移动开发平台

动态发布

产品文档

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2023 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

腾讯云

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100。

文档目录

动态发布
热修复
离线包
应用发布

版权所有：腾讯云计算（北京）有限责任公司

动态发布
热修复

最近更新时间：2023-07-31 09:43:23

服务概述

热修复服务提供紧急修复线上 bug，无需重新发版即可在用户无感知的情况下完成原生代码问题修复的能力。热修复服务支持多种修复策略设置，包括但不限于修复版本、修复范围等。热修复组件拥有成熟、完整的流程，支持热修复补丁的下发、暂停、灰度、修复、回滚等流程。发布平台可以实时监控热修复补丁下发的阶段，并具备完备的漏斗统计，可以实时统计补丁触达率等。

> **注意：**
> 请开发者注意限制该热修复能力，避免使用此能力擅自更改 App 的主要功能。

- **Android**
  修复是基于用户已安装的母包和新包之间进行 diff 生成补丁，通过推送系统动态的将补丁下发到用户手机上，然后和母包进行合并和加载，实现新功能动态加载到母包上的效果，从而达到动态修复线上问题的效果。
- **iOS**
  在 iOS 开发领域，应用迭代周期长且发版速度慢，当线上遇到严重问题时是无法通过发版渠道进行快速修复的。因此对于 iOS 应用来说，能快速修复线上严重问题而无需发版的热修复方法是非常重要的。TMF iOS 热补丁提供基于 JSPatch 的热修复方法，在利用 Objective-C 运行时特性的基础上，通过 JavaScript 原生代码替换原有的方法实现，从而修复线上的问题。

应用场景

- 热修复组件可在线上快速推送修复补丁到终端进行修复操作。
- 热修复组件可以通过下发功能性补丁包来完成轻量级业务修改，完成线上快速更新。
- 热修复组件可以通过下发更新性补丁包来完成轻量级运营或新特性灰度。

产品优势

Android

TMF 是 Android 的热补丁解决方案，支持动态下发代码、So 库以及资源，让应用能够在不需要重新安装的情况下实现更新。

类型	TMF 热修复
类替换	支持

So 替换	支持
资源替换	支持
全平台支持	支持
即时生效	重启进程生效
开发透明	支持

- 支持全面、开发透明，补丁包小，性能损耗低。
- 全量替换新的 dex（无需插桩，没有 art 地址错乱问题，无兼容问题，开发透明）。
- 基于 diff 算法新旧 dex 差异计算得到补丁包，体量小。
- 基于自研 dexdiff 算法，利用 dex 格式进一步优化包大小。
- 支持完善（Gradle 接入，进程管理，安全校验等）。
- 海量验证（腾讯系亿级海量用户验证）。
- 精准触达，十分钟内即可获取推送质量的数据反馈，一天内触达率高达90%以上。

iOS

以下是热修复业界核心方案对比，TMF 采取 JSPatch 的热修复核心方案：

工具	Wax	JSPatch
引擎	Lua	JavascriptCore，iOS 原生支持，效率最高
补丁语言	Lua	Javascript（JSPatch）
系统支持	第三方库，iOS 所有版本	>=iOS 7
方法调用	va_list，效率高，但存在64位问题	forwardInvocation，方案无位数限制
修复语言支持	ObjC	ObjC + C
block 支持	支持浮点类型，安装包增量大	不支持浮点类型
开发工具	Lua 开发工具，有代码提醒，要 import 大量文件，包增大	JSPatch 开发插件，有代码提醒 JSPatch 语言转换工具
开源情况	开源，多年前停止更新，有较多第三方修改版	开源，使用用户多，收集和解决问题迅速

TMF 热修复下发方案的优势

- 强大的下发管理：

  - 支持热修复下发和版本管理。
  - 支持多补丁下发。
  - 支持热修复中途暂停、回滚等回退操作，具有强大的可控能力。
  - 支持补丁灰度下发，热修复下发前可以充分测试补丁，更好的确保线上正式版安全。
- 高到达率：
  - 利用 TMF 数据同步框架，终端框架利用推拉结合。
  - 尽管 iOS 系统对 App 后台限制严苛，从平台下发热补丁后，6 小时覆盖率高于 70%，12 小时覆盖率高于 90%。
- 修复能力强：
  - 支持 Objective-C 语言的 runtime 钩子。
  - 支持 C 语言的 Fish Hook 。
- 安全性与完整性：

  使用签名验证补丁包，本地加密存储保证补丁安全 。
- 实时监控：
  - 客户端针对补丁的触达、下载、回滚等操作进行上报。
  - 可实时监控完整的下发链路，掌握补丁应用情况。

离线包

最近更新时间：2022-10-27 16:29:57

服务概述

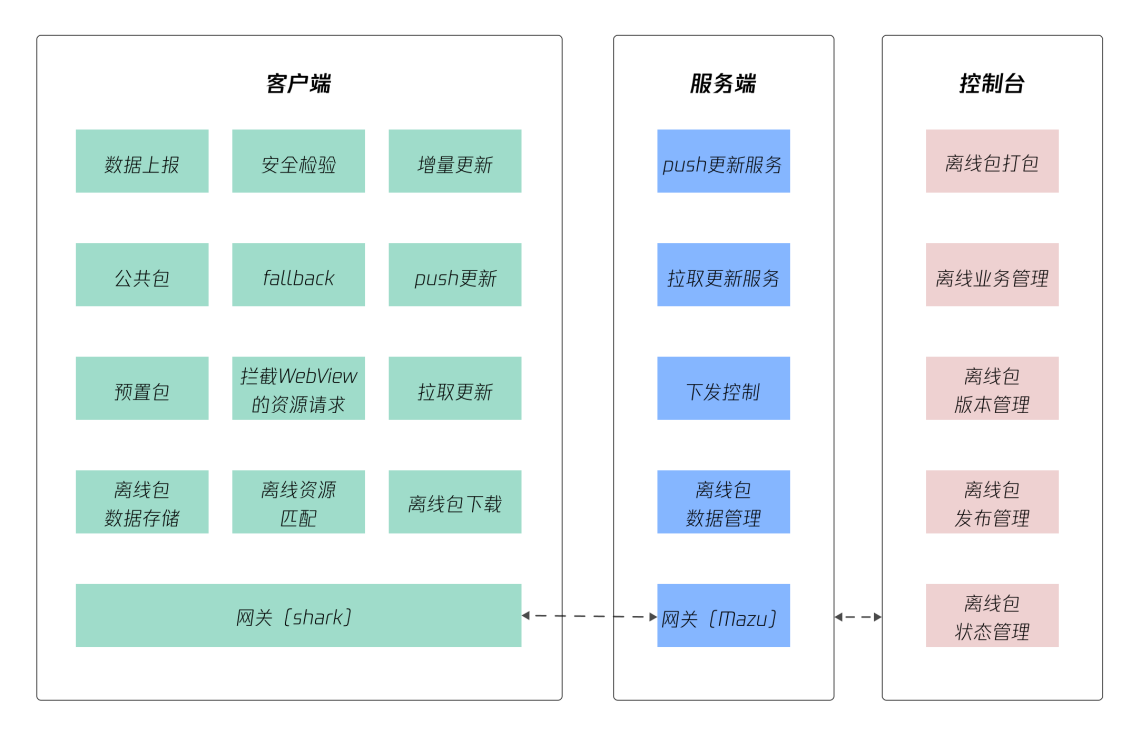
通过 TMF 的离线包组件，可以将网页的静态资源（html、js、css、图片等）缓存在本地，当用户访问 H5 页面时，这些资源都不需要从服务器即时下载，离线包组件会拦截 WebView 的资源请求并直接读取本地的离线资源，达到“瞬间秒开”的效果。
TMF 控制台实现了离线包的自动打包、状态管理和发布功能，支持推拉结合的更新方式及增量更新机制，可按多种维度进行灰度下发，且能便捷地按 H5 的版本采集灰度数据，对不同版本的业务数据进行对比分析。对于重要业务（如一级页面），还可以直接将离线包预置在发布包中，首次进入页面无需下载即可使用，大幅提升 H5 业务的用户体验。当本地离线包没有准备好时，则直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

- 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。
- 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。
- 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

TMF 离线包组件具有以下优势：

- 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。
- 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。
- 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。
- 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。
- 支持多维度下发：借助 TMF 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。
- 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。
- 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。

应用发布

最近更新时间：2023-07-31 09:43:23

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 TMF 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，可通过发布灰度任务进行功能验证后，再进行全量发布完成升级任务，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。

- 灰度发布：可以通过自定义部分用户下发规则来做小规模发布（例如内部员工）以验证新包的功能是否达到预期。
- 全量发布：当灰度发布流程均正常完成时，可进行全量任务发布可进行全量用户推送。

产品优势

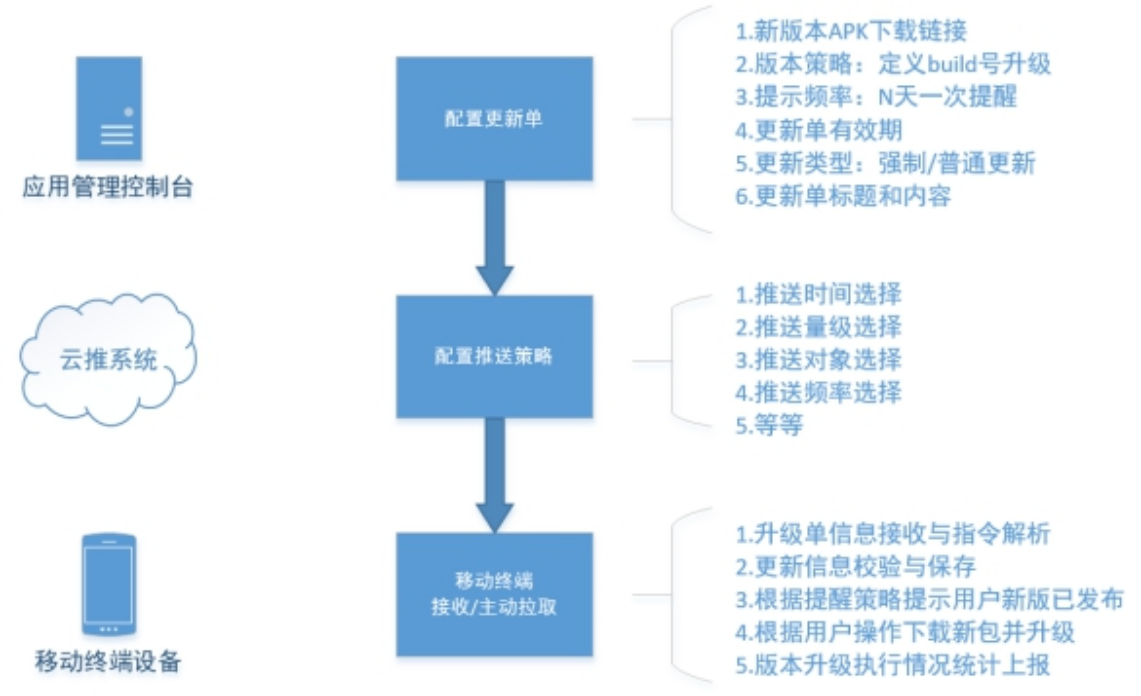
- 应用发布组件提供从控制台配单到终端回调 APK 升级的全过程，从升级策略、推送策略到发布策略，均实现可按需自定义配置，实现了新版本发布功能的高灵活性。
- 发布策略可选择发布灰度任务、全量任务，满足开发测试过程、小范围灰度检验的需求，可在灰度发布验证正常之后发布全量任务，实现发布功能的高稳定性。
- 依赖云推系统的统一推送配置，实现高到达率和高及时性，同时统一实现统一回收数据，实现对升级单下发情况和终端执行情况的数据监控。
- 终端 SDK 通过实时监控 + 定期检测的方式实现对升级单的高执行率，通过对升级单的二次检测，确保升级单的有效性，即云端停单后，终端缓存的升级单也同时失效。

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。

- 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。
- 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。
- 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：


应用管理控制台
云推系统
移动终端设备
配置更新单
配置推送策略
移动终端
接收/主动拉取
1.新版本APK下载链接
2.版本策略：定义build号升级
3.提示频率：N天一次提醒
4.更新单有效期
5.更新类型：强制/普通更新
6.更新单标题和内容
1.推送时间选择
2.推送量级选择
3.推送对象选择
4.推送频率选择
5.等等
1.升级单信息接收与指令解析
2.更新信息校验与保存
3.根据提醒策略提示用户新版已发布
4.根据用户操作下载新包并升级
5.版本升级执行情况统计上报