特别提示：

* 本文件必须按照“必须知道，最少授权”的原则交付，不得全部或部分提供给任何未经授权的人。
* 如果本文件不在您的访问权限范围内，应立即销毁，不得存储、传输、复制、打印或使用本文件的任何内容。

移动开发平台

移动框架详细设计说明书

2023年03月27日

文档修订记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | \*变化  状态 | 修订说明 | 日期 | 变更人 | 备注 |
| V0.1 | A | 新建 | 2023/03/27 | persilee |  |
| V0.2 | M | 修改基本样式 | 2023/03/29 | persilee |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

\*变化状态：A——增加，M——修改，D——删除

目录

[1 引言 2](#_Toc1721526335)

[1.1 编写目的 2](#_Toc645936314)

[1.2 背景 2](#_Toc721793813)

[1.3 术语定义 2](#_Toc53493188)

[1.4 参考文档 2](#_Toc1411846270)

[2 Flutter（Android/iOS） 3](#_Toc1353444187)

[2.1 系统功能 3](#_Toc1189661885)

[2.1.1 功能架构 3](#_Toc1574547625)

[2.1.2 功能概述 4](#_Toc2128435041)

[2.2 系统设计 5](#_Toc1972626008)

[2.2.1 整体结构设计 5](#_Toc1072774070)

[2.2.2 目录设计 6](#_Toc1988577925)

[3 小程序 8](#_Toc741187214)

[3.1 系统功能 8](#_Toc1728353098)

[3.1.1 功能框架 8](#_Toc1566708764)

[3.1.2 功能概述 8](#_Toc1377200681)

[3.2 系统设计 9](#_Toc1033098201)

[4 H5容器 10](#_Toc876178212)

[4.1 系统功能 10](#_Toc631841605)

[4.1.1 功能框架 11](#_Toc55220820)

[4.1.2 功能概述 11](#_Toc383386236)

[4.2 系统设计 12](#_Toc1121527452)

# 引言

## 编写目的

本文档是详细说明了用户中心的整体架构、功能列表、业务流程设计、接口设计、数据库设计和与其它系统的接口关系。在项目实施阶段，将依据本文档进行系统的开发。

## 背景

移动开发框架主要为银行、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框架。目前由于项目越来越多，衍生出一系列问题，如：每个项目都存在差异、代码差异、版本差异、代码不易维护，版本不易维护等，移动开发框架使用MVVM的架构来解决代码不易维护问题，封装统一标准的组件通过组件平台管理，来解决代码和版本差异问题，从而提高研发效能。

## 术语定义

| **术语** | **定义** |
| --- | --- |
|  |  |

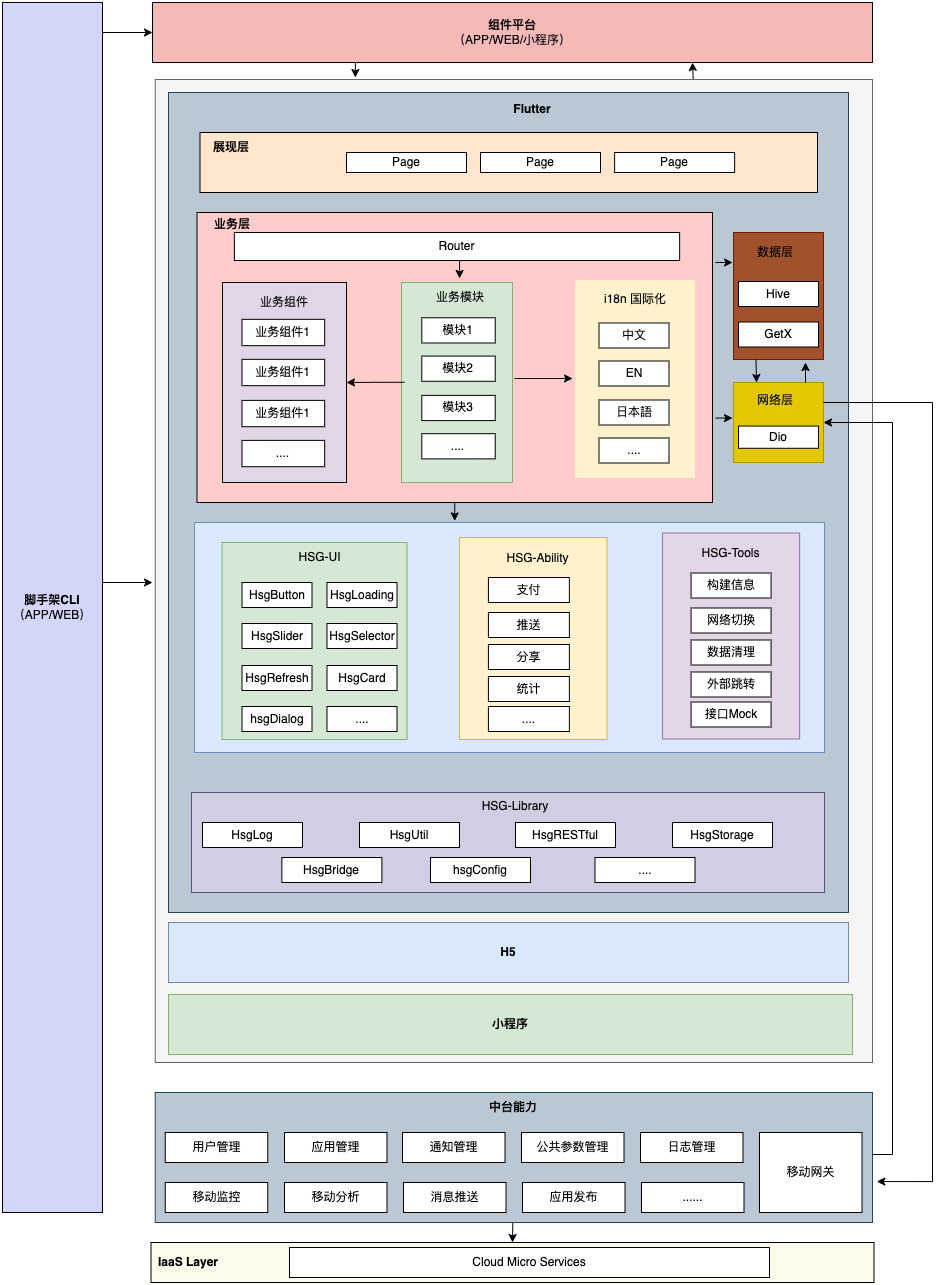
## 参考文档

1. 移动开发平台概要设计文档 v0.2.docx

# Flutter（Android/iOS）

## 系统功能

### 功能架构

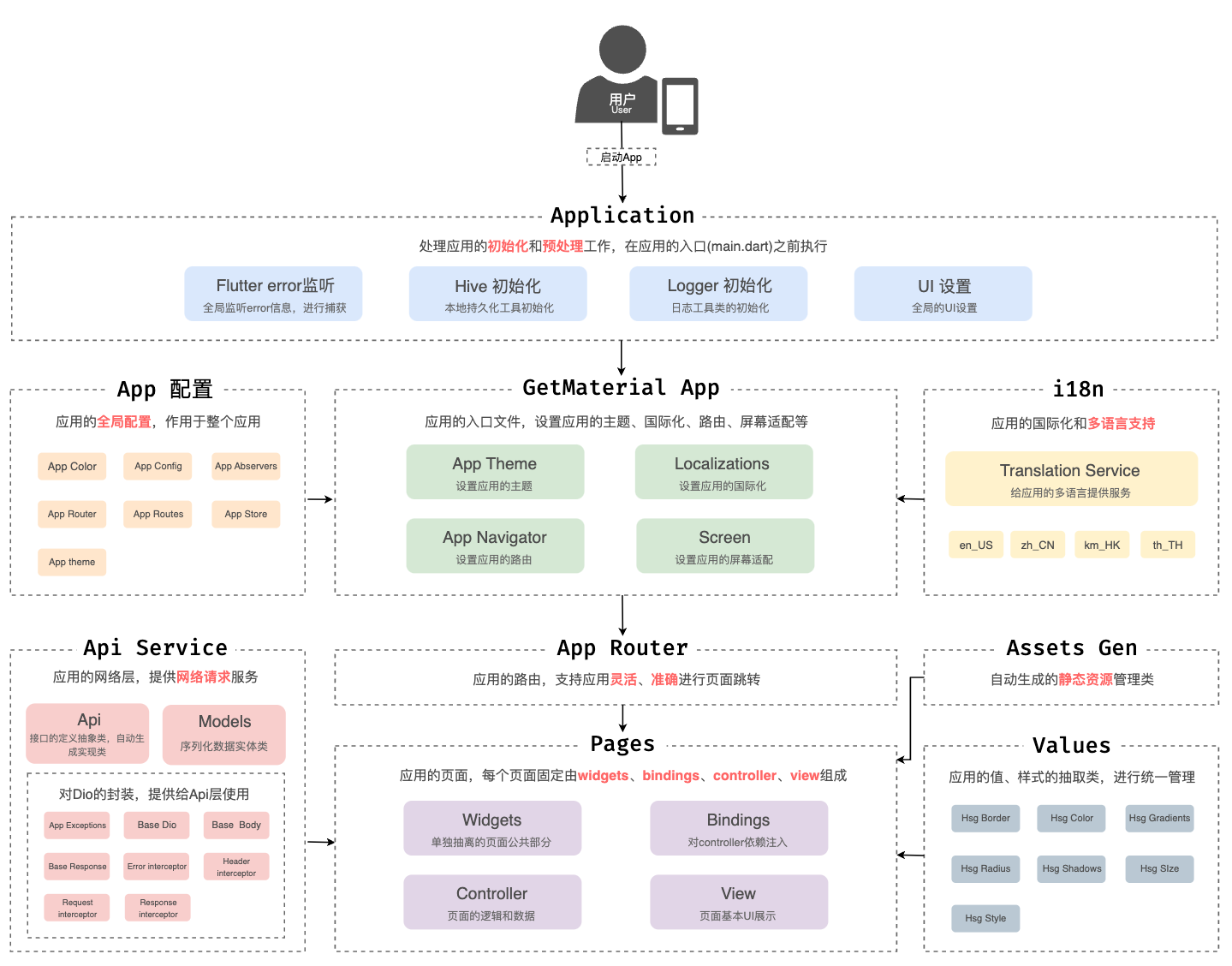


### 功能概述

移动开发框架的Flutter（Android/iOS）主要由展现层、业务层、数据层、网络层4部分组成，具体功能如下：

* **展现层**：由UI组件、业务组件和功能组件等，通过网络层和数据层的服务得到的动态页面，展现给用户的页面，主要负责和用户交互，完成用户的业务操作。
* **业务层**：由UI组件、业务逻辑、页面路由、国际化等组成业务组件；业务组件主要处理展现层用户的交互逻辑及信息和数据。
* **数据层**：由Getx处理应用和用户交互的全局数据，完成全局数据共享，实现国际化、主题切换等功能；由Hive处理应用和用户交互的数据持久化，实现应用访问加速、离线访问等。
* **网络层**：由Dio、Retrofit等组成，通过统一的封装后，可以通过简单的几行代码可以定义网络接口，通过简单的一行代码可以调用网络接口获取数据。
* **HSG UI**：主要是应用开发中常用的组件，经过统一封装，存储在组件平台进行统一维护和管理，提供给应用的业务层使用。
* **HSG Ability**：主要是应用开发中的一些功能性组件，如：支付、推送、分享等；经过统一封装，存储在组件平台进行统一维护和管理，提供给应用的业务层使用。
* **HSG Tools**：主要是应用开发中常用的一些工具方法和工具类，如：时间格式化、金额格式化、加密、解密等，通过统一的提取和封装，形成工具包，存储在组件平台进行统一维护和管理，提供给应用的业务层使用。
* **HSG Library**：主要是应用开发中基础功能服务，进行提取、总结和封装成功能库，如：日志管理库、网络请求库、本地存储库等。

## 系统设计



### 整体结构设计

架构整体结构设计使用了MVVM的设计思想，使用Getx的Controller来使业务逻辑单独剥离出来，view主要UI界面的展示。

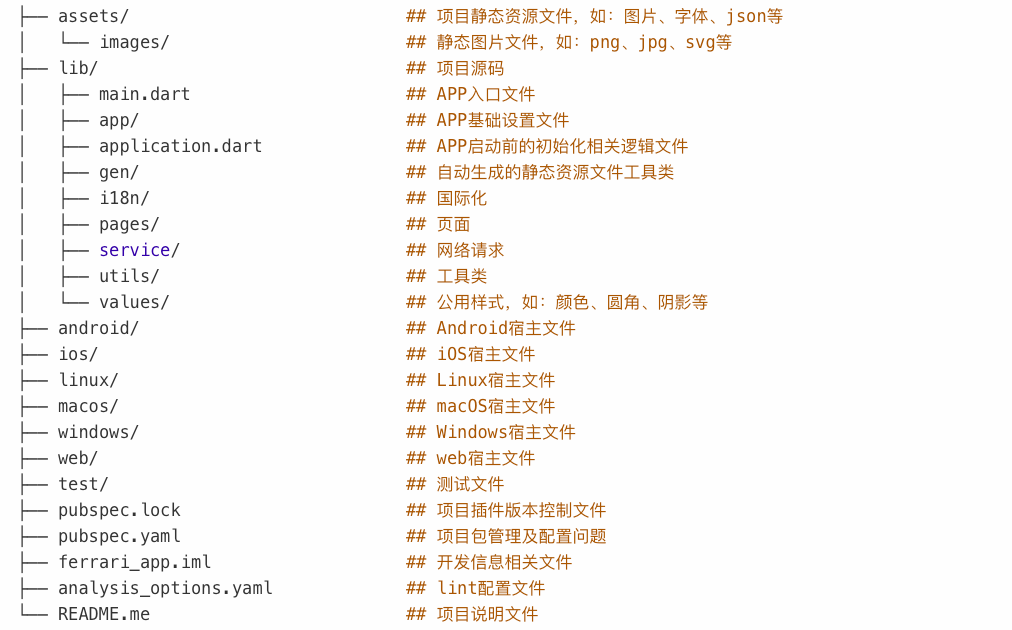
在应用启动之前，会通过Application统一处理应用的初始化和预处理工作，如：全局异常监听捕获、持久化工具初始化等。

通过GetMaterialApp设置应用的主题、国际化、路由、屏幕适配等。

通过Getx的路由管理机制来实现灵活的页面路由跳转。

通过Getx的Translation Service实现应用的多语言切换的国际化功能。

### 目录设计



#### lib/pages 目录

项目的每个页面的基本固定结构(之后会用脚手架自动生成)，包含 widgets、bindings、controller和view组成。



#### lib/service 目录

项目的网络请求部分，是基于axios 封装的，封装思想来源于retrofit库，只需要通过简单几行代码就可以定义和使用，详情如下：

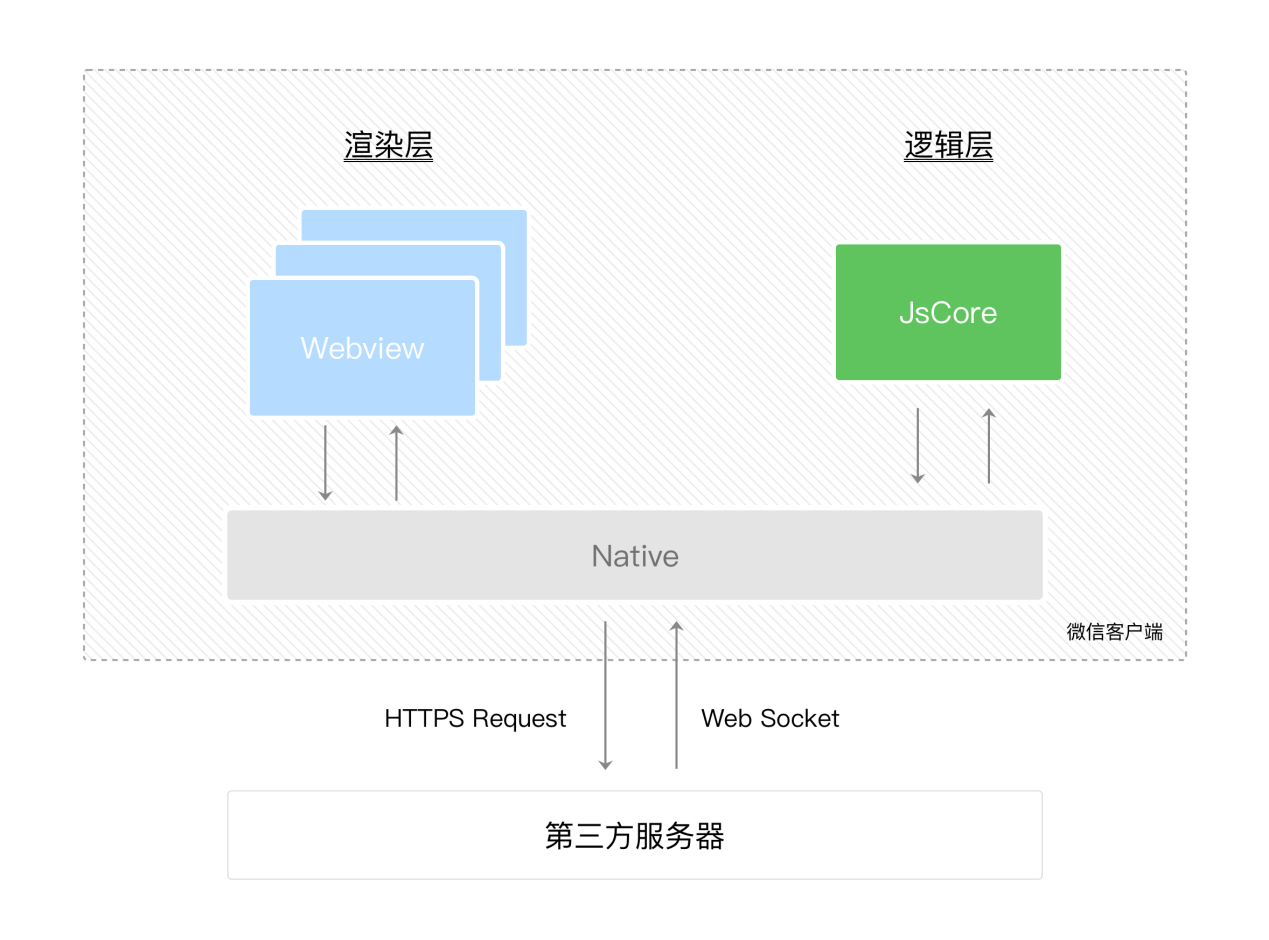


阿斯顿发阿斯顿发生

# 小程序

## 系统功能

### 功能框架

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。

### 功能概述

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。

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。

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

## 系统设计

# H5容器

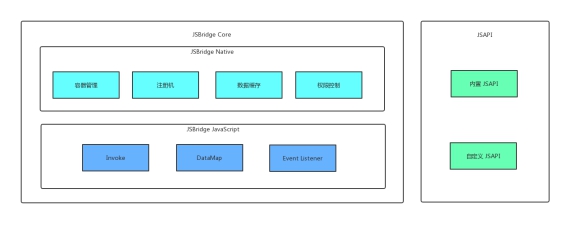
## 系统功能

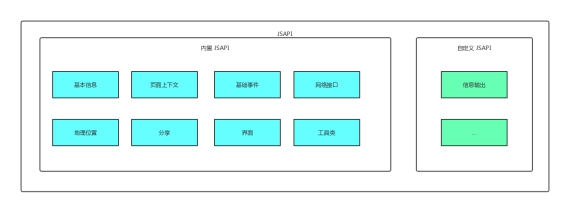
H5容器组件，提供了成熟的移动端 Hybrid 解决方案，拥有丰富的内置功能，也提供了良好的外部扩展功能，具备单/多容器管理、单/多页面管理、事件透传、信息互换、埋点上报等基础能力，并提供插件定制 JSAPI。

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。

* 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

### 功能框架





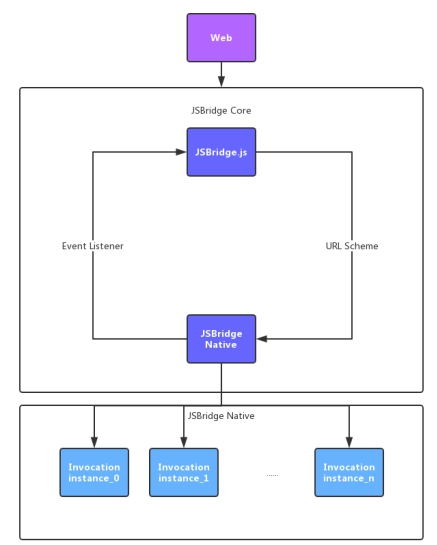
**H5 容器的架构分为两个部分：JSBridge Core 和 JSAPI**

* **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。
* **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。

### 功能概述

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。

## 系统设计



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：

* H5 调用 Native 功能
* Native 调用 H5 功能
* 注册与权限控制
* 多容器管理
* 自定义 JSAPI