656509)

[2.2.4.2 应用场景 45](#_Toc193996218)

[2.2.4.3 系统架构及技术原理 45](#_Toc614259780)

[2.2.4.4 产品优势 46](#_Toc910231331)

[2.2.5 数据同步 47](#_Toc1731962536)

[2.2.5.1 服务概述 47](#_Toc2100991114)

[2.2.5.2 系统架构及技术原理 47](#_Toc284045377)

[2.2.5.3 产品优势 48](#_Toc94503958)

[2.2.6 应用发布 48](#_Toc1337606973)

[2.2.6.1 服务概述 48](#_Toc1301578415)

[2.2.6.2 应用场景 49](#_Toc1359992563)

[2.2.6.3 系统架构及技术原理 49](#_Toc1726551320)

[2.2.6.4 产品优势 50](#_Toc1348996976)

[2.2.7 离线包 51](#_Toc1607314253)

[2.2.7.1 服务概述 51](#_Toc933854558)

[2.2.7.2 应用场景 51](#_Toc1483064030)

[2.2.7.3 系统架构及技术原理 51](#_Toc14461481)

[2.2.7.3.1 总体架构 52](#_Toc388459056)

[2.2.7.4 产品优势 52](#_Toc481067312)

[2.2.8 移动监控 53](#_Toc22381829)

[2.2.8.1 服务概述 53](#_Toc361761778)

[2.2.8.2 系统架构及技术原理 54](#_Toc603998189)

[2.2.8.3 应用场景 54](#_Toc242363154)

[2.2.8.4 产品优势 55](#_Toc1768534566)

[2.2.9 移动分析 57](#_Toc439292635)

[2.2.9.1 产品介绍 57](#_Toc142538059)

[2.2.9.2 产品架构 57](#_Toc1192891208)

[2.2.9.3 产品优势 59](#_Toc15204464)

[2.2.10 设备标识 59](#_Toc2138356102)

[2.2.10.1 服务概述 60](#_Toc1212173769)

[2.2.10.2 应用场景 60](#_Toc1974660141)

[2.2.10.3 系统架构及技术原理 60](#_Toc900709049)

[2.2.10.4 功能组成及详细说明 61](#_Toc604758840)

[2.2.11 金融键盘 61](#_Toc141722629)

[2.2.11.1 服务概述 61](#_Toc372861080)

[2.2.11.2 产品优势 62](#_Toc318889614)

[2.2.12 智慧投放 65](#_Toc1606043233)

[2.2.12.1 服务概述 65](#_Toc1046657888)

[2.2.12.2 架构说明 65](#_Toc1140571039)

[2.2.12.3 应用场景 65](#_Toc1138419351)

[2.2.12.4 产品优势 66](#_Toc1482221134)

[2.2.13 移动开发框架 67](#_Toc880293938)

[2.2.13.1 服务概述 67](#_Toc1085371783)

[2.2.13.2 应用场景 67](#_Toc1117459263)

[2.2.13.3 系统架构及技术原理 68](#_Toc1393340226)

[2.2.13.4 产品优势 69](#_Toc1707491494)

[2.3 安全设计 70](#_Toc985564797)

[2.3.1 环境安全 70](#_Toc846173868)

[2.3.1.1 概述 71](#_Toc1007489042)

[2.3.1.1.1 环境安全SDK服务解决方案产品价值 71](#_Toc2107255946)

[2.3.1.1.2 环境安全SDK服务解决方案产品原理 71](#_Toc350378098)

[2.3.1.1.3 环境安全SDK服务解决方案产品组成 72](#_Toc404533012)

[2.3.1.2 产品详述 72](#_Toc53106282)

[2.3.1.2.1 病毒扫描 72](#_Toc1351568069)

[2.3.1.2.2 网址检测 76](#_Toc1870001364)

[2.3.1.2.3 WiFi安全检测 79](#_Toc689750903)

[2.3.1.2.4 防界面劫持 85](#_Toc526700215)

[2.3.1.2.5 防录屏 87](#_Toc322920571)

[2.3.1.2.6 防截屏 88](#_Toc634860828)

[2.3.1.2.7 root检测 90](#_Toc1407177900)

[2.3.1.2.8 越狱检测 92](#_Toc201560889)

[2.3.1.2.9 模拟器检测 93](#_Toc1052150104)

[2.3.1.3 产品优势 96](#_Toc1106448530)

[2.3.1.3.1 深耕移动安全，扎实能力积累 96](#_Toc1019544337)

[2.3.1.3.2 国际领先AI威胁检测引擎 97](#_Toc709652546)

[2.3.1.3.3 屡获各类国际测评机构最高认证 97](#_Toc6165184)

[2.3.1.3.4 腾讯独有的态势感知与威胁线索追踪能力 98](#_Toc539032432)

[2.3.1.3.5 多年B端合作经验，海量C端用户服务经验 98](#_Toc1432061578)

[2.3.2 应用加固 99](#_Toc1809709517)

[2.3.2.1 腾讯安全加固技术点简介 99](#_Toc976959758)

[2.3.2.2 安全产品及服务 100](#_Toc102687744)

[2.3.2.2.1 安全产品及服务列表 100](#_Toc1443544867)

[2.3.2.2.2 产品交付及使用方式 102](#_Toc1535819510)

[2.3.2.3 反编译与反汇编保护 104](#_Toc1912551277)

[2.3.2.3.1 背景 104](#_Toc714084243)

[2.3.2.3.2 技术原理 104](#_Toc1475252665)

[2.3.2.3.3 技术参数 107](#_Toc1872836040)

[2.3.2.4 客户端防篡改保护 108](#_Toc1087510201)

[2.3.2.4.1 背景 108](#_Toc550628590)

[2.3.2.4.2 技术原理 108](#_Toc907677207)

[2.3.2.4.3 技术参数 108](#_Toc1754473408)

[2.3.2.5 客户端防注入保护 109](#_Toc336611299)

[2.3.2.5.1 背景 109](#_Toc954176095)

[2.3.2.5.2 技术原理 109](#_Toc1577236516)

[2.3.2.5.3 技术参数 109](#_Toc75985844)

[2.3.2.6 客户端反调试保护 110](#_Toc1488793790)

[2.3.2.6.1 背景 110](#_Toc1825257333)

[2.3.2.6.2 技术原理 110](#_Toc296098336)

[2.3.2.6.3 技术参数 110](#_Toc805123053)

[2.3.2.7 客户端本地数据与资源文件加密保护 111](#_Toc408692024)

[2.3.2.7.1 背景 111](#_Toc1234144262)

[2.3.2.7.2 技术原理 111](#_Toc1865548708)

[2.3.2.7.3 技术参数 111](#_Toc1015889156)

[2.3.2.8 SO库绑定保护 112](#_Toc1554051242)

[2.3.2.8.1 背景 112](#_Toc1243109480)

[2.3.2.8.2 技术原理 112](#_Toc72628697)

[2.3.2.8.3 技术参数 112](#_Toc899798983)

[2.3.2.9 运行时环境检测技术 113](#_Toc341665107)

[2.3.2.9.1 背景 113](#_Toc2141664918)

[2.3.2.9.2 技术原理 113](#_Toc988869459)

[2.3.2.9.3 技术参数 113](#_Toc553053280)

[3 系统管理 114](#_Toc857252744)

[3.1 应用管理 114](#_Toc379080685)

[3.1.1 我的应用 114](#_Toc1772575793)

[3.1.2 应用管理 114](#_Toc1788201767)

[3.2 通知管理 115](#_Toc273458204)

[3.3 审计日志 115](#_Toc397030048)

[4 部署方案 115](#_Toc652325507)

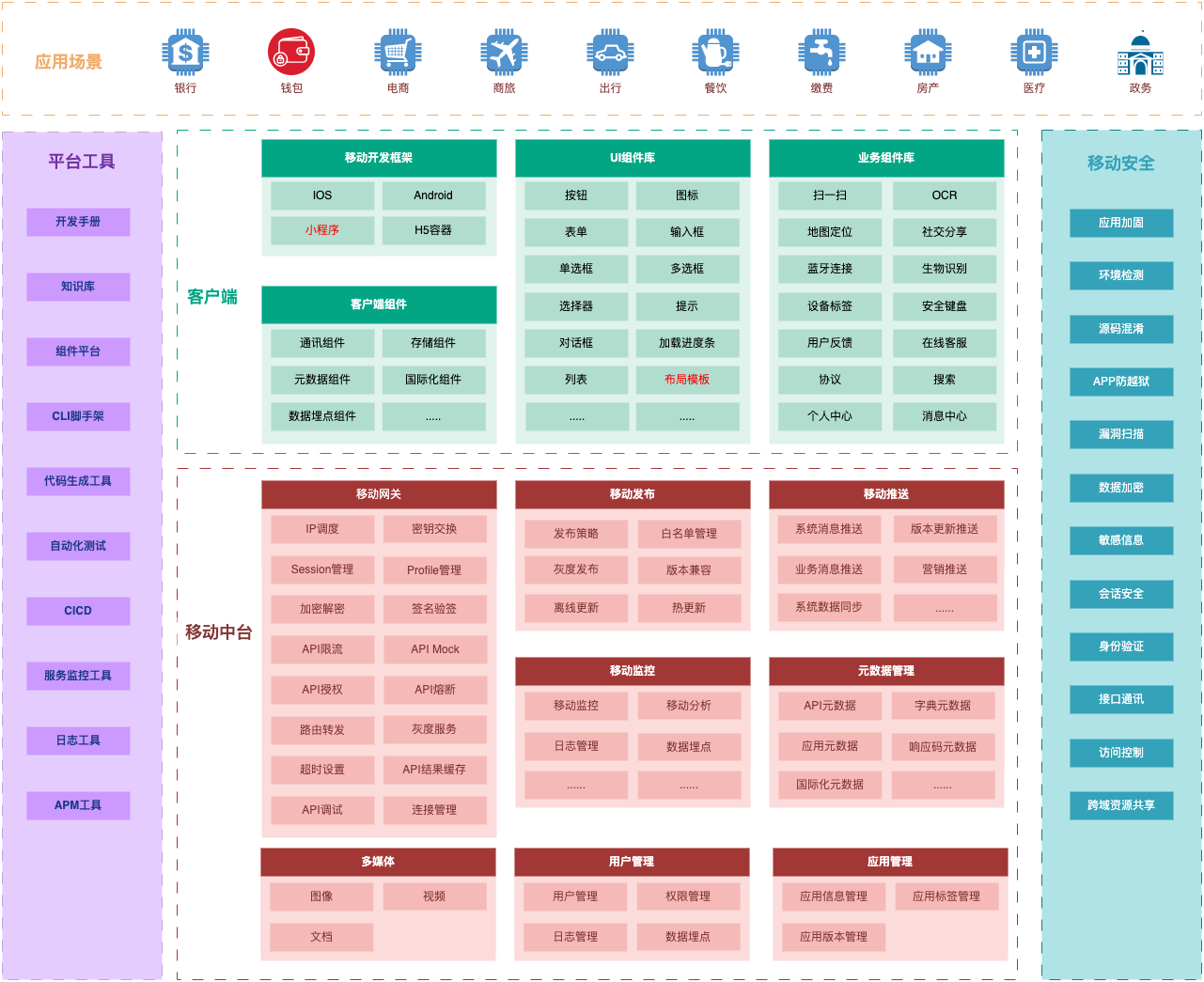
# 整体介绍

## 平台介绍

移动金融开发平台（Ferrari Mobile FinTech Platform）整合了开发、测试、发布和运营的技术能力，为银行、互联网金融提供一站式、覆盖全生命周期的移动端金融技术平台。能够快速构建手机银行、电子钱包、商家收单等移动互联场景的应用程序。

平台核心功能模块包括移动客户端开发组件、移动网关、移动发布、移动推送、移动监控、元数据管理、移动安全、CLI脚手架等移动端开发运营关键服务模块，帮助金融机构低成本、高效率地打造移动金融服务。

1. **产品架构**



**平台由客户端、移动中台、移动安全、平台工具构成**

* 客户端：主要是包含开发框架底坐、客户端组件、UI组件库、业务组件库等功能
* 移动中台：支撑客户端应用移动网关、移动发布、移动推送、移动监控、元数据管理、应用管理、用户管理、多媒体等标准化的服务功能
* 移动安全：包括客户端应用安全、数据安全、网络安全、合规检查要求等
* 平台工具：提供开发手册、知识库、组件库、脚手架、自动化测试工具、CICD、服务监控工具、日志工具、APM等辅助开发的工具

1. **核心功能**

|  |  |  |
| --- | --- | --- |
| **类别** | **服务模块** | **服务说明** |
| 客户端 | 通讯组件 | 安全通讯技术组件，能够标准化和后端的通讯 |
| 存储组件 | 提供客户端持久化存储解决方案，底层提供多样化的存储方式以满足不同的存储需求，支持数据库加密功能 |
| 元数据组件 | 系统参数、控制开关、返回码、选项字典等数据 |
| 国际化组件 | 能够基于元数据快速配置国际化 |
| UI组件库 | 统一组件库以标准化的视觉规范为基础，将抽象的视觉规范概念转化为控件实体。作为开发人员，通过使用统一组件库，可以在接入控件时，实现客户端视觉规范的统一。 |
| 扫一扫 | 能快速扫描识别二维码、条形码等码，得到其蕴含的相关信息。 |
| OCR |  |
| 定位 | 定位SDK是一套简单的LBS定位接口，您可以使用这套定位API获取定位结果、逆地理编码等功能 |
| 社交分享 | 分享组件提供微博、微信、QQ、短信等渠道的分享功能，提供给开发者统一的接口，无须处理各 SDK 的接口差异性。 |
| 地图 | 基于地图基础数据和定位配合使用，需要集成如谷歌服务或地图数据提供厂商的服务 |
| 蓝牙连接 | 按标准的蓝牙协议连接硬件设备支持扩展IOT的应用场景 |
| 生物识别 | 根据手机设备的特性来识别和提供标准流程的生物认证服务，主要指纹识别、人脸识别 |
| 设备标签 | 用户运行的环境信息采集，不同的标签用来支撑推送、安全、监控等应用场景，包括设备号、设备类型、系统版本、网络环境、语言等标签信息 |
| 安全键盘 | 安全键盘组件提供键盘控件和安全的 EditText 输入控件，保证信息输入安全；同时提供云端拉取键位显示与键值映射数据，保证与云端的动态联动；约束好 一次一密流程，保证数据流的安全性 |
| 用户反馈 | 用户反馈是一种非实时的问题反馈功能，可由用户主动提出再由后台跟进处理 |
| 在线客服 | 提供实在在线的问答客服，能更高效的处理问题，增强用户体验，需要集成第三方服务来实现 |
| 移动中台 | 移动网关 | 接口网关，连接客户端与内部微服务的组件，客户端像调用本地方法一样直接调用网关接口，网关将请求转发给内部服务接口。网关服务还支持加密、签名验证、限流、服务发布、服务授权等功能，提高客户端访问的安全。 |
| 应用版本管理 | 管理应用的历史版本、下载数据、发布策略等版本信息 |
| 应用标签管理 | 用户进行应用下载安装和使用过程中产生标签，基于标签可对目标用户进行定向推送和风控 |
| 灰度发布 | 基于多维度的灰度规则进行的客户端升级功能。支持App灰度发布策略管理，支持按用户、机型、系统版本、App版本等多种维度的灰度策略，精准控制发布范围。 |
| 版本兼容 | 不同版本可能导致和后端API不兼容需要通过兼容配置来正确的处理对应版本APP的业务请求 |
| 离线更新 | 提供将HTML、CSS等静态资源预加载到客户端的能力，提升访问速度和在弱网或无网情况下的访问体验 |
| 热更新 | 提供紧急修复线上bug，无需重新发版即可在用户无感知的情况下完成问题修复的能力，支持多种修复策略设置 |
| 系统消息推送 | 系统级公共通知类消息推送 |
| 版本更新推送 | 根据版本发布策略推送 |
| 业务消息推送 | 定义业务消息模板，由业务场景触发推送，提供专业的移动消息推送方案，针对不同的场景推出多种推送类型，满足用户的个性化推送需求，在提供控制台快速推送能力的同时，也提供了服务端接入方案，方便用户快速集成移动终端推送功能，与用户保持互动，从而有效地提高用户留存率，提升用户体验。 |
| 数据同步 | 提供一个安全的数据通道，及时、准确将服务器端的业务数据，下推到手机终端 App，用户可以预先定义实时发布所需要的各种配置数据。不需要每次手工输入，提升效率，降低出错可能性。 |
| 移动监控 | 提供移动App卡顿、闪退、webview分析等移动客户端异常监控能力，支持查看趋势分析、异常详情等。 |
| 移动分析 | 提供用户概览、渠道分析、版本分析、活跃度分析、留存分析、流失与回流分析、页面分析等多种移动应用专业分析指标，并支持自定义事件上报和自定义大盘，帮您用数据实现产品、运营、推广的决策。 |
| 日志管理 | 提供应用异常日志、接口调用日志等日志数据存储和查询服务 |
| 数据埋点 | 针对APP业务场景采集用户行为日志 |
| API元数据 | API接入的请求参数、响应参数、基础校验规则 |
| 字典元数据 | 应用程序需要使用到的选项字典、 |
| 应用元数据 | 应用程序全局配置的元数据管理，系统初始化数据、缺省配置、各种开关配置 |
| 响应码元数据 | 一笔业务请求发起后业务系统都会返回通讯状态和业务响应码，有些场景是需要根据响应码进行不同的展示和交互，需要集中管理 |
| 国际化元数据 | 前端产生的国际化资源由前端管理  所有引用国际化的资源都需要有缺省值，其它语言都由元数据拉取映射  后端产生的国际化由元数据统一映射，包括（API参数、字典数据、返回码、返回描述、菜单名称） |
| 小程序 | 小应用是一种基于Hybrid技术的开发模式，综合H5的开发高效、跨平台的特性和Native应用的性能优势，让开发者可以快速开发⾼性能的⻚面，提供优异的用户体验。 |
| H5容器 | 移动端 Hybrid 解决方案 SDK，提供了良好的外部扩展功能，拥有功能插件化、事件机制、JSApi 定制和 H5App 推送更新管理能力，支持多容器 |
| 浏览器内核 | 提供独立浏览器内核，提升网页打开速度，解决移动端webview使用过程中出现的android设备适配问题，支持更丰富的文件浏览和视频播放能力，优化用户的浏览体验 |
| 智慧投放 | 智慧投放组件提供 App 和小程序内的个性化广告投放能力，支持针对定向人群进行个性化广告投放，帮助 App 运营人员精准、及时触达用户，实现留存、促活、增长业务的目的。 |
| 移动安全 | 数据加密 | 支持RSA非对称加密、签名；支持国密； 按监管要求进行不同加密算法的切换扩展； 传输加密、存储加密 |
| 应用加固 | 提供Dex/SO反编译、iOS防破解、防篡改完整性保护、反调试动态运行防护、本地数据与资源文件加密等移动安全防护能力。 |
| 环境检测 | 移动安全检测SDK服务依托海量的数据计算平台和领先的AI引擎智能研判，国内最大的病毒样本库，领先的启发式病毒查杀引擎，敏锐的恶意样本感知能力，为移动端提供稳定可靠的安全能力和服务，主要服务包括病毒扫描、WiFi检测、防截屏、防录屏、防界面劫持、网址检测等。 |
| 源码混淆 |  |
| 网络安全 |  |
| APP防越狱 |  |
| 漏洞扫描 |  |
| 平台工具 | 组件平台 | 在线组件库，客户端开发时可通过平台快速获取组件配置到项目中 |
| CLI工具 | 具有交互式图形化的工具，该工具用于帮助开发者能够快速构建新项目辅助进行开发工作。 |
| 代码生成工具 | 基于数据模型和代码模板快速反转生成接口对象、后台管理系统维护功能。 |

## 平台优势

1. **统一的平台**

平台通过提供统一的开发框架、统一的运维模式、统一的安全能力以及统一的运营体系，助力移动服务开发体系化和高效化，实现全流程闭环管理。

1. **卓越的性能**

针对移动端 WebView 常见问题进行了深度优化，crash 率降低到0.15%以下，加载速度提升30%+，支持46种视频和文件格式。多年实战检验，有效提升移动端访问的稳定性、兼容性和流畅性。

1. **安全的服务**

拥有海量全面的病毒库，提供云+端的多重安全防护措施，识别潜在风险，检测移动威胁，从通讯安全、应用安全、环境安全等多维度全面覆盖客户端软件安全和数据传输安全。

1. **专业的指标**

多维度剖析，提供用户行为轨迹、页面访问深度等80多项产品分析指标的预设，确保运行数据的完整收集，提升分析的有效性和分析效率。

1. **灵活的发布**

支持按用户、按机型等多维度灰度发布形式，精准控制发布范围，保障版本顺利发布；提供热更新、离线包管理，快速修复现网 bug，提高访问速度，提升产品活跃度。

1. **丰富的组件**

丰富的即插即用的 AI、传播和场景工具，覆盖推送、分享、人脸识别、OCR 识别、智能语音等创新业务能力，丰富移动金融服务玩法，实现金融服务的跨界融合。

## 应用场景

1. **新应用开发**

基于囊括开发框架、运维系统、运营体系、安全组件的统一开发平台，为移动金融应用开发提供全生命周期的支撑和管理，协助金融机构高效率地开发支持多端协同的移动应用。

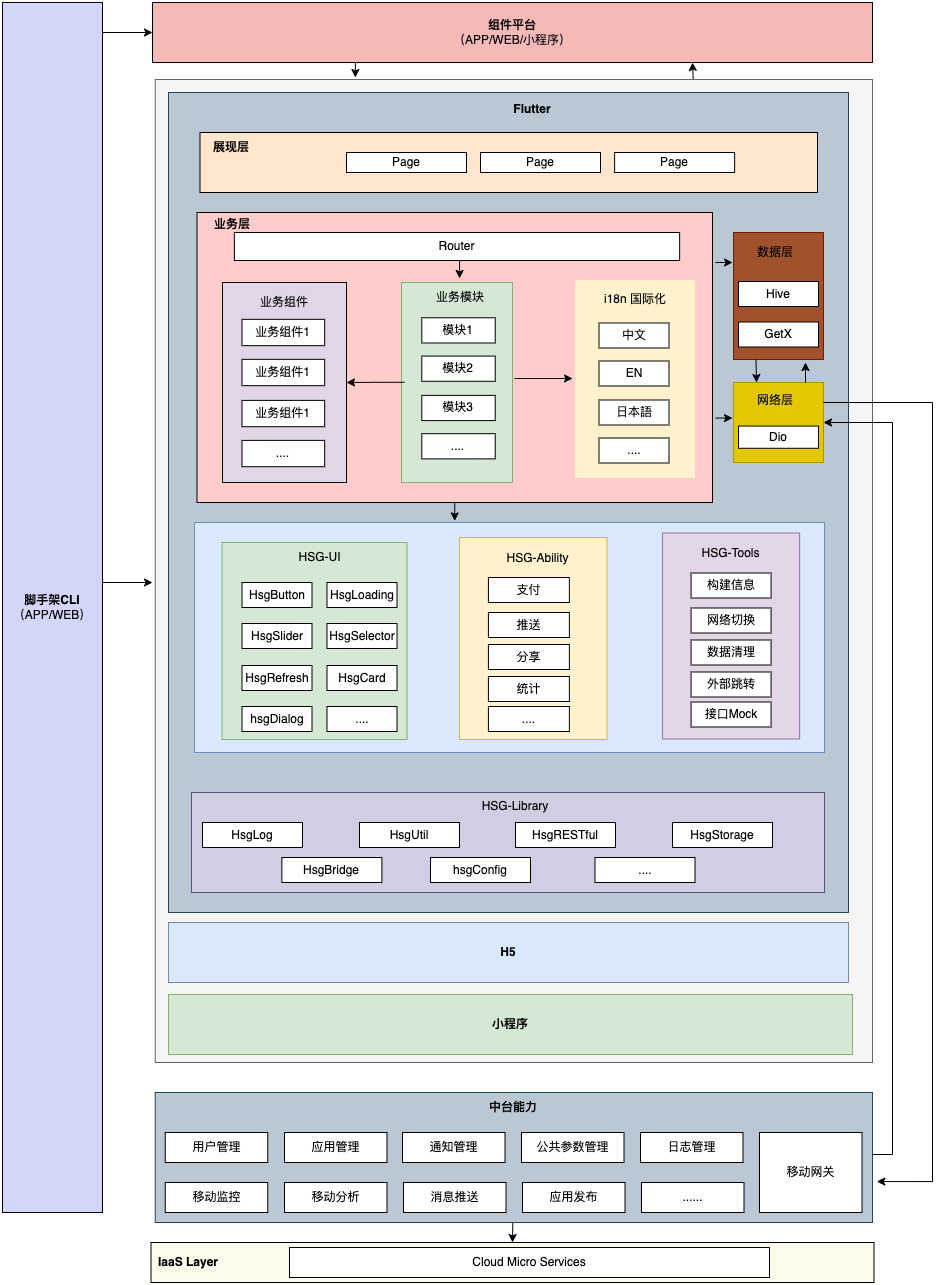
1. **存量应用集成**

平台能力可拆分作为独立的服务进行输出，您可根据需求灵活地将所需能力集成到已有的移动应用中，包括网关服务、消息推送服务，发布服务、H5 离线包、热更新服务、安全组件，统计分析服务，监控服务、传播组件、 AI 组件等。

# 概要设计

## 客户端设计

### 移动开发框架



* 平台支持4大研发框架Native（Android、iOS）开发框架、Flutter开发框架、H5开发框架、小程序开发框架。
* 平台拥有多个功能性组件，如：通讯、数据埋点、用户反馈、消息推送等。
* 平台拥有众多个UI组件，如：按钮、加载、弹框、卡片等。
* 组件平台来存储和维护管理组件
* 脚手架CLI来智能的快速构建默认配置的基准项目，自动发布和拉取组件，快速生成构建模块模板等。

#### 小程序

##### 服务概述

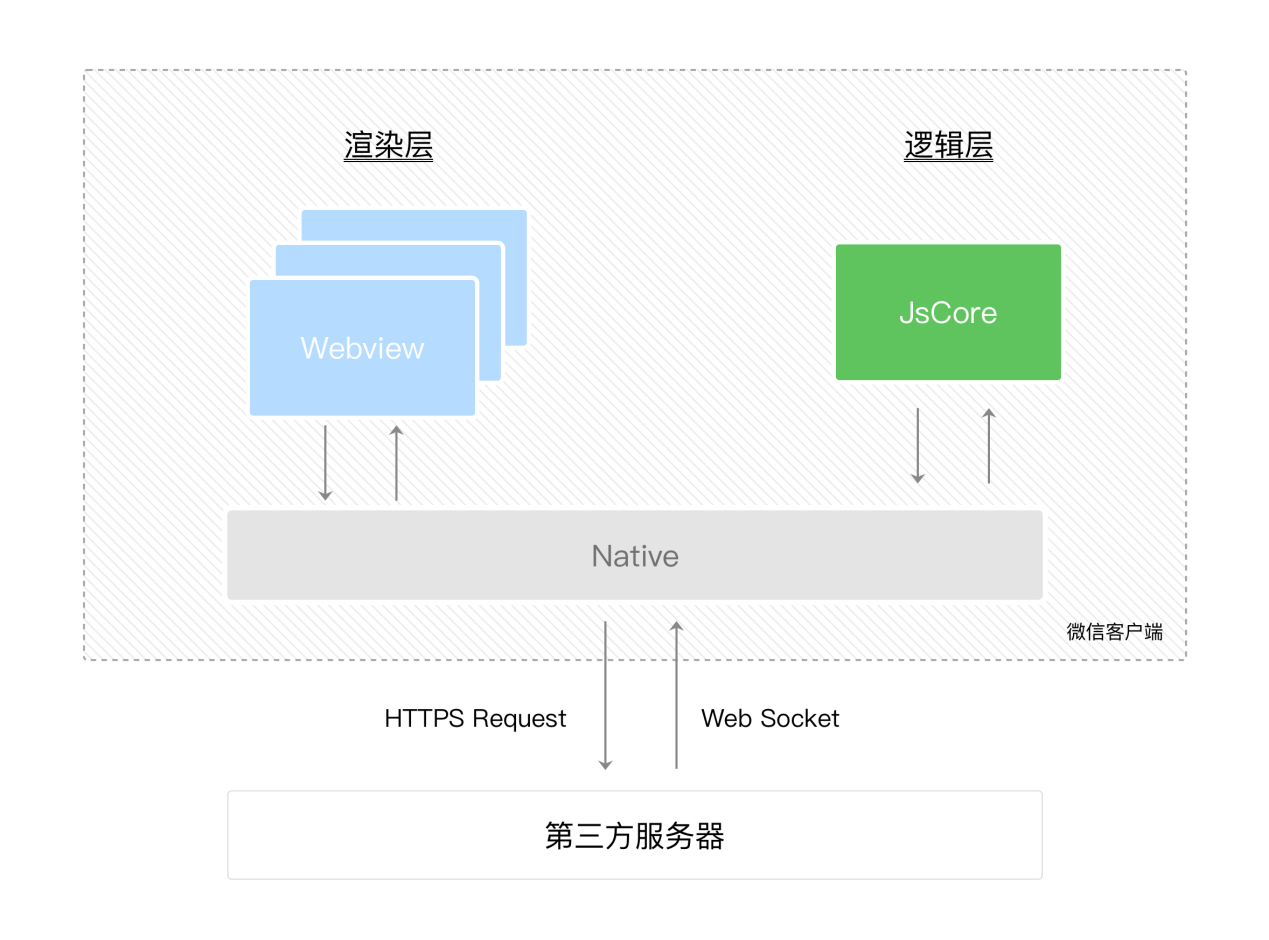
小程序易开发性、跨平台性以及 Native 性能，不仅帮助开发者实现面向自有 App 投放小程序，还可快速构建打包。

##### 应用场景

小程序不会占用用户的手机储存，开发者能够快速优化发布包大小，节省流量和存储；无需安装或卸载，随时可用，即用即走；同时，服务迭代不再受发版限制，快速发布，快速迭代。甚至，基于统一的开发标准，小程序仅需开发一次，便可快速投放至多端。

##### 系统架构和技术原理

小程序的渲染层和逻辑层分别由两个线程管理：渲染层的界面使用 WebView 进行渲染；逻辑层采用 JSCore 运行 JavaScript 代码。一个小程序存在多个界面，所以渲染层存在多个 WebView。这两个线程间的通信经由小程序 Native 侧中转，逻辑层发送网络请求也经由 Native 侧转发，小程序的通信模型下图所示。



###### 整体架构

小程序的基础库是 JavaScript 编写的，基础库提供组件和 API，处理数据绑定、组件系统、事件系统、通信系统等一系列框架逻辑，可以被注入到渲染层和逻辑层运行。在渲染层可以用各类组件组建界面的元素，在逻辑层可以用各类 API 来处理各种逻辑。

小程序的基础库主要分为：

WAWebview：小程序视图层基础库，提供视图层基础能力

WAService：小程序逻辑层基础库，提供逻辑层基础能力

#### H5容器

##### 服务概述

H5 容器组件，提供了成熟的移动端 Hybrid 解决方案，拥有丰富的内置功能，也提供了良好的外部扩展功能，具备单/多容器管理、单/多页面管理、事件透传、信息互换、埋点上报等基础能力，并提供插件定制 JSAPI。

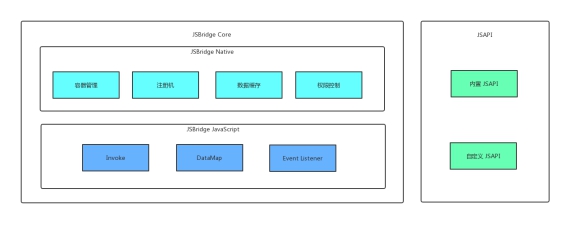
通过 H5 容器组件，可以极大的简化客户端与 H5 页面的交互流程，规范化 H5 页面的会话管理。H5 容器，可以加载 H5 页面及按照会话概念管理各个页面，内置丰富的 JSAPI、标题设置等功能，扩展业务需求。支持自定义网络库、网络通道、键盘、地理位置获取、分享等各模块。通过自定义插件，可轻松实现定制 JSAPI。

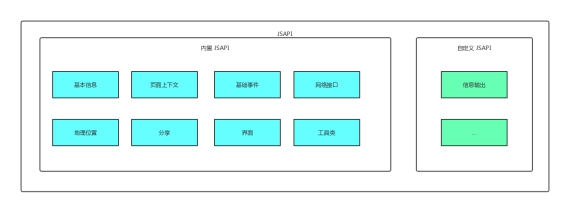
##### 应用场景

H5容器技术打通了客户端与 H5 页面之间的阻碍，可用于客户端与 H5 页面双向数据访问等。H5 容器技术，也可以应用于复杂的 H5 页面管理，完美掌控 H5 页面的状态与安全，充分保障 H5 在客户端的用户体验。

##### 产品架构和技术原理

###### 总体架构

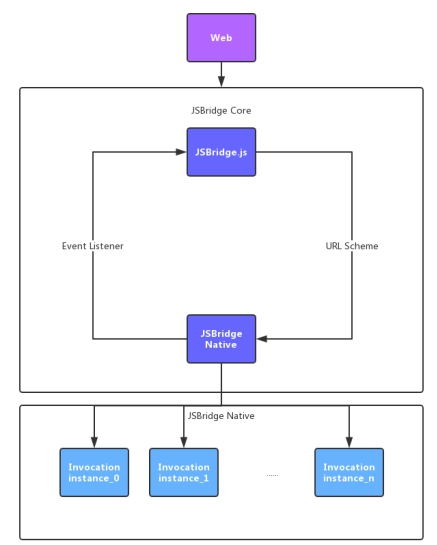




H5 容器的架构分为两个部分：JSBridge Core 和 JSAPI

1. JSBridge Core：移动端 Hybrid 方案的核心中介，JSBridge 架起了一座连接 JavaScript 和 Native Code 的通信桥梁，让双端可以互相调用。
2. JSAPI：移动端 Hybrid 方案的接口，用于在 H5 页面调用，并在客户端实现相应的处理。除了基本 JSAPI 外，还可以自定义 JSAPI，实现扩展功能。支持基础信息获取、页面控制、事件透传、地理位置获取、分享、上报、网关等能力。

###### H5 容器工作流程



H5 容器可以看成是客户端提供给 H5 页面展示的一个管理器，通过捕获 web 请求来实现一些自定义的流程注入或操作处理。整个流程的核心可分为关键的几部分：

1. H5 调用 Native 功能
2. Native 调用 H5 功能
3. 注册与权限控制
4. 多容器管理
5. 自定义 JSAPI

##### 产品优势

H5 容器组件具有以下优势：

* 强大的管理能力：
  + - 支持单/多页面，单/多容器的模式进行管理，整个流程高效规范
    - 组件集成一套成熟的辅助工具，封装常用流程，便捷开发操作
* 高稳定性与高兼容性：
  + - 接入多个腾讯系应用（腾讯手机管家、腾讯WiFi管家等），经受过亿级用户量考验。
    - 组件采取系统主流API实现，版本迭代兼容性强，对新增系统适配代价低
* 丰富的接口：
  + - 内置丰富的 JSAPI，在无需定制的情况下可完成绝大多数需求
* 可扩展性强：
  + - 支持定制 JSAPI，可根据自身需求定制接口

### 客户端组件

#### 通讯组件

##### 服务概述

通过统一封装，可以快速的接入App实现网络请求；可以灵活的切换通讯协议和通讯地址，底层网络请求库无耦合，可根据业务需求灵活更换以及灵活切换环境；可以处理通讯报文的加密。

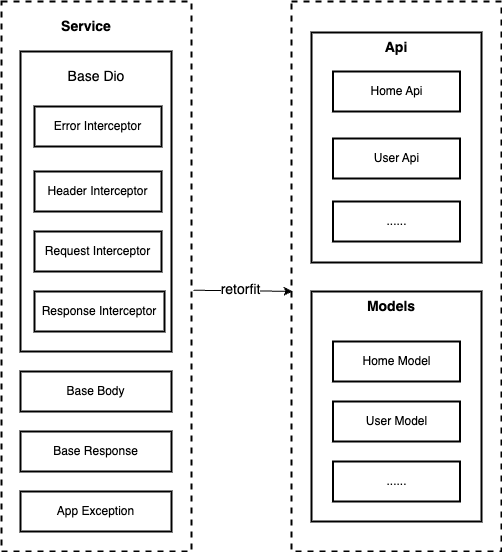
##### 应用场景

可用于应用的请求服务端来获取数据，处理报文的加密，处理多个环境切换，如，DEV、SIT、UAT、PROD。

##### 系统架构和技术原理

组件是基于Dio的封装，结合retrofit，可以快速生成实现类。

###### 整体架构



整体结构分为Service、Api、Models三层：

* Service：对dio的封装和常见错误码的封装以及基础请求体和响应体的定义。
* Api：根据dio和retorfit自动生成的api实现类。
* Models：自动生成的序列化实例类。

###### Dio的封装

dio的封装是通过重写4个拦截器，可以自由的控制请求、响应、异常。

Error Interceptor：错误拦截器，统一处理异常问题

Header Interceptor：请求头拦截器，处理请求的header，请求头数据如下

Access-Version: 2

Accept-Language: km-KH

Access-Token:

contentType: application/json; charset=utf-8

responseType: ResponseType.json

followRedirects: true

connectTimeout: 0:00:30.000000

receiveTimeout: 0:00:30.000000

Request Interceptor：请求拦截器，拦截请求数据，进行数据的二次加工，如：调整数据结构，加密等

Response Interceptor：响应拦截器，拦截响应数据，进行数据的二次加工，如：调整数据结构，加密等

###### 基础请求体

基础请全体是一个固定不变的结构，用于包装业务请求体，内容如下：

data：业务请求数据

key：应用的key

sign：应用的签名

appId：应用的Id

###### 基础响应体

基础响应体是一个固定不变的结构，用于包装业务响应体，内容如下：

system : "wallet-app-backend"

state : 1

code : "PARAM\_ILLEGAL"

message : "请求参数有误"

data : {"authId":"2022100911014910001001","effectiveTime":120}

detail : [{"field":"data.mobile","value":"null","issue":"mobile NotBlank"}]

###### API模块

API模块是每个业务模块的api的定义，是一个抽象类，可以通过retorfit

注解自动生成实体类，可以节省大量编码时间。

###### Models模块

Models模块是一些用于序列化的实例类，通过 json\_annotation 或者编辑器插件生成的实例类。

#### 存储组件

###### 服务概述

存储组件提供完整的持久化解决方案，组件提供了多种存储方式以满足不同的存储需求。目前，存储组件主要由Hive组成。

[Hive](https://links.jianshu.com/go?to=https://pub.flutter-io.cn/packages/hive" \t "/Users/lishaoying/Documents\\x/_blank) 是一个纯 Dart 编写的、基于文件存储的、轻量且功能强大的 Key-Value 型数据库。

###### 应用场景

统一存储技术，可以应用于所有涉及数据存储、数据持久化的业务。提供客户端持久化存储解决方案，底层提供多样化的存储方式以满足不同的存储需求，支持数据加密功能。

###### 产品优势

Hive 是一个轻量级、快速的键值数据库解决方案，它是跨平台的（运行在移动、桌面和 Web 上）并且是用纯 Dart 编写的。 这使它比不支持 Flutter web 的 sqflite 具有立竿见影的优势。

下图是 Hive 与其他类似数据库解决方案的基准测试图：



这是在装有 Android Q 的设备上执行的 1000 次读写操作的基准测试。

#### 元数据组件

##### 服务概述

元数据组件主要处理应用的公共的数据参数、配置等，如：

#### 国际化组件

##### 服务概述

国际化组件用于处理应用的多语言，是UI字符串以不同语言显示，可以在应用里进行语言的切换。

##### 应用场景

可应用于需要多种语言，需要对各种语言灵活配置的应用。

##### 系统架构及技术原理

应用通过调用 [Localizations.of(context,type)](https://api.flutter-io.cn/flutter/widgets/Localizations/of.html) 来引用这些对象。如果设备的语言环境变化了，Localizations widget 会自动地加载新的语言环境的值，然后重建那些使用了语言环境的 widget。

#### 数据埋点组件

##### 服务概述

* 功能按钮埋点

1. 应用的顶部tab、底部tab、内容筛选tab的点击
2. 应用内各入口，如，点击了贷款模块、点击了定期模块等
3. 应用内具体的内容点击，如点击了贷款模块里的banner、点击了贷款模块里的’如何提示额度’等
4. 应用内设置选项的点击
5. 应用内功能按钮的点击

* 主要行为埋点

1. 用户核心行为，包括登录、注册、查看贷款额度、提交贷款申请等
2. 用户其他行为，包括修改了语言、重置了密码、修改了用户名等

* 页面埋点
* 支付流程埋点
* 搜索流程埋点
* 数据埋点

1. 用户账号、用户的UDID、机型、操作系统
2. 用户登录ip、经纬度、国家、省份、城市、区县、街道等
3. 用的操作时间，如果是页面时间还记录了开始时间和结束时间
4. 用户进行了什么操作，如应用的id、页面来源、应用所在位置、应用来源模块、是否操作成功等

##### 应用场景

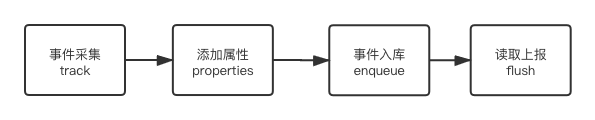
App 的开屏页、App 首页、App 首页 banner 位，还涉及需要埋点的终端，微信小程序、WAP端等多端的埋点。

##### 系统架构及技术原理

数据采集流程中主要包括事件采集、添加属性、事件入库、读取上报等流程，详细的步骤如下所示：

1. 在产品、服务转化的某些关键点，调用埋点相关接口采集事件；
2. 获取有意义的属性丰富该事件，保证数据的广度与深度；
3. 数据采集完成，转换成标准 JSON 数据格式，以队列的形式存储到 SDK 的数据库内；
4. 定时读取数据库中的数据，封装请求并上报数据，并在上报成功后，删除数据库内存储的已上报数据。

整体流程如图所示：



代码埋点的实现原理比较简单，主要是初始化 SDK 之后，在某个事件发生时调用相关接口，将触发的事件和属性保存到数据模型中。并将数据模型转化为 JSON 串，存储到本地数据库中。然后，按照发送策略将数据发送到指定的服务端。例如：我们想统计 App 里面某个按钮的点击次数，可以在这个按钮对应的点击方法里面调用 SDK 提供的接口来采集事件。

### UI组件库

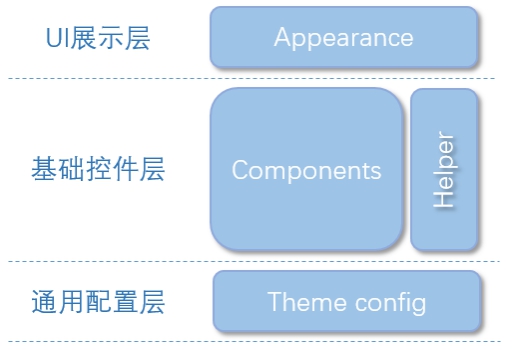
#### 服务概述

UI组件库用于辅助快速搭建一个具备基本设计还原效果的 Android/iOS 项目，同时利用自身提供的丰富控件及兼容处理，让开发者能专注于业务需求而无需耗费精力在基础代码的设计上。不管是新项目的创建，或是已有项目的维护，均可使开发效率和项目质量得到大幅度提升。

#### 应用场景

UI组件库可用于从0快速搭建应用UI界面，按照说明进行基础配置，即可使用丰富控件搭建UI界面展示。也适用于已搭建好的应用，摘选控件库中所需控件直接应用于App中。

#### 系统架构及技术原理



QMUI为Android/iOS应用UI展示提供统一展示样式，提供丰富控件及通用简便帮助类。组件依靠 theme 进行样式管理，可实现全局控制的效果，即通用配置层theme一经修改，全局同步更新。

#### 产品优势

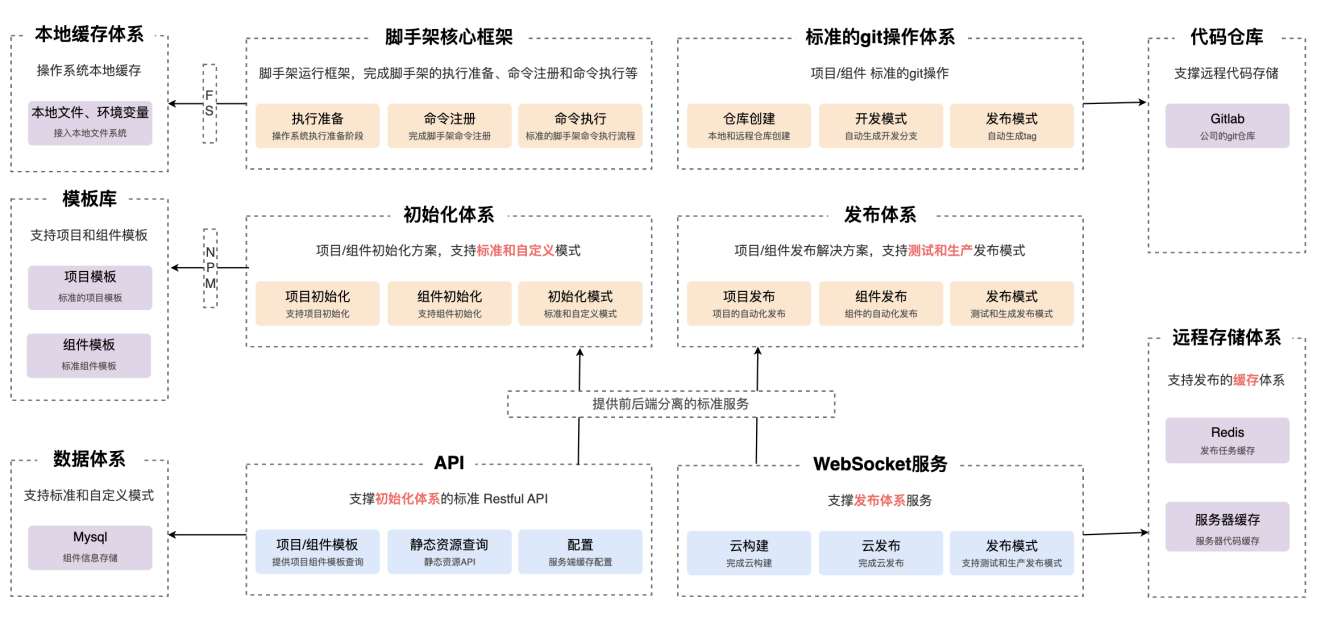
* 全局 UI 配置：只需要修改一份配置表就可以调整 App 的全局样式，包括组件颜色、导航栏、对话框、列表等。一处修改，全局生效。
* 丰富的 UI 控件：提供丰富常用的 UI 控件，例如 BottomSheet、Tab、圆角 ImageView、下拉刷新等，使用方便灵活，并且支持自定义控件的样式。
* 高效的工具方法：提供高效的工具方法，包括设备信息、屏幕信息、键盘管理、状态栏管理等，可以解决各种常见场景并大幅度提升开发效率。

### 脚手架CLI（App/Web）

#### 服务概述

脚手架CLI可以帮助我们完成项目的创建，如：标准模板的创建、自定义规则的创建、创建组件、自动安装等，还可以帮忙我们完成项目的发布，如：项目自动化、项目的自动发布、组件的自动发布等，可以显著的提升前端的研发效能，实现自动化、标准化、数据化。

#### 系统架构及技术原理



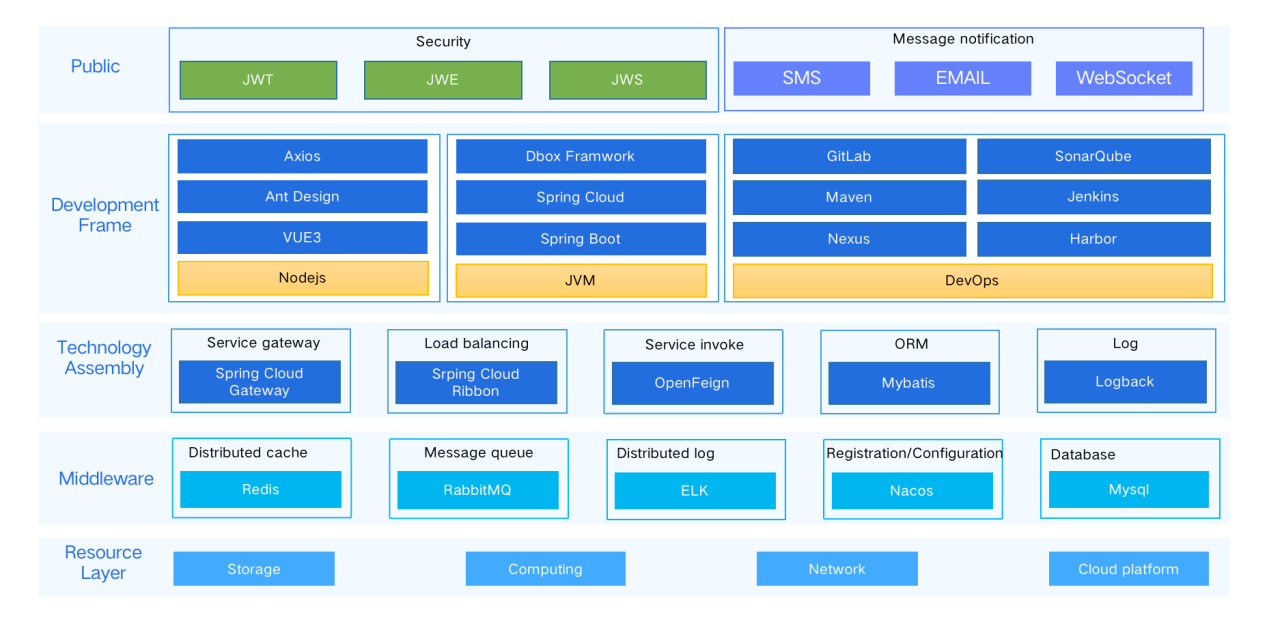
* **脚手架核心框架**：使用原生Node.js和Lerna多包管理工具实现，完成操作系统执行准备、命令注册、命令执行。
* **标准的git操作体系**：由脚手架完成统一标准的git操作，完成本地和远程仓库的创建、自动生成分支、自动生成生产tag等。
* **初始化体系**：由脚手架完成基准项目的初始化、组件的初始化，项目的初始化可支持模式的选择等。
* **发布体系**：由脚手架完成项目的自动化发布、组件的自动化发布，可支持选择测试模式和生产模式等。
* **API**：给初始化体系提供服务，提供项目和组件的查询、静态资源的查询已经服务器的缓存配置。
* **WebSocket服务**：给发布体系提供服务，可完成项目的云构建和云发布等
* **代码仓库**：使用公司的gitlab私有库，来存储项目、组件的代码，支撑标准git操作体系。
* **远程存储体系**：使用Redis或服务器缓存来支撑websocket服务。
* **本地缓存体系**：使用操作系统的文件系统和环境变量来实现缓存脚手架的基本操作等，使脚手架运行更加流畅快速。
* **模板库**：使用NPM来存储项目和组件模板来支持脚手架的初始化体系。
* **数据体系**：使用mysql来存储项目、组件和相关配置信息，来支撑API给初始化体系提供服务。

### 组件平台

#### 服务概述

## 中台设计

### 技术架构



* 微服务开发技术

系统采用主流成熟的Spring Cloud微服务开发框架，快速构建分布式系统，集成生态组件包括：配置中心、服务发现注册、断路器、消息总线、全局锁、负载均衡。

使用Spring Cloud的开发者可以快速的启动服务或构建应用、同时能够快速和云平台资源进行对接。

* 管理系统WEB开发技术

系统Portal Web使用VUE3开发，基于标准 HTML、CSS 和 JavaScript 构建，易学易用，同时有着丰富的、可渐进式集成的生态系统，可以根据应用规模在库和框架间切换自如。

* 依赖中间件

系统目前依赖的都是主流的开源中间件，如需要跟云厂商进行集成都可平滑切换到云厂商的中间件产品

数据库：Mysql

消息队列：Rabbitmq

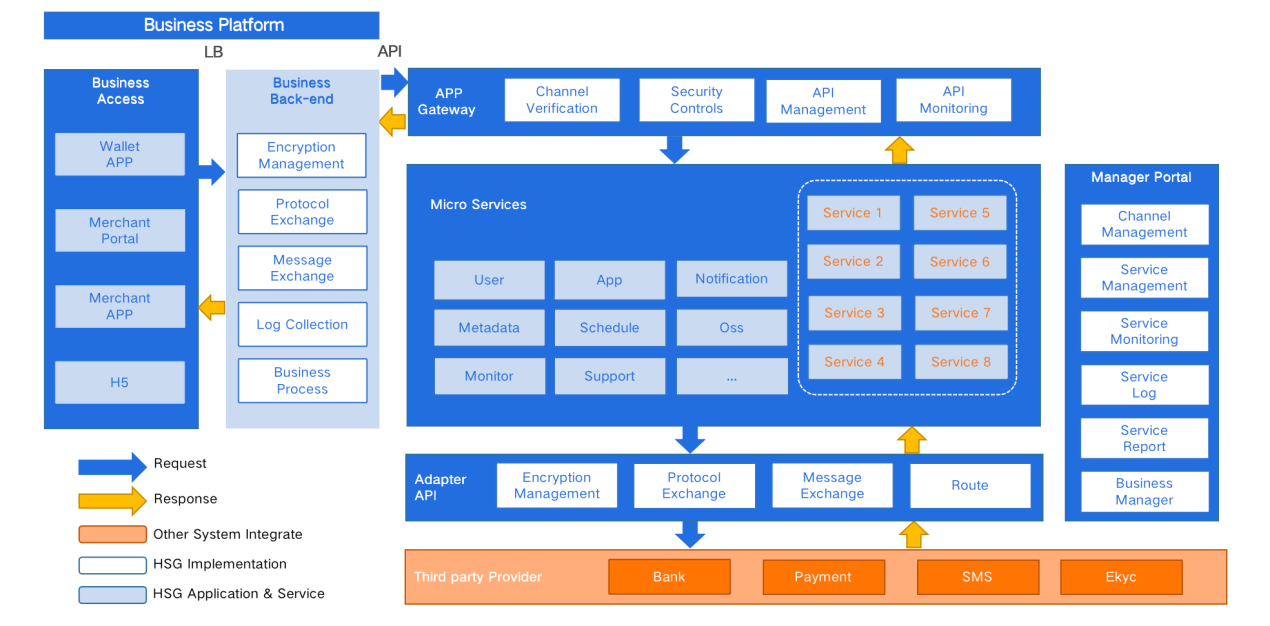
缓存：Redis

对象存储：Minio

* 技术能力输出

|  |  |  |
| --- | --- | --- |
| **技术能力** | **参照技术标准** | **备注** |
| 微服务底座 | spring-boot-starter spring-boot-starter-web spring-cloud-starter | spring-boot-starter-web 不适用Reactive Stack在spring-cloud-gateway中不能使用 |
| 微服务调用 | spring-cloud-starter-openfeign spring-cloud-netflix-ribbon | 声明式微服务调用 |
| 消息事件 | spring-cloud-bus spring-cloud-stream | 全局广播消息(缓存刷新)使用spring-cloud-bus 其它业务消息使用spring-cloud-stream 定义topic发送 |
| 分布式缓存 | spring-data-redis jetcache-starter-redis | jetcache支持本地缓存和远程缓存组合使用，支持方法注解缓存 |
| 数据库访问 | ORM框架：mybatis 数据库连接池：druid-spring-boot-starter crud接口封装：tk.mybatis 分页：pagehelper-spring-boot-starter 分库分表及加密：自研框架 | 默认支持mysql，可扩展支持其它数据库如oracle |
| 日志 | slf4j、logback | 可集成apm 的链路标识做为日志traceid，如skywalking 、elastic apm |
| 注解校验 | validation-api、hibernate-validator | @Null 验证对象是否为null @NotNull 验证对象是否不为null, 无法查检长度为0的字符串 @NotBlank 检查约束字符串是不是Null还有被Trim的长度是否大于0,只对字符串,且会去掉前后空格. @NotEmpty 检查约束元素是否为NULL或者是EMPTY @Size(min=, max=) 验证对象（Array,Collection,Map,String）长度是否在给定的范围之内 @Length(min=, max=) Validates that the annotated string is between min and max included. |
| 工具包 | lombok、hutool、fastjson、apache-commons |  |
| 开发工具 | idea、maven、git、nexus、代码生成插件 |  |
| 网关 | spring-cloud-starter-gateway | 基于spring-cloud-gateway开发增强，Netty 作为 web 容器、WebFlux非阻塞异步 |
| 配置中心 | spring-cloud-starter-alibaba-nacos-config |  |
| 注册中心 | spring-cloud-starter-alibaba-nacos-discovery | 支持切换到eureka |
| 熔断限流 | 框架自研 | 熔断支持可切换到Hystrix，限流也可以用sentinel限流 |
| 分布式缓存 | Redis | 可集成云厂商服务 |
| 分布式消息 | RabbitMQ | 可集成云厂商服务 |
| 链路跟踪监控 | Grafana 、tempo | Grafana 生态开源 |
| 日志平台 | Grafana 、Loki、promtail | Grafana 生态开源 |
| 影像存储 | MinIO |  |
| CI/CD | Gitlab、jenkins、docker |  |
| 监控 | Grafana、Prometheus |  |
| 代码质量扫描 | Sonar |  |
| API文档、用例、自动化接口测试 | Yapi |  |
| 内部微服务请求数据模型 | Request<D>、Response<D> | 方便数据加工处理(加解密） 方便异常处理(异常识别与互转) 方便接口监控(解析) 方便灰度控制(同一标准可全链路传递做全链路灰度） |
| 外部开放接口请求数据模型 | ApiRequest<D>、ApiResponse<D> |  |
| 管理web服务请求数据模型 | Request<D>、WebResponse<D> |  |
| 分页数据模型 | PageHelper | 分页数据查询返回标准结构体 |
| 统一返回码设计 | CommonReturnCode、ReturnCode | 返回码支持占位符动态替换、错误归类方便统计 |
| 统一异常设计 | BizException | 异常统一处理、和Response 、ReturnCode 互转 |
| 统一请求会话设计 | RequestContext、RequestContextHolder | 线程级数据存储共享 |
| 统一的服务返回状态设计 | CommonStateEnum | 业务监控识别 |
| 注解使用 | JSR303 @Validated({Group.class}) | 简化代码,可扩展标签 |
| 接口文档发布 | Java Doc 、EasyYapi | 1.强约束规范代码有完善的注释，提高代码可读性 2.无需手动编写接口文档节省人力资源  3.在线接口文档保持最新状态方便开发、测试 |
| 接口熔断实现 | FallbackFactory | 快速失败且不会抛出异常给业务代码，直接返回标准的Response 交易接口全链路熔断时需要设置state为超时，让调用方做补偿或者查询机制 |
| 统一日志输出格式 | %date{yyyy-MM-dd HH:mm:ss.SSS}|%-5level|%logTraceId|%class.%method:%line|%extJson|%sensitiveMsg%n |  |
| 统一日志跟踪号 | LogTraceIdConverter | 线程级唯一日志id配合链路工具实现全链路日志id |
| 统一日志脱敏 | SensitiveMessageConverter、@Desensitized | 基于对象标记进行统一处理 |
| 统一日志埋点域 | ExtJsonConverter | json埋点数据，可由logstash拆分并进行数据分析和可视化 |
| 集成字典缓存 | FeignCacheService | 可实现业务参数的灵活配置获取 |
| 元数据 |  |  |
| 集成mock配置缓存 | FeignCacheService | 1.通过配置实现(生产prod不支持)，无需硬编码对业务系统无侵入 2.开发阶段减少对环境的依赖，屏蔽环境不稳定对开发的影响提交效率  3.能够完美解决和外部接口对接时环境的问题  4.支持动态配置各种场景的mock返回数据，更好的支持测试 |
| 统一异常处理 | ExceptionHandle | 开发管理好异常返回码和返回状态即可，代码变得规范简单 |
| 统一服务调用日志记录 | FeignLogger | 统一处理应用系统无需手动打日志 |
| 统一返回码转换 | FeignResponseDecoder | 根据返回码配置表进行转换,业务系统不要随意使用 只在最外网关层使用 |
| 统一国际化 | 后端国际化（返回码描述、异常描述、数据字典、后端菜单） | 前后端分离国际化分离原则，谁产生数据谁配置，代码中和配置中默认是英文,，当需要其它语言时配置下语言包即可，不配置使用默认也不影响功能使用，提供工具生成国际化配置模板检查代码编写是否规范 |
| 统一接口调用日志埋点监控 | CallInterfaceInterceptor | 日志工具可视化展示服务的运行数据 |
| 分库分表 | ShardingMapper | 性能扩展、需要分片键配置路由 |
| 数据加密存储 | Shardingjdbc | 安全、监管、 |
| 数据库操作监控 | Druid | 性能 |
| 数据分页插件 | Pagehelper、PageInfo | 分页查询 |
| 支持多数据库 | MySQL,Oracle |  |
| Mapper代码生成 | Mapper\xml\Entity | 自研idea插件 |
| 定时调度快速开发 | elastic-job | 任务处理、串行、并行、动态调整运行参数 |
| 管理系统快速开发 | @Dict @ExtSuffix @NumberConvert @ParamsDecrypted @Permission @SysLog @TokenCheck @Excel @Add @Edit | 统一权限控制、日志记录、标准数据格式、校验分组、标准模板生成 |
| idea插件代码生成 web在线代码生成 | 实现分页查询、下载 实现新增、删除、详情、编辑 实现批量删除、导入 实现查询条件、显示、字典转换 实现数字金额单位转换 实现脱敏转换(后台)  实现全局工具条 实现行工具条  实现汇总数据行 最小化数据更新处理 | 自研idea插件 |

### 应用架构



* 设计原则

1. 单一职责，服务按业务模块划分，清晰独立
2. 服务只能垂直访问对应的数据库，如果要跨数据库访问需要调用对应微服务接口来获取
3. 组件化解耦
4. 最小化原则
5. 无状态设计
6. 参数化配置

* 系统包含下面几个部分组成

1. Business Access(APP/Portal)

产品端，提供给用户进行业务操作的终端，可以是手机APP访问，浏览器网页访问。

1. Business Back-end

受理业务请求的后端服务，主要做安全对接和API编排。

1. API Gateway

标准技术组件，提供API治理的功能。

1. Micro Services

平台基础微服务群：提供平台基础能力，当前划分有8个

User：用户服务，提供用户登录注册功能和用户标签数据

App：应用服务，提供应用管理、版本管理、版本更新维护等服务

Notification：消息服务，提供系统公告，消息管理、消息推送等服务

Metadata：元数据服务，提供字典、响应码、参数、配置等数据服务

Schedule：调度任务

Oss: 对象存储，提供图片视频文档存储服务

Support：公共业务支撑，提供移动端相关的业务组件功能

Monitor：应用控制、数据分析服务

业务扩展微服务群：提供系统业务处理的逻辑实现，按实际情况进行扩展开发

1. Adapter Api

与第三方系统对接的模块，主要负责交易路由、报文转换、与第三方通道通讯。如核心系统对接则会有对应的Core-Api

1. Management Portal

整个系统的后台管理，提供权限管理、功能配置，数据查询，业务配置等功能。

### 用户中心

#### 用户管理

##### 功能概述

用户管理包括基础的用户信息维护和查询，用户数据是识别和使用移动应用最基础的核心数据，所有功能都围绕用户来展开。

用户通过自助注册或后台注册功能完成用户信息的录入。

用户通过登录功能来使用应用提供的业务功能。

用户登录注册支持标准的账户密码模型，设计中预留扩展不同方式的登录模型，包括但不限于 两步验证登录、手机验证登录、第三方关联登录。

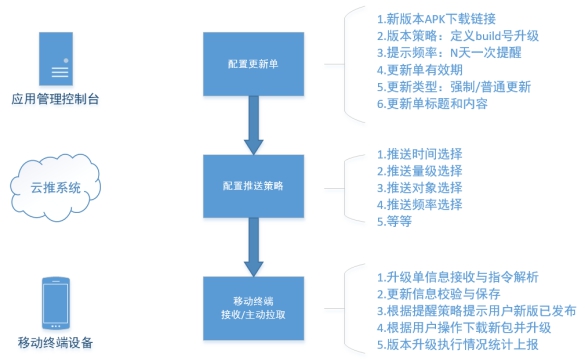
用户管理还支持安全设置、冻结账户、找回账户、主动销户等完整流程的功能。

##### 系统架构及技术原理

用户管理系统架构分为三个部分：客户端业务组件、业务中台、控制台

* 客户端业务组件：指用户注册、登录、找回密码、用户信息查询与维护、用户安全设置
* 用户中心：指提供标准用户数据维护和查询接口的微服务中心。
* 控制台：指web管理页面，主要提供查询用户信息和设置功能。

具体架构流程图如下：

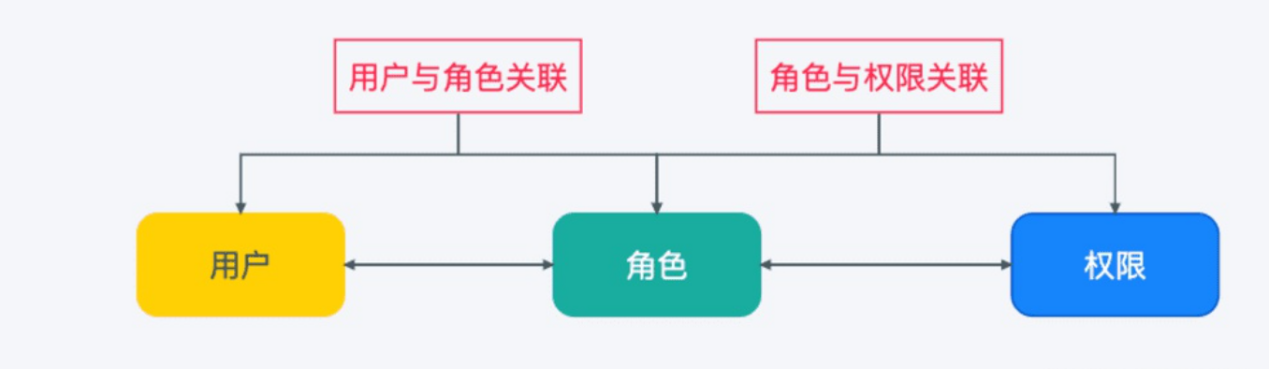


##### 产品优势

* 简洁易懂，基于通用用户信息模型不冗余差异化的信息
* 高效，通过多维度快速检索用户
* 信息安全，安全相关的数据进行加盐加签具有不可逆特性，对敏感数据进行加密存储
* 业务安全
* 灵活扩展
* 应用发布组件提供从控制台配单到终端回调APK升级的全过程，从升级策略、推送策略到发布策略，均实现可按需自定义配置。实现了新版本发布功能的高灵活性。
* 发布策略可选择发布测试任务、灰度任务、正式任务，满足开发测试过程、小范围灰度检验的需求，可在测试通过且灰度发布正常之后发布正式任务，实现发布功能的高稳定性。
* 依赖云推系统的统一推送配置，实现高到达率和高及时性。同时统一实现统一回收数据，实现对升级单下发情况和终端执行情况的数据监控。
* 终端SDK通过实时监控+定期检测的方式实现对升级单的高执行率，通过对升级单的二次检测，确保升级单的有效性，即云端停单后，终端缓存的升级单也同时失效。

#### 角色管理

#### 权限管理



### 应用管理

### 移动发布

#### 应用发布

##### 服务概述

应用发布组件终端SDK用于实现终端App的版本升级过程，当云端发布新版本的时候，SDK可实现实时监控云端下发的升级单，对监听到的升级单做出判断和处理，完成从监听、接收、判断、到升级回调的过程。

应用发布控制台包含母包管理、发布管理功能，可以清晰管理各版本应用安装包的上传及下载，同时可对已添加的应用包进行发布并管理各发布任务。

组件提供了客户端SDK，您可以轻松将应用版本升级的能力集成到客户端。在控制台上传应用包，进行版本管理。同时可根据不同的发布策略进行测试发布、灰度发布、正式发布等。客户端收到推送信息再去拉取新的发布包进行升级。

##### 应用场景

应用发布组件可应用于发布App新版本供终端进行升级，可通过下发测试任务、灰度任务依次进行功能验证后正式发布升级任务，终端SDK自动检测或在用户手动检测到新版本后可完成应用升级并回传数据进行任务闭环。

* 测试任务：用于测试阶段验证应用升级发布流程是否正常。
* 灰度发布：当测试验证完成后，可以通过自定义部分用户下发规则来做小规模发布（比如内部员工）以验证新包的功能是否达到预期。
* 正式发布：当测试与灰度发布流程均正常完成时，可进行正式任务发布，发布任务可以选择有效时间窗，选择定向发布人群，符合预期情况下可进行全量推送。

##### 系统架构及技术原理

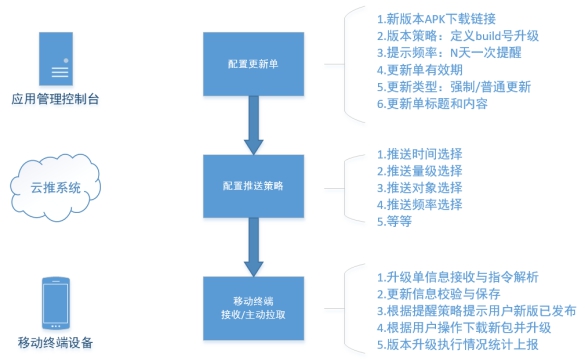
应用发布系统架构分为三个部分：控制台、云推系统、终端SDK

1.控制台：指web端页面，用于上传新版本应用APK包，并配置升级规则和下发范围

2.云推系统：指后台推送服务，用于将配置好的升级单下发到对应终端，同时回收终端升级流程上报数据

3.终端SDK：指提供的升级组件aar包，将该组件集成到客户端后进行简单配置，即可在客户端实现版本升级流程，同时上报终端升级流程数据

具体架构流程图如下：



##### 产品优势

* 应用发布组件提供从控制台配单到终端回调APK升级的全过程，从升级策略、推送策略到发布策略，均实现可按需自定义配置。实现了新版本发布功能的高灵活性。
* 发布策略可选择发布测试任务、灰度任务、正式任务，满足开发测试过程、小范围灰度检验的需求，可在测试通过且灰度发布正常之后发布正式任务，实现发布功能的高稳定性。
* 依赖云推系统的统一推送配置，实现高到达率和高及时性。同时统一实现统一回收数据，实现对升级单下发情况和终端执行情况的数据监控。
* 终端SDK通过实时监控+定期检测的方式实现对升级单的高执行率，通过对升级单的二次检测，确保升级单的有效性，即云端停单后，终端缓存的升级单也同时失效。

#### 离线包

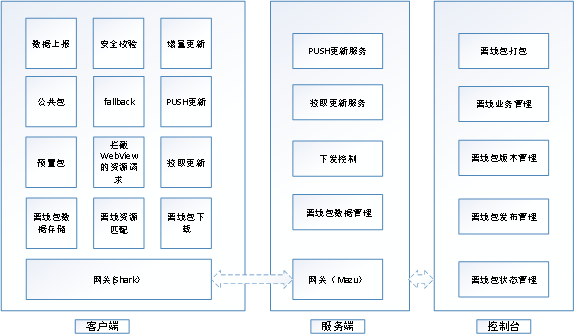
##### 服务概述

通过的离线包组件，可以将网页的静态资源（html、js、css、图片等）缓存在本地，当用户访问H5页面的时候，这些资源都不需要从服务器即时下载，离线包组件会拦截WebView的资源请求并直接读取本地的离线资源，达到“瞬间秒开”的效果。控制台实现了离线包的自动打包、状态管理和发布功能，支持推拉结合的更新方式及增量更新机制，可按多种维度进行灰度下发，且能便捷地按H5的版本采集灰度数据，对不同版本的业务数据进行对比分析。对于重要业务（如一级页面），还可以直接将离线包预置在发布包中，首次进入页面无需下载即可使用，大幅提升H5业务的用户体验。当本地离线包没有准备好时，则直接访问在线资源，不影响正常使用。

##### 应用场景

离线包技术应用于H5业务，将网页的静态资源缓存在本地，拦截WebView的资源请求并直接读取本地的离线资源，免去了从服务器即时下载资源的耗时，达到“瞬间秒开”的效果，可大幅提升H5业务的用户体验。

##### 系统架构及技术原理



1. 客户端：客户端的核心是拦截WebView的资源请求并根据请求的资源的URL匹配本地的离线资源，如果存在与之匹配的离线资源则直接读取离线资源，否则就通过WebView请求在线资源。可通过拉取和推送两种方式对离线包进行更新。
2. 服务端：服务端实现了离线包的数据管理，提供离线包更新服务，并支持按多种条件发布离线包。支持按平台（iOS、Android）、系统版本、客户端版本、时间窗、用户量级、地域、用户标签、设备ID等条件进行灰度验证，验证通过再进行全量发布。
3. 控件台：控制台包含离线包的打包服务、离线业务管理、离线包版本管理和发布管理等。

##### 产品优势

离线包组件具有以下优势：

1. 离线秒开：离线包支持html、js、css和图片等静态资源。打开页面时自动匹配和读取本地离线资源，无需即时下载，达到“瞬间秒开”的效果。
2. 便捷的离线包管理平台：支持创建离线包业务、自动打包、状态管理、查看离线包详情、自动生成增量补丁和灰度发布等完善的功能。
3. 支持推拉结合更新离线包：实现了推送更新，不依赖用户操作，极大提升新版本的触达速度和覆盖率。
4. 支持增量更新：使用了bsdiff技术实现增量更新，并做了深度优化，极大减小更新包的大小。
5. 支持多维度下发：借助云推平台，可按平台（ios、Android）、系统版本、客户端版本、时间窗、用户量级、地区、指定用户账号、设备ID等多种条件精准地进行灰度下发。
6. 便捷地按H5的版本采集数据：H5可设定页面标识，通过jsapi上报数据时，自动带上页面标识及离线包ID和版本信息，可方便地对不同版本的数据进行对比分析。
7. 批量上传，一键发布：应用于多业务下的离线包批量上传，上传后可一键将同批次上传离线包生成发布任务，避免了多番选择指定离线包的繁琐操作步骤。

### 移动推送

#### 消息推送

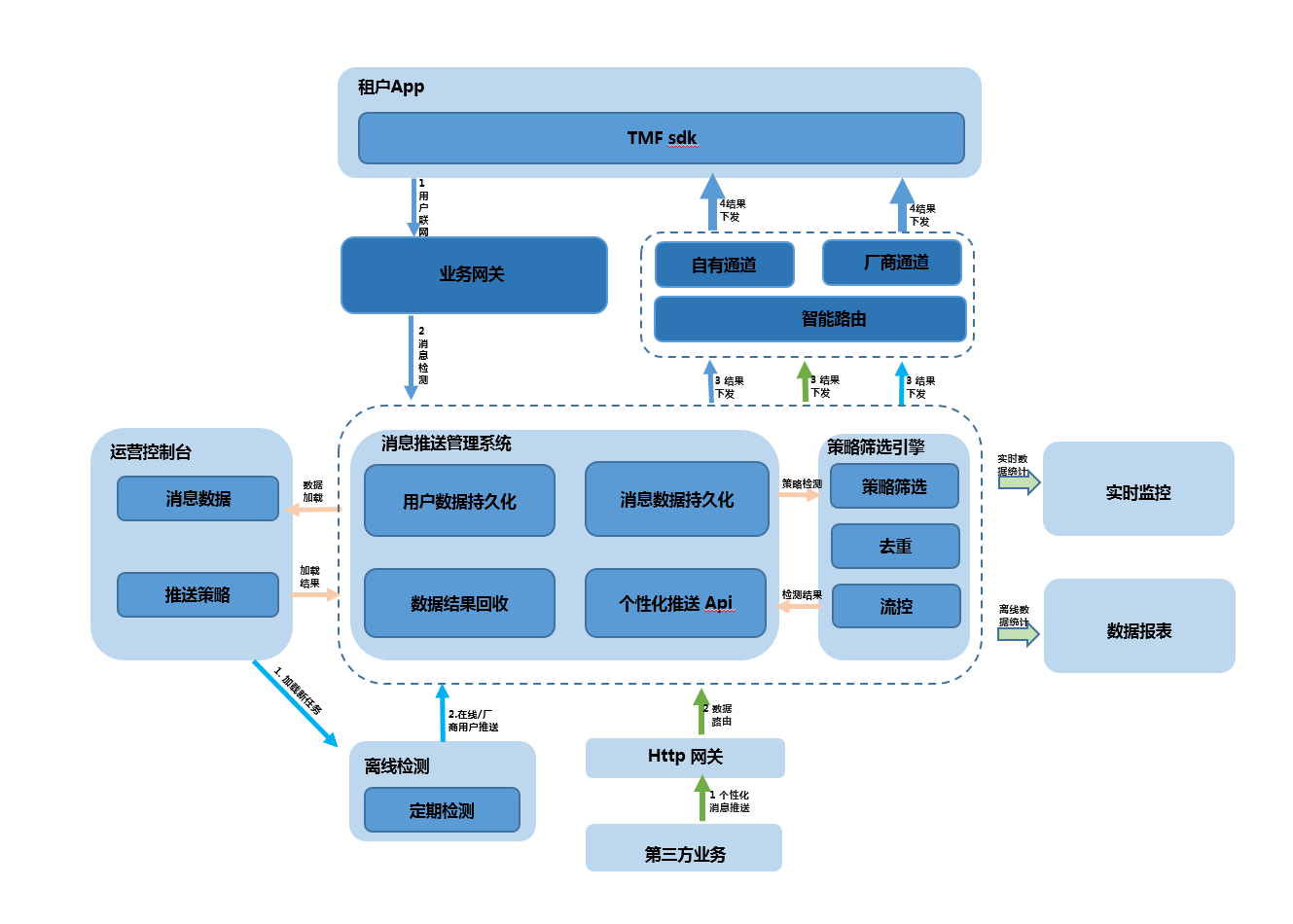
##### 服务概述

消息推送组件提供专业的移动消息推送方案，针对不同的场景推出多种推送类型，满足您的个性化推送需求。为了提升推送的到达率，集成了华为、小米等厂商的推送功能，在提供控制台快速推送能力的同时，也提供了服务端接入方案，方便用户快速集成移动终端推送功能，与App用户保持互动，从而有效地提高用户留存率，提升用户体验。

##### 应用场景

1. 唤醒沉睡用户
2. 快速推送新闻内容、促销活动、产品信息、版本更新提醒、订单状态提醒等通知栏信息

##### 系统架构及技术原理



技术原理: 消息推送提供控制台和个性化推送Api两种下发方案。控制台提供多维策略配置，针对单一消息数据，支持多个任务配置，满足产品风险控制需求。个性化Api提供业务实时推送接口，支持单个账户，设备ID推送。消息数据下发支持自有通道和厂商通道，优先厂商通道下发，厂商通道无效则走自有通道。消息推送保存用户的消息推送记录，避免重复下发打扰用户。

##### 产品优势

1. 操作简单：控制台提供推送界面设置，无须编写代码，即可满足简单的推送需求。同时，也提供服务端API接入方案，满足更为复杂的业务需求。
2. 个性化推送：

* 支持针对单个用户做测试推送和向自定义用户群做灰度推送
* 支持按照机型，账号，设备id，App版本，时间窗口，操作系统版本，地区，自定义标签等多个维度推送个性化消息。
* 支持定义消息有效期，确保消息最终送达目标用户。

1. 智能路由：基于用户信息，智能选择厂商和自有通道下发消息
2. 分批推送：对于同一个消息，可以分批次下发，满足产品自测，灰度，正式投放多阶段风控需求
3. 高速率：每分钟消息推送数量不低于1000万
4. 高到达率：支持根据App用户名发送消息，不受登录设备限制，保证消息准确送达。在线消息平均到达率：不低于99%
5. 低延迟：平均每条推送消息延迟时间：少于1.5秒。
6. 反馈及时：提供消息回执，实时展示推送反馈数据，智能分析消息送达率，打开率，明确推送效果。
7. 数据传输安全：1）厂商通道数据通过Https方式进行安全通信保证数据安全；2）自有长连接通道使用加密信道进行通信。

#### 数据同步

##### 服务概述

数据同步是的核心基础组件，为了解决产品运营多变的场景和产品运营风险控制需求，基于 TCP+SSL 的安全通道能力，提供了模板化定义数据，强大的云端控制能力，数据快速触达用户和数据下 发执行实时监控能力。

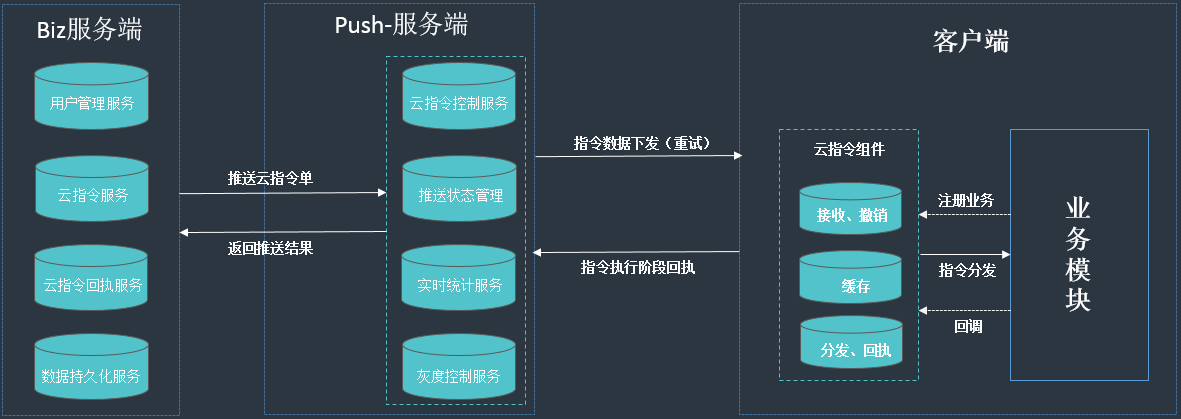
通过安全的数据通道，准确及时地将服务器端业务数据同步至手机终端App。帮助客户解决客户端重复调⽤请求、客户端感知不及时、客户端获取不同业务的数据时需多次调用请求的问题。

数据同步系统为在线App提供了及时升级服务，配置更新通知，消息和活动推送服务以及业务逻辑的控制策略等能力。其本质是从云端向终端下发数据，终端各业务插件拿到云端的数据后执行相应的操作。

组件提供动态配置信息的全设备推送的能力，将 App 功能开关、动态参数、动态配置等信息实时推送至指定客户端，批量动态地改变 App 在运行期间的业务参数、业务配置等。提供安全数据通道，在线 App 可实时接收推送数据。提供数据持久化能力，App 可在上线后接收离线期间的相关数据。

##### 应用场景

##### 系统架构及技术原理



##### 产品优势

基于数据同步，可实现应用发布、热修复、tips通知栏、离线包等多种类型业务的个性化推送。同时也可以通过强大的自定义命令字配置完成各类指令设置。

* 强大的云控能力 提供丰富的下发策略，下发量控制能力。
* 支持撤销回滚下发到本地的指令数据。
* 数据多样化支持 数据格式模板化管理
* 数据省流，快速下发，数据下发执行状况实时监控
* 省流：多个任务数据合并推送；推送过的数据不会再推送
* 快速下发：推拉结合，业务主动触发拉取云端数据；业务建立链接时，主动推送云端数据；业务主动调用接口，用户连接保持时，立刻推送，用户不在线时，会根据接口时间持久化，在有效期内下 一次用户上线时主动推送。
* 数据下发阶段和时间监控

### 多媒体

## 安全设计

### 环境安全

#### 概述

据工信部发布的《2018年10月通信业主要指标完成情况》中的数据表明，截至今年10月，移动互联网用户总数超过13.8亿户。智能设备爆发式增长,存储着的个人核心隐私信息越来越多，价值随之越来越高，是不法分子紧盯着的目标。

手机病毒的牟利性使得病毒制造者趋之若鹜，制造各种手机病毒。2018年，Android新增病毒包达800.62万个，其中支付类病毒包新增近5.90万个。Android手机病毒感染用户数近1.13亿，其中支付类病毒感染用户数近394.21万。2018年，腾讯手机管家共查杀病毒达8.12亿次。手机病毒来源渠道主要以手机资源站、软件捆绑和电子市场为主，占比分别为19.37%，18.64%和17.50%。

移动互联网的发展使手机成为日常生活不可或缺的一部分，随着手机功能的增强，手机价值越来越大，与之相伴的是越来越严峻的安全挑战。

##### 环境安全SDK服务解决方案产品价值

腾讯移动应用安全-环境安全SDK服务解决方案服务是一套专为客户提供移动安全服务和技术的解决方案，涵盖有病毒扫描、网址检测、短信拦截、号码识别、垃圾清理、流量校准、WiFi检测等功能。腾讯移动应用安全-环境安全SDK服务解决方案领先的综合安全能力，能全方位消除移动风险，深度保障用户的安全体验，构建了坚不可摧的防护屏障。

##### 环境安全SDK服务解决方案产品原理

腾讯移动应用安全-环境安全SDK服务解决方案服务积累了丰富的行业经验，依托海量的数据计算平台和领先的AI引擎智能研判，提供稳定可靠的安全能力和服务，满足全方位的安全场景需求。通过强大的样本收集能力，形成国内最大的病毒样本库，领先的启发式病毒查杀引擎，敏锐的恶意样本感知能力，使腾讯移动应用安全-环境安全SDK服务解决方案能够极速捕获新病毒，响应迅速。

##### 环境安全SDK服务解决方案产品组成

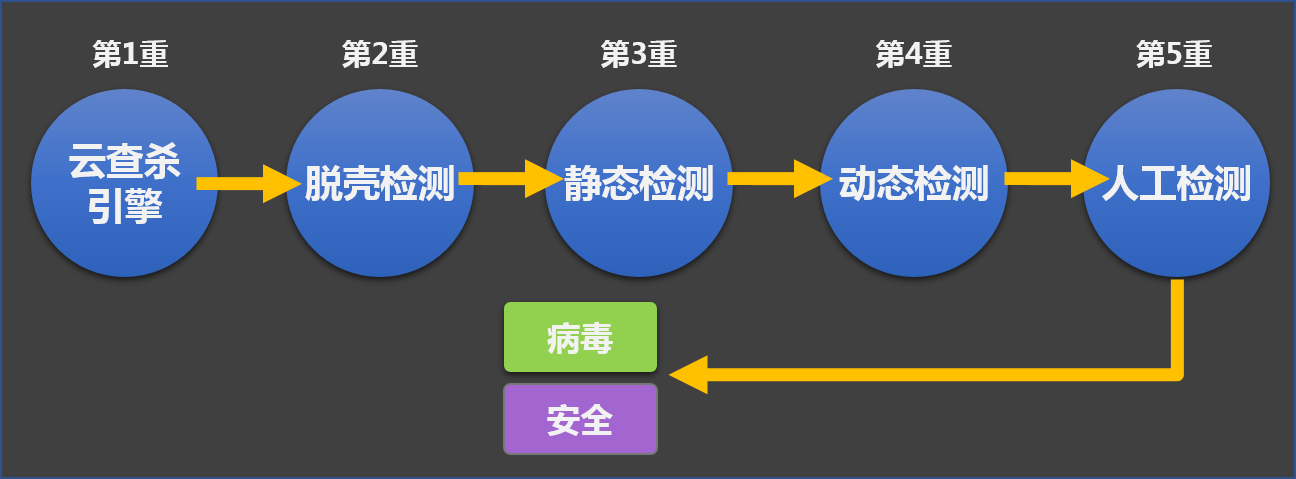
腾讯移动应用安全-环境安全SDK服务解决方案服务包含以上内容，安卓支持4.2以上所有机型，iOS支持iPhone5以上所有机型。  
 腾讯移动应用安全-环境安全SDK包体小，易集成，安卓安全环境清场各模块SDK接入后包体新增小于1M;IOS环境安全SDK接入后包体新增大小268k。

#### 产品详述

##### 病毒扫描

###### 产品原理

腾讯移动应用安全-环境安全SDK服务解决方案病毒扫描服务拥有强大的样本收集能力，与国内多家头部手机厂商展开深度合作，拥有国内最大的病毒样本库，并多次在国际评测中获得第一。同时腾讯拥有领先的启发式病毒查杀引擎—TRP-AI智能引擎，拥有敏锐的恶意样本感知能力，能够极速捕获新病毒，结合基于大数据构建的动静态复合检测系统、特征系统形成云+端的立体查杀能力，全方位保护用户手机安全，可免疫99%以上的加壳或变种病毒。AI云查杀+智能静态本地查杀双引擎运作机制，攻克了传统的“病毒库不升级就杀不了新病毒”的技术难题，不仅本地病毒查杀能力更强，效率更高，自身体积也极大的缩减，查杀速度比传统引擎快至少一倍。



###### 产品组成

腾讯病毒扫描服务以本地+云查的方式，支持扫描已经安装的应用宝和已下载未安装的apk文件。其中，病毒特征样本APK库达3.5亿，云查病毒特征库分钟级更新，本地病毒特征库支持实时更新。每日服务用户数为亿级。

###### 产品功能

* 快速查杀

针对移动智能设备上已经安装的移动应用程序App进行安全检测，调用TRP-AI智能引擎在本地环境下进行病毒查杀。

* 全盘查杀

在【快速查杀】功能基础上，叠加移动智能设备SD卡/手机存储上存储的安装包进行安全性检测，全面检测移动智能设备安全性，发现木马病毒应用进行报警提示用户进行卸载或者删除。

* 安装监控

在用户使用设备过程中，如果有软件安装的行为，会自动触发扫描，对应用的安全性进行判定。安装监控为常驻服务以静默的方式后台检测用户所安装的软件，防范各种安全威胁和风险。

* 云查杀

当本地TRP-AI智能引擎检测后，且智能设备联网情况下，将把检测到的所有App包名/apk安装包信息上报云端，请求云查引擎基于云端病毒库进行云端二次补充查杀，避免漏掉本地未知样本的病毒问题。联网云查杀消耗的流量非常之小，100款App应用程序经过测试只消耗4-5KB流量。

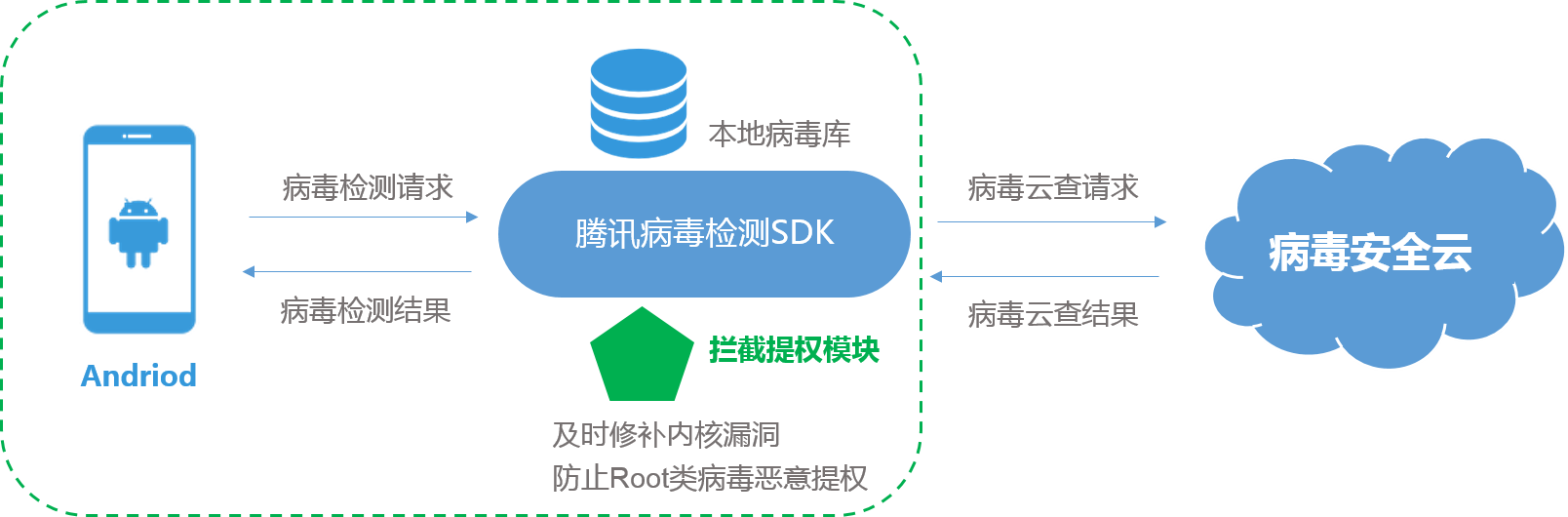
* 病毒库更新

根据产品病毒库的更新节奏，自动请求云端病毒库，当发现云端有病毒库更新时，则自动推送到本地，完成本地病毒库的更新。

* 安全日志

程序能够以log日志的形式，详细的显示扫描日志。安全日志将全面记录每一次扫描的触发时间，扫描结果等等信息。

###### 产品架构



腾讯病毒查杀SDK能力，保留原有的本地检测和云查的能力，新增拦截恶意提权的模块，全面提升对Root类病毒的防御能力。基于Android高权限，拦截病毒的恶意提权，让用户免受提权型病毒或者恶意程序的攻击，保护用户数据的安全。在不升级系统的前提下，通过SDK拦截提权模块，阻断第三方应用拿Root权限的通道。

合作开发者得到腾讯提供的病毒查杀SDK，通过调用杀毒引擎接口对自身应用进行集成使自身应用具备病毒查杀能力，杀毒引擎在使用过程中，连接腾讯云安全中心进行云查杀和病毒库升级。

###### 主要技术参数

|  |  |
| --- | --- |
| 主要参数（以2018年数据为准） | 详情 |
| 年度截获Android新增高危病毒包 | 800.62万 |
| Android手机病毒感染用户数 | 1.13亿 |
| 查杀病毒 | 8.12亿次 |
| 支持扫描的病毒类型 | 恶意软件、隐患软件、支付风险、盗号风险、恶意文件、非官方证书、广告插件 |
| 功能触发方式 | 手动触发及自动触发均支持  手动触发：用户主动扫描  自动触发：APK安装前；应用安装后；病毒库更新后 |
| 病毒库更新机制 | 可实时更新，本地查杀支持获取最新病毒库，云端始终保持最新病毒库，建议使用病毒云查，以直接获取最新病毒检查结果 |

##### 网址检测

###### 产品原理

恶意网址是指恶意种植木马、病毒等恶意程序在网站内，通过“伪装的网站服务内容”诱导用户访问该网站，一旦进入这些网站，便会触发网站内种植下的木马、病毒等程序，导致访问者计算机被感染，面临丢失帐号或者隐私信息等危险。

基于海量数据处理能力，海量社交用户举报数据，结合全面的互联网产品接入，腾讯网址检测智能过滤技术及千万级恶意网址特征，实现精准过滤，误差低；海量多纬度数据结合人工智能算法，能快速感知，打击新的变种；十年安全沉淀，拥有业界最大的恶意网址黑库。



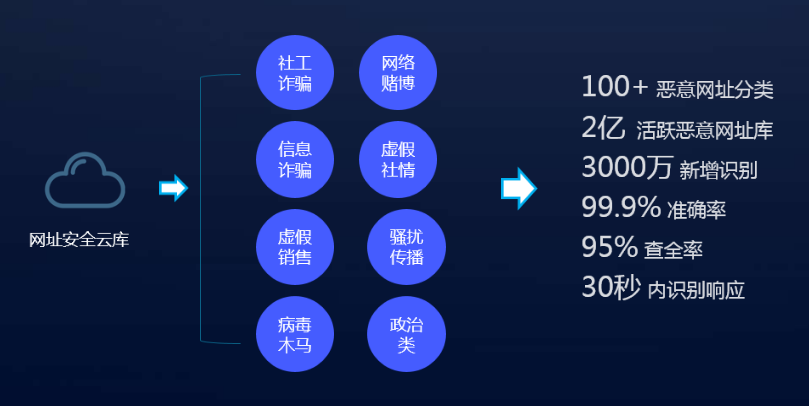
腾讯网址检测服务支持安卓及iOS系统。

###### 产品组成

腾讯网址检测服务中恶意网址样本库共计2亿活跃恶意网址库，包含上百种恶意网址分类，包含但不限于信息诈骗、恶意文件、博彩网站、风险网站、色情网站、虚假销售等类型

###### 产品功能

本方案提供了恶意网址检测的功能，应用向SDK输入需要备检测的URL，SDK返回URL的类型。功能详述见下：



###### 产品架构



###### 主要技术参数

|  |  |
| --- | --- |
| 输入数据 | 输出数据 |
| 网址URL | 网址类型：钓鱼欺诈，信息欺诈，虚假广告，恶意程序，博彩网站，色情内容，垃圾信息，非法内容  拦截建议：正常，恶意，可疑，未知 |

##### WiFi安全检测

###### 产品原理

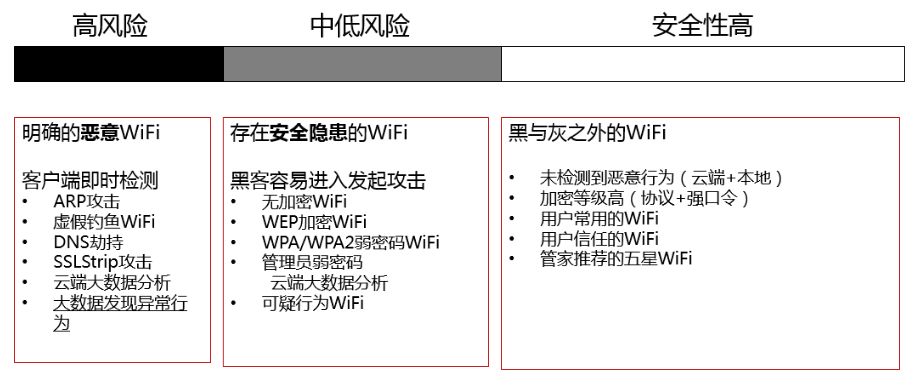
腾讯安全WiFi联盟的数据显示，全球约10%的人口正在使用WiFi，作为全球最大的市场之一，中国目前公共场所WiFi热点覆盖至少超过千万个，这项服务几乎已经成为了公共场所服务范围内的标准配置。

据腾讯手机管家《爱蹭WIFI的人必看》报告显示，在Android联网用户中，高达49.75%的人用WIFI联网，86.03%的用户最爱用WIFI聊天，67.23%的人认为WIFI联网的最大问题是速度慢，62.05%的人吐槽WIFI连接麻烦需要密码，而有49.14%的人担心WIFI的安全问题。

腾讯手机管家安全专家分析，虚假WiFi钓鱼是当前免费WiFi的主要安全风险, 315晚会现场黑客盗取用户隐私的主要方法就是利用了虚假WiFi钓鱼的手法。所谓虚假WiFi钓鱼，是指犯罪分子通过架设一个与某公共WiFi热点同名的WiFi网络，吸引用户通过移动设备接入该网络，然后就可以通过分析软件窃取这些接入虚假WiFi热点用户的资料，包括WiFi登录密码，从而成功破译。

而且通过这一手段，还能窃取到用户的银行账户、网络支付账户密码，从而实施资金的盗刷。除了虚假WiFi钓鱼外，“DNS劫持”、“ARP欺骗攻击”等黑客攻击手段也会被用来在免费WiFi网络下对在网用户进行恶意攻击，导致网络瘫痪、窃取网购支付账号密码等。

基于腾讯手机管家、腾讯WiFi管家等产品矩阵的安全检测触点，每日感知数以亿次的WiFi连接，实时防御ARP攻击，DNS欺骗，SSLStrip攻击等风险。同时，根据海量WiFi风险检测数据，借助大数据学习和AI算法，实时判断虚假钓鱼WiFi，构建业内最大的WiFi威胁情报库，保护用户WiFi网络安全。



###### 产品组成

腾讯问题wifi安全检测服务支持安卓及iOS系统。

###### 产品功能

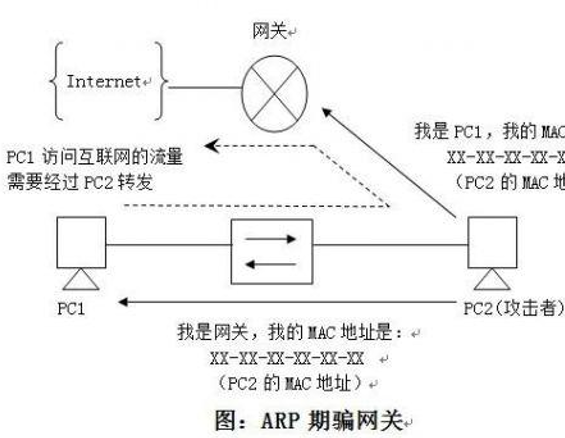
* ARP攻击检测

ARP攻击就是通过伪造IP地址和MAC地址实现ARP欺骗，能够在网络中产生大量的ARP通信量使网络阻塞，攻击者只要持续不断的发出伪造的ARP响应包就能更改目标主机ARP缓存中的IP-MAC条目，造成网络中断或中间人攻击。

ARP攻击主要是存在于局域网网络中，局域网中若有一台计算机感染ARP木马，则感染该ARP木马的系统将会试图通过“ARP欺骗”手段截获所在网络内其它计算机的通信信息，并因此造成网内其它计算机的通信故障。

攻击者向电脑A发送一个伪造的ARP响应，告诉电脑A：电脑B的IP地址192.168.0.2对应的MAC地址是00-aa-00-62-c6-03，电脑A信以为真，将这个对应关系写入自己的ARP缓存表中，以后发送数据时，将本应该发往电脑B的数据发送给了攻击者。同样的，攻击者向电脑B也发送一个伪造的ARP响应，告诉电脑B：电脑A的IP地址192.168.0.1对应的MAC地址是00-aa-00-62-c6-03，电脑B也会将数据发送给攻击者。

至此攻击者就控制了电脑A和电脑B之间的流量，他可以选择被动地监测流量，获取密码和其他涉密信息，也可以伪造数据，改变电脑A和电脑B之间的通信内容。



目前ARP攻击在WiFi风险占比中最大。本方案提供ARP攻击检测能力。

* DNS劫持检测

DNS劫持又称域名劫持，是指在劫持的网络范围内拦截域名解析的请求，分析请求的域名，把审查范围以外的请求放行，否则返回假的IP地址或者什么都不做使请求失去响应，其效果就是对特定的网络不能访问或访问的是假网址。

DNS（域名系统）的作用是把网络地址（域名，以一个字符串的形式）对应到真实的计算机能够识别的网络地址（IP地址），以便计算机能够进一步通信，传递网址和内容等。由于域名劫持往往只能在特定的被劫持的网络范围内进行，所以在此范围外的域名服务器(DNS)能够返回正常的IP地址，高级用户可以在网络设置把DNS指向这些正常的域名服务器以实现对网址的正常访问。所以域名劫持通常相伴的措施——封锁正常DNS的IP。

如果知道该域名的真实IP地址，则可以直接用此IP代替域名后进行访问。比如访问百度域名，可以把访问改为202.108.22.5，从而绕开域名劫持 。

DNS劫持(DNS钓鱼攻击)十分凶猛且不容易被用户感知，本方案提供DNS劫持检测功能，通过网关，掩码，DNS列表及周围AP列表这些条件进行判断。

* SSLstrip攻击检测

在浏览器、电子邮件、即时通讯、VoIP、网络传真等应用程序中，广泛支持SSL/TLS协议。主要的网站，如Google、Facebook、Twitter等也以SSL/TLS协议来创建安全连接，发送数据。SSL/TLS已成为互联网上保密通讯的工业标准。这很大一部分程度上杜绝了用户的隐私信息的泄露。然而，HTTPS本身也并非无懈可击，随着对HTTPS的研究的不断深入，针对其的攻击方法也日渐增多，这些攻击不仅仅让用户的隐私受到威胁，同时也阻碍了互联网安全的发展。其中最典型的一种攻击方式就是中间人攻击（MITM）。 　　所谓中间人攻击，是指攻击者与通讯的两端分别创建独立的联系，并交换其所收到的数据，使通讯的两端认为他们正在通过一个私密的连接与对方直接对话，但事实上整个会话都被攻击者完全控制。在中间人攻击中，攻击者可以拦截通讯双方的通话并插入新的内容。目前，主流的攻击方式有SSL劫持攻击、SSLStrip攻击以及漏洞攻击等。由于HTTPS所使用的证书颁发者必须是经过认证的对象，因此如果使用SSL劫持攻击则必须让用户信任攻击者自己签发的证书，这使得该攻击的成功率大大降低。鉴于目前针对HTTPS的漏洞攻击尚不成熟，本文就着重来探讨SSLStrip的攻击实现方式，以及针对该类攻击的防御措施。

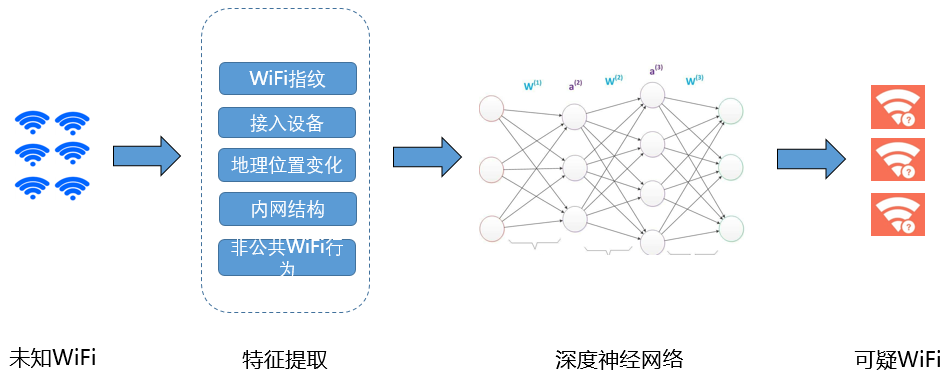
SSLStrip，是一种不针对任何程序错误，而是基于HTTPS体系实现的攻击方式。其原理为基于多数用户不会主动请求SSL协议来保护自己和服务器之间的通信，从而不会主动在浏览器中输入https访问域名。但是由于http跳转https往往是通过服务器端302重定向实现，因此本攻击方式就抓住了这个特点，在用户端接受到跳转信息之前通过明文HTTP协议去除掉HTTPS跳转的过渡，从而使得原先本应被加密的信息都明文呈现在了攻击者的面前，使得攻击者能够自由获取需要的信息。

本方案通过攻击逻辑进行防御提醒，提供SSLStrip攻击检测功能。

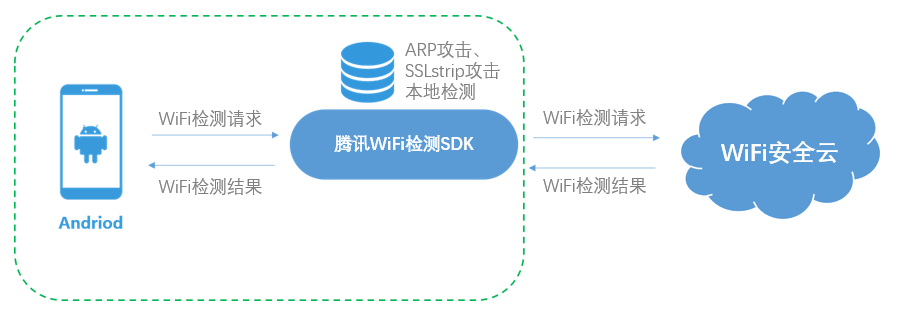
* 虚假钓鱼WiFi检测

用智能手机上网，手机可以自动搜索到附近的Wi-Fi网络账号，有时甚至有多个账号供选择登录上网，多数Wi-Fi网络账号需要登录密码，而在机场咖啡店等公共场所，Wi-Fi网络一般是不需要密码的，部分商家把免费Wi-Fi网络作为招揽人气的手段，但也给“钓鱼WiFi”窃取账号、密码的机会。

本方案根据WiFi特征，通过深度神经学习网络及随机森林双结合的方法，提供快速识别虚假钓鱼WiFi能力。



###### 产品架构



###### 主要技术参数

|  |  |  |
| --- | --- | --- |
| 检测接口 | 输入数据 | 输出数据 |
| ARP攻击检测 | ip和mac对应表 | 是否APR攻击 |
| DNS劫持检测 | 当前连接bssid ssid | 是否dns劫持 |
| SSL strip攻击检测 | 无 | 是否SSL strip攻击 |
| 虚假钓鱼WiFi检测 | 当前连接bssid ssid | 是否虚假钓鱼wifi |

##### 防界面劫持

###### 产品原理

界面劫持一般指的是Android平台上的病毒或木马在后台实时监控某个应用程序中特定界面的打开事件，例如QQ、微信、支付宝、手机银行等软件的登录界面或其他目标界面。一旦发现目标界面出现，恶意程序就会马上弹出一个一模一样的界面覆盖在原始界面之上。

此时，即发生了界面劫持攻击事件，用户会在无察觉的情况下将自己的账号、密码等敏感信息输入到恶意程序的界面中，恶意程序再把这些数据上报到其自身服务器，导致用户敏感信息泄露，造成财产损失。

界面劫持攻击技术含量低、但用户难察觉、无明显病毒特征导致安全软件查杀困难等特点而被广泛使用。

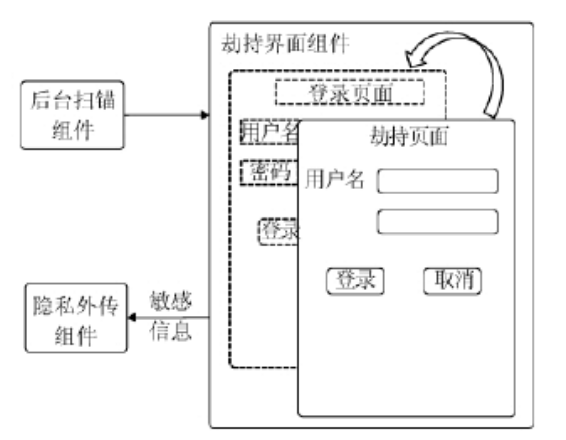
腾讯安全防界面劫持SDK针对此攻击事件中关键行为进行检测来达到对界面劫持事件的感知，即检测当前正在运行的应用程序界面是否是属于本应用程序的界面，如不是则会及时弹出提醒或后台进一步处理。

###### 产品组成

产品形态为SDK，可以定制化配置，并包含白名单系统，避免某些情况下来自于被允许的系统应用悬浮窗的误报。

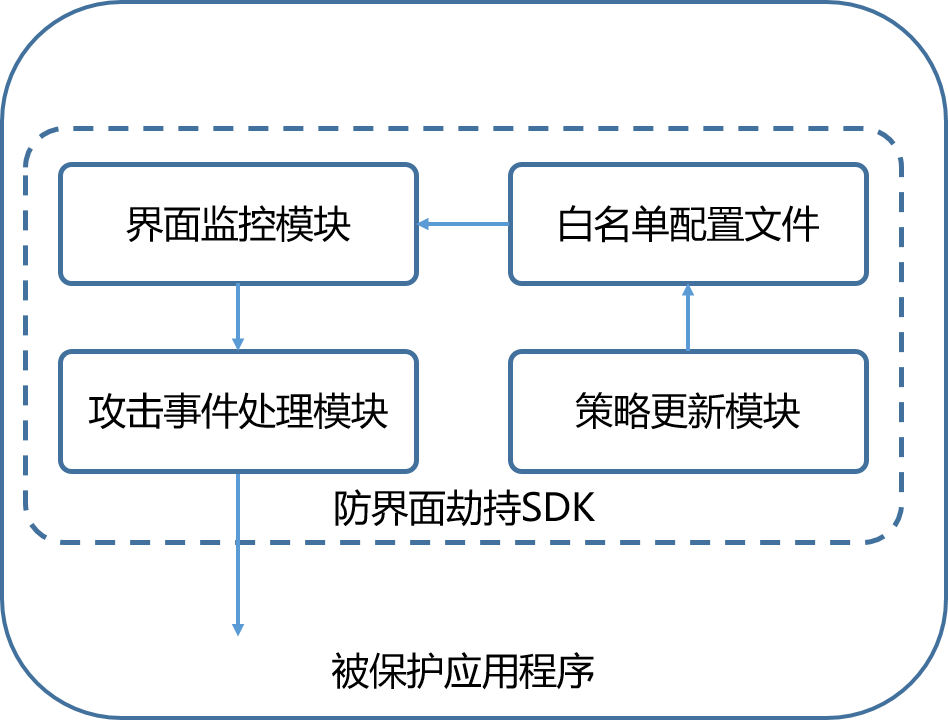
###### 产品功能

界面劫持攻击手法在PC时代就已经比较成熟。虽然在移动互联网时代系统厂商对于权限的限制，此类攻击并无PC时代猖獗，但是随着智能手机能力越来越强，用户对于资产的管理都可以用智能手机操作，一旦发生界面劫持盗取用户信息的攻击行为，给用户造成的损失往往会比较严重，所以仍然不可掉以轻心。



本产品功能主要是在应用程序有界面切换行为时进行疑似攻击事件的检测，及时提醒用户注意防范。

###### 产品架构



###### 主要技术参数

|  |  |
| --- | --- |
| 关键监测点 | 输出数据 |
| 应用程序界面切换行为 | 若界面被覆盖则弹出相应提示 |

##### 防录屏

###### 产品原理

很多应用程序的界面会显示用户的隐私数据，例如身份住址信息、财产信息、密码信息等等，如果被第三方软件录屏，则会导致用户隐私数据泄露。另外，自由版权的内容产品，如有被第三方恶意录屏，也会受到严重版权侵犯。

腾讯安全防录屏SDK从系统底层实现一套自定义属性的安全Activity，被保护应用的Activity只需要继承SDK内的安全Activity，即可从根源防止被保护应用被截图或录屏。

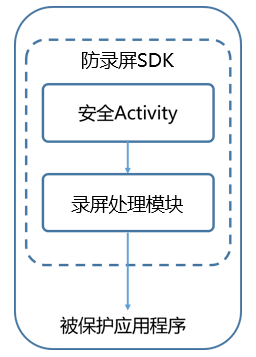
###### 产品组成

产品形态为SDK，可以对特定的Activity进行细粒度的保护，也可以定制被录屏时的提示语。

###### 产品功能

防止应用程序内敏感界面被录屏。

###### 产品架构



###### 主要技术参数

|  |  |
| --- | --- |
| 关键监测点 | 结果操作 |
| 截屏事件监测 | Android平台对截屏事件进行拦截  iOS平台对截屏事件进行提醒 |
| 录屏事件监测 | Android平台对录屏事件进行拦截  iOS平台对录屏事件进行提醒 |

##### 防截屏

###### 产品原理

很多应用程序的界面会显示用户的隐私数据，例如身份住址信息、财产信息等等，如果被第三方软件截图、或用户自己截图时被第三方应用抓取，则会导致用户隐私数据泄露。

腾讯安全防截屏SDK从系统底层实现一套自定义属性的安全Activity，被保护应用的Activity只需要继承SDK内的安全Activity，即可从根源防止被保护应用被截图或录屏。

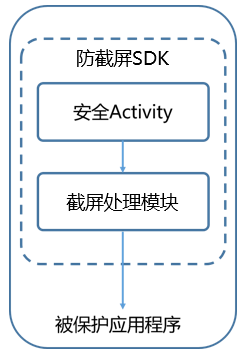
###### 产品组成

产品形态为SDK，可以对特定的Activity进行细粒度的保护，也可以定制被截图时的提示语。

###### 产品功能

防止应用程序内敏感界面被截屏。

###### 产品架构



###### 主要技术参数

|  |  |
| --- | --- |
| 关键监测点 | 结果操作 |
| 录屏事件监测 | Android平台对录屏事件进行拦截  iOS平台对录屏事件进行提醒 |

##### root检测

###### 产品原理

Android安全架构是基于Linux多用户机制的访问控制。应用程序在默认的情况下不可以执行其他应用程序，包括读或写用户的私有数据(如联系人数据或email数据)，读或写另一个应用程序的文件。

一个应用程序的进程就是一个安全的沙盒（在受限的安全环境中运行应用程序，在沙盒中的所有改动对操作系统不会造成任何危害）。它不能干扰其它应用程序，除非显式地声明了“permissions”，以便它能够获取基本沙盒所不具备的额外的能力。

每一个Android应用程序都会在安装时就分配一个独有的Linux用户ID，这就为它建立了一个沙盒，使其不能与其他应用程序进行接触。这个用户ID会在安装时分配给它，并在该设备上一直保持同一个数值。

所有的Android应用程序必须用证书进行签名认证，而这个证书的私钥是由开发者保有的。该证书可以用以识别应用程序的作者。签名影响安全性的最重要的方式是通过决定谁可以进入基于签名的permisssions，以及谁可以share 用户IDs。通过这样的机制，在不考虑root用户的情况下，每个应用都是相互隔离的，实现了一定的安全。

在系统中，每个文件、目录和进程，都归属于某一个用户，没有用户许可其它普通用户是无法操作的，但对root除外。

root用户的特权性还表现在：root可以超越任何用户和用户组来对文件或目录进行读取、修改或删除（在系统正常的许可范围内）；对可执行程序的执行、终止；对硬件设备的添加、创建和移除等；也可以对文件和目录进行属主和权限进行修改，以适合系统管理的需要（因为root是系统中权限最高的特权用户）；root是超越任何用户和用户组的，基于用户ID的权限机制的沙盒是隔离不了它的。

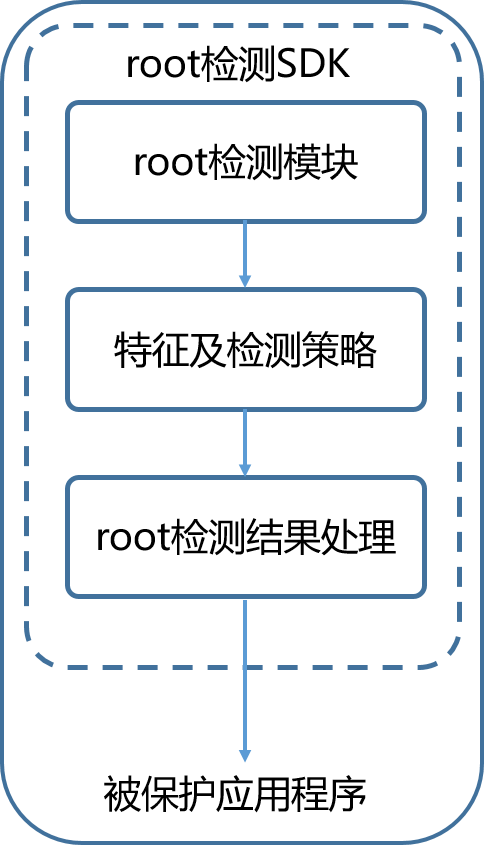
###### 产品组成

产品形态为SDK，可以对运行系统是否被root进行检测。

###### 产品功能

用于检测运行系统是否被root。

###### 产品架构



###### 主要技术参数

|  |  |
| --- | --- |
| 关键技术 | 技术原理 |
| 多维度检测技术 | 防止误判，本SDK采集了系统key相关信息、su相关命令、权限管理相关apk、root后必备工具、敏感目录权限等多维度数据，进行综合判断 |
| 防绕过技术 | 本SDK采用最底层实现，防止攻击者通过hook所有IO操作来实现绕过root检测 |

##### 越狱检测

###### 产品原理

iOS 越狱(iOS Jailbreaking)，是用于获取苹果公司便携装置操作系统iOS最高权限的一种技术手段，用户使用这种技术及软件可以获取到 iOS 的最高权限，甚至可能可以进一步解开运营商对手机网络的限制。手机越狱后会面临更多的安全风险，不建议对iOS设备进行越狱。在开发中可以针对越狱设备进行安全防范。

检测越狱手机一直是iOS应用安全防护的第一道门槛。

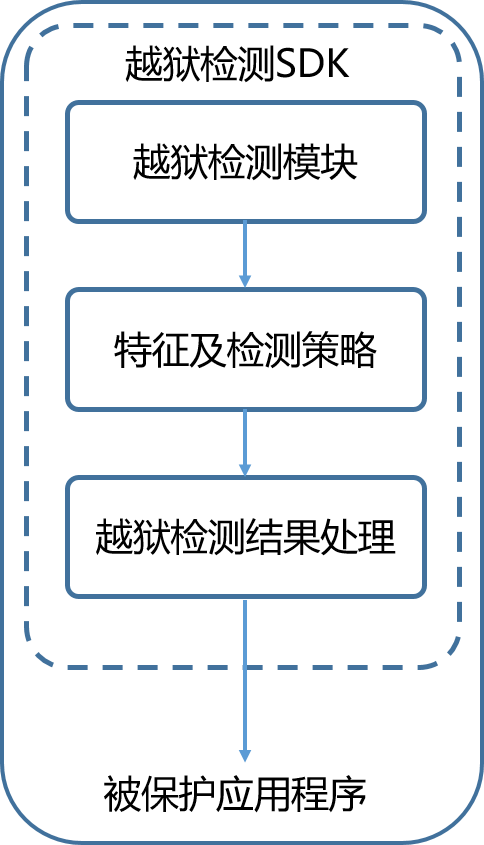
###### 产品组成

产品形态为SDK，可以对运行系统是否越狱进行检测。

###### 产品功能

用于检测运行系统是否越狱。

###### 产品架构



###### 主要技术参数

|  |  |
| --- | --- |
| 关键技术 | 技术原理 |
| 多维度检测技术 | 防止误判，本SDK采集了沙盒完整性信息、常见越狱文件、fstab文件信息、系统库符号链接文件、越狱后必备工具、敏感目录权限等多维度数据，进行综合判断 |
| 防绕过技术 | 本SDK采用最底层实现，防止攻击者通过hook所有IO操作来实现绕过越狱检测 |

##### 模拟器检测

###### 产品原理

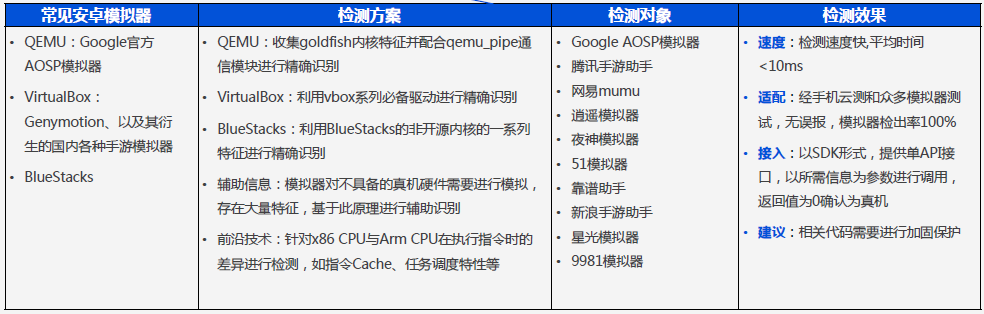
在Android开发过程中，防作弊一直是重点问题，而模拟器的检测往往是防作弊中的重要一环。模拟器检测可用于防止利用模拟器进行刷单，刷流量等恶意操作。

###### 产品组成

模拟器检测方案主要包括模拟器检测、模拟环境检测、App运行状态检测三个模块。

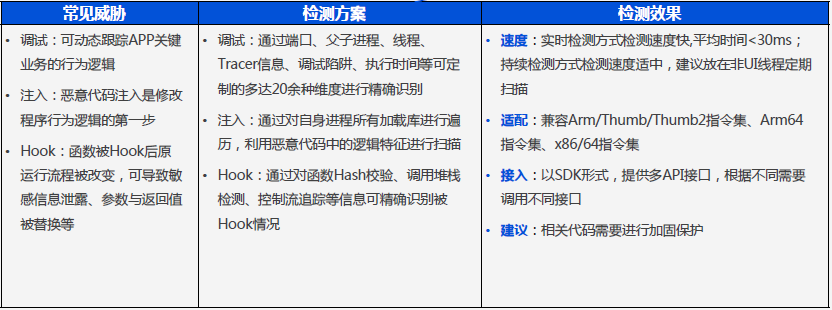
###### 产品功能

* 模拟器检测



虚拟环境检测  


* App运行状态检测



###### 产品架构



###### 主要技术参数

|  |  |
| --- | --- |
| 关键技术 | 技术原理 |
| 多维度检测技术 | 防止误判，本SDK采集了多维度数据，进行综合判断，支持市面上常见的模拟环境以及衍生版，可动态调节检测阈值 |
| 防绕过技术 | 本SDK采用最底层实现，防止攻击者通过hook所有IO操作来实现绕过模拟器检测 |
| 热更新技术 | 对抗模拟环境是不断攻防的过程，本SDK支持云端热更新技术，通过热更新可以支持最新模拟器及最新版本的检测 |

#### 产品优势



##### 深耕移动安全，扎实能力积累

腾讯安全是由腾讯公司打造，以腾讯安全联合实验室作为实力技术支撑的互联网安全领先品牌。腾讯安全旗下产品已成为中国最强的互联网安全矩阵，包含面对用户的全方位网络安全应用、对企业与政府网络安全的系列解决方案，已发展成为保卫腾讯互联网业务发展及用户、企业、政府三方网络安全的国防军。



技术实力雄厚，国际一流的腾讯安全联合实验室矩阵能力保障。

##### 国际领先AI威胁检测引擎

1）具备国内外最大手机病毒库：

通过5000种病毒类型的跟踪，1400个病毒家族的监控，278个病毒传播渠道分析，数百个黑产团伙的追踪，20年来积累大量威胁情报。

20多款病毒检测引擎，19种应用加固脱壳能力，建立大数据机器学习检测引擎，通过静态+动态沙箱行为检测，快速精准病毒检测并查杀。

基于10亿的设备指纹信誉数据，278个手机病毒监控渠道，具备海量手机病毒库，3.5亿apk文件，自建存储机房。

软件样本来源广泛，具有业界领先的覆盖率。通过特有优势的拦截来源，拥有超过4500万的安卓软件池，覆盖了超过99%的热门、常用手机软件。

2）具备业界最大的恶意网址黑库：

95%的互联网活跃网址覆盖，100多家合作伙伴接入，打击准确率高达99.9%

3）具备业内最新的恶意wifi黑库：

WiFi安全检测每日调用亿级，主要依赖云端大数据深度分析黑产行为及特征，精准打击，并佐以深度学习训练保障首发检测

##### 屡获各类国际测评机构最高认证

AV-Test 腾讯杀毒引擎100%检出，0误报，第五次排名世界第一

赛可达实验室2018 年度全球手机安全软件横评，腾讯杀毒引擎100%查杀率获得满分，持续领跑安全行业。

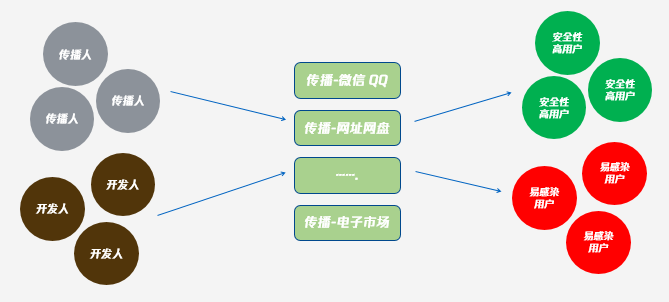


安全机构权威资质认证，质量保障。

##### 腾讯独有的态势感知与威胁线索追踪能力

独有的微信、QQ社交网络数据，监控黑产团伙的开发与传播；完整掌控病毒开发，传播，危害用户等群体，全面的鉴别APK的前世今生。

持续重点投入研究病毒的感染态势、发展趋势、入侵技术追踪、免杀技术分析，在对抗中保持优势。



##### 多年B端合作经验，海量C端用户服务经验

病毒/网址/WiFi等检测产品与众多头部手机厂商合作，渗透率达90%。

依托月活过亿的腾讯手机管家、腾讯WiFi管家两款App，积累丰富的C端用户服务用户以及大数据模型，其中，每日过亿的病毒查杀次数，WiFi检测功能日均服务2亿+用户。



安全服务经验丰富，政-企-民一致认可服务保障。

### 应用加固

#### 腾讯安全加固技术点简介

腾讯安全的加固技术业界领先，获得CNCERT认证，截至2018年终提供十亿用户使用。

腾讯安全向客户提供的移动应用代码保护技术如下：

1. 反编译与反汇编保护
2. 客户端防篡改保护
3. 客户端防注入保护
4. 客户端反调试保护
5. 客户端本地数据与资源文件加密保护
6. SO库绑定保护
7. 运行时环境检测技术

#### 安全产品及服务

##### 安全产品及服务列表

|  |  |  |  |
| --- | --- | --- | --- |
| **安全加固类别** | **功能模块** | **基础版** | **专业版** |
| 静态代码保护 | DEX反编译保护 | APK整体加壳保护 | APK整体加壳保护 |
| DEX文件整体加固保护 | DEX文件整体加固保护 |
|  | Java To Native函数级保护 |
|  | Java虚拟机(VMP)函数级保护 |
| SO反编译保护 | SO库加壳保护 | SO库加壳保护 |
| SO库字符串加密 | SO库字符串加密 |
|  | SO库防盗用校验 |
|  | 基于编译器的高强度混淆保护 |
|  | C/C++虚拟机(VMP)函数级保护 |
| iOS防破解保护 |  | 二进制代码混淆保护 |
|  | 库文件代码混淆保护 |
| 完整性保护 | APK防篡改保护 | APK防二次打包保护 | APK防二次打包保护 |
| APK签名文件校验保护 | APK签名文件校验保护 |
| 源代码防篡改保护 | DEX文件防篡改 | DEX文件防篡改 |
| SO库防篡改 | SO库防篡改 |
| 资源防篡改保护 |  | Assets资源防篡改 |
|  | Res资源防篡改 |
|  | Raw资源防篡改 |
|  | 配置文件防篡改 |
| 动态运行防护 | 反调试保护 | 防Java代码调试 | 防Java代码调试 |
| Native代码调试检测 | Native代码调试检测 |
|  | 代码注入检测 |
|  | 防内存dump保护 |
| 安全环境检测 | 模拟器检测 | 模拟器检测 |
| 设备Root状态检测 | 设备Root状态检测 |
| Xposed环境检测 | Xposed环境检测 |
|  | 应用多开检测 |
|  | 系统函数Hook检测 |
|  | 唯一设备识别 |
| 数据与资源保护 | 资源防窃取保护 |  | Assets资源加密 |
|  | Res资源加密 |
|  | Raw资源加密 |
|  | SSL证书加密 |
| 本地数据保护 | 本地数据库文件加密 | 本地数据库文件加密 |
|  | 安全键盘 |
|  | 防日志泄漏 |
|  | 应用防截屏/录屏 |

##### 产品交付及使用方式

###### 独立服务器私有化部署

**交付方式：**

腾讯安全提供独立服务器设备，其已安装Linux/Mac双系统，并在系统中部署一个Web平台。

**使用方式：**

对安卓应用，通过内网访问Web平台，根据平台操作指引，上传apk，点击加固按钮，稍等片刻后即可下载加固好的安卓apk。

对iOS应用，使用Mac系统打开web平台，根据平台操作指引，将iOS工程代码打包成zip文件后上传，点击加固按钮，即会调用腾讯安全定制版Xcode对该工程进行编译，编译完毕后即可得到加固好的iOS应用包。

###### 客户准备服务器私有化部署

**交付方式**：

腾讯安全派技术人员到企业，在企业事先准备好的符合规格的服务器内安装Linux/Mac双系统。安装完毕后双系统均将自带一个Web平台。

**使用方式**：

与2.2.1的使用方式相同。

###### 远程在线服务

**交付方式**：

腾讯安全提供分别针对安卓与iOS的加固网络接口。

**使用方式：**

对安卓应用，企业调用安卓加固接口提交APK，接口处理完毕后返回已加固APK的下载地址，企业自行下载即可。

对iOS应用，企业需将工程预编译为BitCode文件，打成压缩包后调用iOS加固接口提交，接口处理完毕后返回已加固BitCode文件压缩包的下载地址，企业自行下载并解压后进行二次编译即可。

#### 反编译与反汇编保护

##### 背景

Android平台采用Java做为主要开发语言，C/C++语言为辅助开发语言。其中Dex文件是由Java源代码经过一系列处理而生成。Dex文件很容易被反编译，并且反编译后的代码可读性非常高，接近源代码。

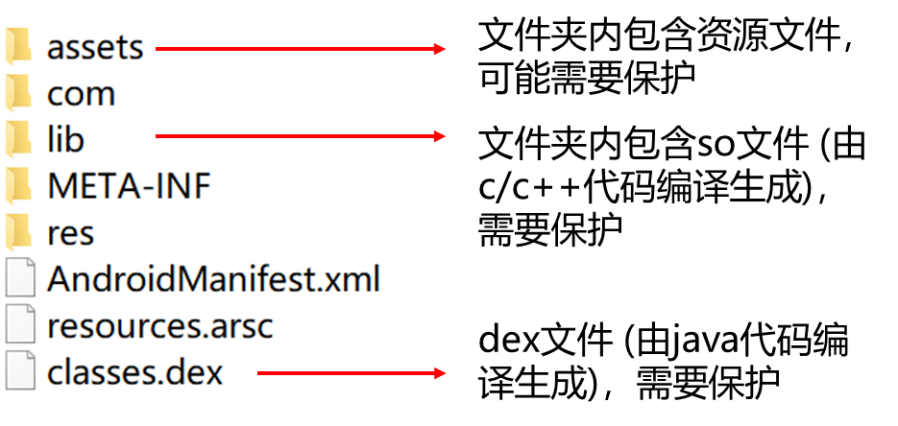
Android平台的SO库文件采用C/C++语言编写，iOS平台的二进制程序与DYLIB库文件采用Swift/Objective-C/C/C++编写，以上方式编译的程序在编译后都会生成对应平台的汇编指令，如现在智能手机平台ARM指令集，虽然汇编指令较难看懂，但是在目前例如IDA等辅助逆向工具的存在，也大大降低了攻击者的分析门槛。

攻击者可通过逆向分析对关键业务逻辑进行攻击，绝大部分发生在终端的攻击都依赖于逆向分析的结果。腾讯安全为客户提供的代码保护方案，可以有效对抗对终端各类代码的反编译与反汇编攻击，在第一步就将绝大部分攻击者拒之门外，筑起终端安全的第一道防线。

##### 技术原理

###### Android代码保护原理

Android平台的安装包文件内部如下图所示，其中Dex文件，SO库文件及资源文件，都需要保护。



腾讯安全采用第四代代码保护技术对Android平台Dex文件进行保护，每一代技术特点如下：

|  |  |  |
| --- | --- | --- |
|  | **技术方案** | **安全性** |
| **第一代** | 对整个Dex文件进行加密，执行时完全解密执行 | 可对抗静态分析，但安全性极低 可通过Hook或Dump内存方式进行破解 |
| **第二代** | 对整个Dex文件进行加密，在函数执行前解密单函数 | 可对抗静态分析与Dump攻击，但安全性较低 可通过DexHunter进行破解 |
| **第三代** | 加壳后将部分Java代码转移到Native层 | 可对抗静态分析及Dump攻击及DexHunter 安全性一般，可通过动态调试进行破解 |
| **第四代** | 嵌入虚拟机，在运行时由虚拟机对指令进行解析执行 | 可对抗以上所有攻击方式，安全性很高 |

###### Android SO库保护原理

SO库的加密保护利用的是加壳技术。主要原理是在原SO库文件的外层包裹一层保护壳，保护壳具备以下安全能力：

1. 汇编指令集加密

加密原始汇编指令，通过IDA等逆向分析工具无法看到原始汇编指令，从而阻止被逆向分析。

1. SO库文件头部信息擦除

SO库文件头部存放着大量的辅助分析信息，容易被攻击者利用。保护壳对这些数据进行擦除，并且不影响程序正常运行，让破解者无法根据这些信息进行逆向分析，极大提到逆向分析门槛。处理的数据如：重定向信息，符号表等。

1. SO库加载流程保护

在Android平台上，未经保护的SO库文件是由系统的linker加载，攻击者容易通过修改加载器来进行对SO库的破解。保护壳内嵌了自定义的加载器，防止攻击者修改系统连接器进行的攻击。

1. SO库内存防Dump保护

保护壳具备内存中函数加解密功能，函数未执行时，在内存中代码是加密的，在执行之前解密，并在执行之后重新加密，内存中不出现完整SO库文件有效抵抗攻击者的Dump攻击。

###### iOS代码保护原理

由于iOS系统的封闭性，对于iOS代码进行加密保护或加壳保护的方案不可行，为此我们专门设计了一套基于编译器的代码混淆保护方案，具有强度高，性能影响小，稳定性高的特点。

安全编译器支持Swift/Objective-C/C/C++语言，在编译过程中，通过对中间语言文件的处理，将原函数控

制流进行扁平化，并加入虚拟代码块，在对部分指令进行等价指令替换后，将字符串与导入表等信息进行隐藏，有效抵抗静态分析与动态分析。

##### 技术参数

1. Dex文件加密，防止反编译
2. SO库文件加壳，防止反汇编
3. SO库文件头信息擦除
4. SO库文件防内存Dump攻击
5. 基于源码的混淆技术

#### 客户端防篡改保护

##### 背景

客户端篡改是指向客户端内部添加或修改代码，添加或修改资源文件，再对客户端进行重打包投放到市场传播。例如向客户端内添加病毒模块、广告及推广类SDK等，对用户的使用造成了严重的安全威胁，因此，对客户端需要进行防篡改的保护。

腾讯安全为客户提供的客户端防篡改保护方案，采用对客户端所有文件进行完整性校验的方式，比传统的签名校验方案保护范围更广，并且更安全。

##### 技术原理

加固时对客户端内所有代码文件及资源文件进行正版Hash认证，并加密存储于保护壳内。运行时一旦检测到有代码文件或资源文件被篡改，可以及时阻止盗版程序继续运行。

##### 技术参数

1. 客户端文件Hash校验，防止被篡改
2. 支持对代码文件，资源文件，配置文件，数据库文件的防篡改

#### 客户端防注入保护

##### 背景

注入是指程序在运行时，攻击者通过ptrace或定制系统框架，将自己的代码运行在被攻击的程序空间内，可以窃取被攻击应用程序内存中的敏感数据或私有目录下的敏感文件，同时也可以改变原应用程序的执行流程。

##### 技术原理

常用注入程序在进行代码注入的过程中，存在特征，保护壳通过识别这些特征能够精确发现被注入的情况。同理，进程空间在被注入攻击代码后，会在程序空间残留痕迹，保护壳也会对这些信息进行检测。

##### 技术参数

1. 对dlopen返回值进行检测
2. 对map表数据进行检测

#### 客户端反调试保护

##### 背景

动态调试攻击指攻击者利用调试器跟踪目标程序运行，查看、修改内存代码和数据，分析业务逻辑，进行攻击和破解行为，导致敏感数据泄露，加密算法被破解，通讯协议被破解等风险。

##### 技术原理

当程序处于被调试状态时，无论从系统层面还是程序内存层面，都有相应的变化特征，腾讯安全提供的反调试方案就是在程序运行时，通过对这些多维度特征数据进行计算，判断当前程序是否处于被调试状态。

##### 技术参数

1. 检查TracerPid状态
2. 检测android\_server端口号状态
3. 利用Linux信号机制检测进程状态
4. 使用inotify接口检测proc内存读写操作
5. 检测函数执行时间

#### 客户端本地数据与资源文件加密保护

##### 背景

Android平台本身具有沙箱机制，应用程序之间不可跨沙箱访问其他应用程序的数据和文件。然而在Android平台，攻击者和防御者往往不在一个权限维度，Android手机在root后，攻击者可以穿透沙箱，直接读取任意应用程序私有目录的数据和文件，造成用户数据泄露。虽然本地数据存放在私有目录，也应该进行加密存储。

##### 技术原理

通过对Android文件系统层函数进行接管，保护壳内会拦截本进程内所有I/O操作，在读写文件时自动对写入或读出的数据进行加解密操作，对开发者与用户都是透明的无感知的。一旦本地私有目录内的文件被拷贝到其他地不在本进程内打开，由于应用程序无法得到自动解密的服务而无法打开，从而起到保护私有数据的作用。

##### 技术参数

1. 支持所有文件格式的数据加密
2. 特别针对移动端常用的Sqlite数据库进行加密优化
3. 加解密过程对使用者无感知
4. 加解密过程无需开发者参与

#### SO库绑定保护

##### 背景

SO库作为一种动态链接库，本身是可以被其他APK嵌入并进行调用。由于历史积累以及运行效率等问题，目前很多算法或协议的代码都是以SO库形式存在，攻击者只需要自己写一个新的APK，简单实现Java层调用代码，即可对算法或协议进行复用和攻击。因此，有必要对SO库文件做绑定运行的保护，脱离原始APK则无法正常运行。

##### 技术原理

SO库绑定技术的原理是将SO库的解密密钥放在保护壳中，并在保护壳解密原SO库时，对被保护程序的包名和签名特征进行二次校验，因此如果脱离了加固的APK，该SO由于无法取得密钥，以报名和签名特征的差异，从而无法正常运行。同理，如果加壳后的SO库被替换，由于保护壳会试图去解密SO库，导致解密失败，程序无法运行。

##### 技术参数

1. SO库文件解密密钥加密存储在保护壳中
2. SO库运行时检查包名与签名信息
3. 经过绑定保护的SO库文件无法被替换或篡改

#### 运行时环境检测技术

##### 背景

目前市面上存在多种Android模拟器以及多种虚拟环境框架，因其易用且功能强大，逐渐成为攻击者实施攻击行为的基础环境。所以对安全高度敏感的应用应该具备对运行时环境安全程度进行检测的能力。

##### 技术原理

腾讯安全对于模拟器的检测，是针对这几种模拟器存在的特征文件，以及必备的与主机通信的驱动文件进行检测，同时辅以X86 CPU与ARM CPU在执行指令时的差异进行检测，如指令Cache和任务调度特性等。对于虚拟环境

的检测，原理是对特定系统文件特征，程序内部映射的系统文件，程序当前路径的字符串特征进行检测，同时扫

描应用安装列表信息中是否存在框架特征APK进行辅助检测。

##### 技术参数

1. 可检测模拟器包括：GoogleAOSP模拟器，Genymotion，BlueStacks，腾讯手游助手，网易mumu，逍遥模拟器，夜神模拟器，51模拟器，靠谱助手，新浪手游助手，星光模拟器，9981模拟器
2. 可检测虚拟环境包括：Xposed，Substrate，VirtualApp，VirtualXposed，Magisk

## 管理系统

### 技术架构

### 应用管理

#### 我的应用

我的应用模块，是对关联了该用户所在的用户组的应用进行管理。

1）点击应用列表管理按钮，进入应用管理页面。

2）点击管理按钮，进入对应应用的产品服务总览页。

3）点击应用配置下载按钮，进入对应应用的应用配置下载页。创建应用后，需要先下载应用配置文件。

在应用配置下载页填写相关信息后，点击下载iOS配置和下载Android配置按钮，即可分别得到iOS和Android的配置文件。

注意：

1. 如果需要使用活体检测功能，要联系商务经理申请。
2. 点击二维码icon，会弹出二维码，使用demo扫描该二维码后可以直接访问。

#### 应用管理

应用管理模块，是对应用的增删查操作。

1）新增应用：创建一个新的应用，需要填写基本信息（应用名、备注）并关联用户组。

2）关联用户组：为应用关联用户组。

3）删除应用：对于下线的应用进行删除。

注意：删除应用后，该应用所有数据都会被清除，请谨慎删除。

4）查看应用：点击具体的应用名，跳转到该应用的详情页面，查看该应用的详细信息。还可为应用关联用户组和解除关联。

注意：解除关联后，用户组将失去该应用的操作权限。

### 通知管理

通知管理记录系统每次发送的消息，并可根据消息内容搜索消息。

1）查看消息：点击具体的消息，跳转到该消息的详情页，查看该消息的详细信息。

2）删除消息：点击删除按钮，弹出提示框，点击确定，即可删除消息。

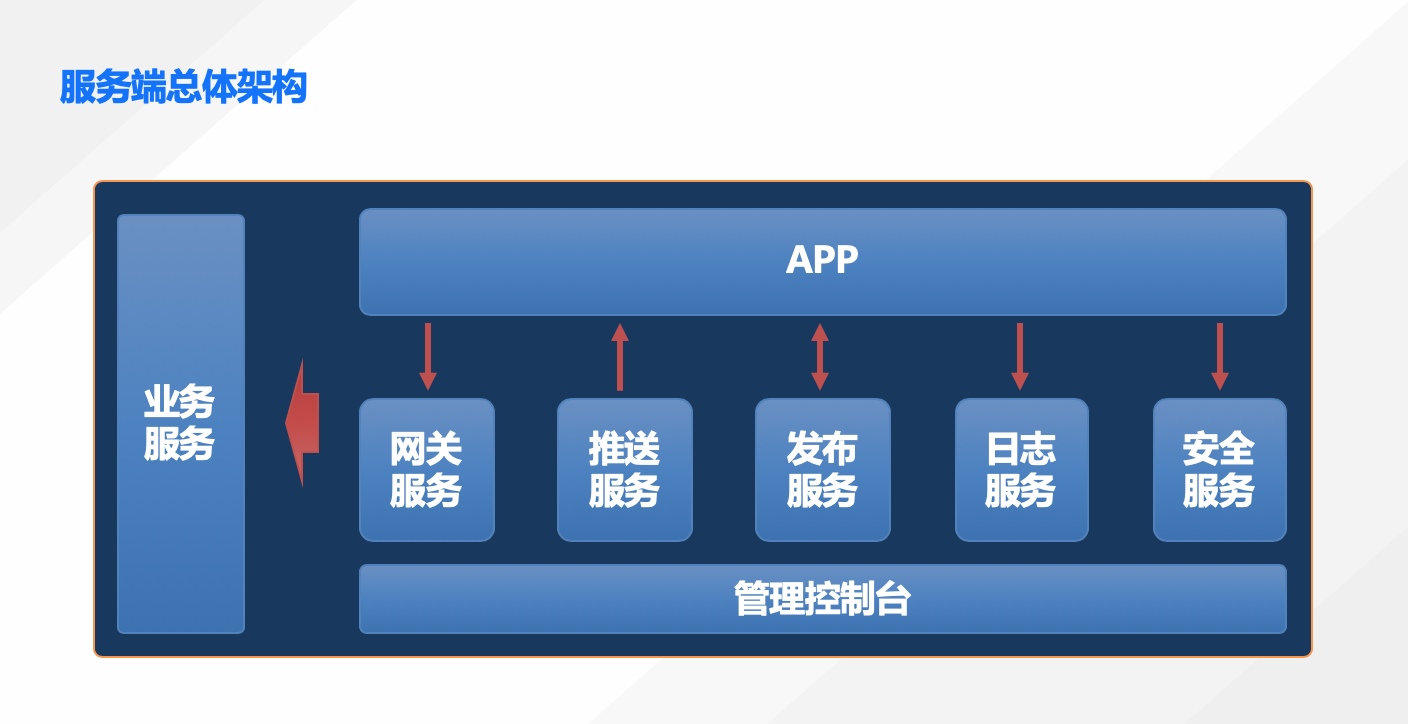
### 审计日志

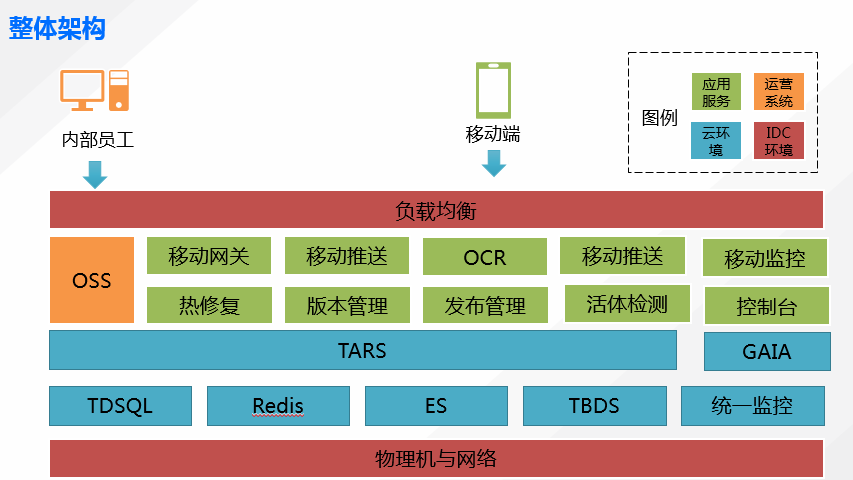
移动金融开发平台提供了用户每次操作系统的记录，并可根据关键字、所属服务、登录账号、日期来查询日志。

1. 可点击左上角的下载按钮，下载筛选出来的日志。
2. 查看事件：点击查看事件按钮，查看该日志的详细信息。

# 部署方案

移动金融开发平台的服务总体架构如下，后端主体服务组件包括：网关服务、推动服务、发布服务、日志服务、安全服务。





移动开发平台基于容器平台搭建，容器化部署，可根据实际运行需求进行弹性伸缩。

网关服务为移动App提供统计的中台接入服务，具体功能详情可详见网关功能描述章节。移动的业务流量会经过网关，由网关经过预处理后再根据情况传到后端业务服务。

推送服务为推送相关功能(包括消息推送、热修复、离线包、移动发布等)提供服务，具体功能详情可详见消息推送、热修复、离线包、移动发布功能描述章节。推送服务会根据推送需求将消息推送到移动App端。

日志服务为移动开发人员和运维人员提供移动监控和移动分析功能，移动监控为开发员工提供App运行数据，移动分析为业务运营人员提供业务埋点数据，详细功能见移动监控和移动分析相关描述章节。

安全服务为移动开发人员提供加固、环境安全等安全服务。