

客户端配置系统

一、项目基本情况

客户端配置系统是为客户端（手机端）提供一套管理和获取配置的解决方案。传统配置中心的配置内容都是静态的纯文本数据，每个客户端拉取到的内容都是一样，这里要求提供一套动态计算配置的机制，可以根据客户端的参数，返回不同的内容，为每个客户端定制个性化的数据。

同时要求管理端能够集中管理多个应用的配置，具备完整的发布流程，完善的权限管控和安全审计。

举个例子：

某个活动的首页需要展示一组图片，要求可以按需更换图片内容。存在有两种做法：

1. 简单处理：为该活动创建一个配置，配置内容返回一个图片链接的列表，客户端根据图片链接进行展示即可。
2. 高级做法：同样是为该活动创建一个配置，但是配置内容会根据客户端(用户)的特性（比如年龄、性别、兴趣爱好等多维特征）为其返回特定的一组图片链接，比如针对年轻的用户，返回一组积极有活力的图片，针对老年人，可以返回一组通俗易懂，便于理解的图片。


以上两种做法，方法2的效果会优于方法1，这也是动态计算配置的优势。

除了以上提到的可以下发图片链接这种文本，也可以下发开关的bool类型，还可以是结构化数据。

平台演示：[📺 Settings演示.mov](#)

二、需求说明

1. 核心功能

	功能模块	说明	示意图	备注
管理平台	平台概览	<ul style="list-style-type: none">· 以web形式呈现· 平台简介· 业务接入情况<ul style="list-style-type: none">◦ 展示接入的应用数、配置数量以及发布次数等· 平台的核心能力· 平台公告（平台管理员能发布）<ul style="list-style-type: none">◦ 新功能、特性发布		<ol style="list-style-type: none">1. SDK 和 APP 是并行的两种类型，功能基本一致，学生可考虑实现 APP 即可2. 新建的配置默认处于生

	<ul style="list-style-type: none"> 重大事情公告 		效状态，如果不想生效需要通过按钮手动失效掉。
注册登录	<ul style="list-style-type: none"> 进入平台需要先登录（没有账号需要先注册） 		
应用管理	<ul style="list-style-type: none"> 应用注册 <ul style="list-style-type: none"> 提供一个页面，注册新应用 应用列表 <ul style="list-style-type: none"> 可以查看所有应用和我关注的应用 关注和取消关注应用 申请加入应用成员，由应用负责人/管理员审批 应用搜索 <ul style="list-style-type: none"> 可以根据应用id、名称、分组、管理员进行搜索 应用管理 <ul style="list-style-type: none"> 管理员设置 应用公告(只有应用管理员可以设置) 	 	<ol style="list-style-type: none"> 应用分组：每个App都归属于某个应用组，是多对一的关系。举例：今日头条、今日头条极速版、今日头条专业版等都属于 news_article 分组。 任何人都可以注册新应用，注册成功后注册者会成为应用负责人
配置管理	<ul style="list-style-type: none"> 配置新建 配置列表 <ul style="list-style-type: none"> 包括所有配置和关注配置的列表 配置管理 <ul style="list-style-type: none"> 展示配置基本信息（包括配置简介、状态、创建更新时间、负责人管理员等） 修改状态，失效或删除等 配置内容 <ul style="list-style-type: none"> 设置配置生效的条件 编写配置规则，可以用动态语言进行编写 需要草稿功能 单元测试 <ul style="list-style-type: none"> 编写测试case，对编写的代码进行输入和输出结果比对 	 	<ol style="list-style-type: none"> 图里的配置变量其实是指规则编辑区里的代码变量，也可以在规则编辑区直接定义 提交审核时，前端界面把所有有权限的人展示出来，由发布人选择特定人审核（只有选中的人能收到

	<ul style="list-style-type: none"> · 发布审核及通知 <ul style="list-style-type: none"> ◦ 提交审核，通知相应的人进行review ◦ reviewer 可以看到修改前后的diff · 配置发布 <ul style="list-style-type: none"> ◦ 查看发布的配置代码 ◦ 需要灰度发布，并且可以查看发布进度，发布过程过程中遇到客户端异常，可以快速回滚 ◦ 可以查看发布历史记录 		<p>审核通知，其他有权限之人不会收到通知，但仍然有权限审核)</p> <p>3. 灰度时可以根据客户端id (device_id) 进行哈希取模后按比例发布</p> <p>4. 灰度发布和快速回滚能力需要实现，客户端异常只要模拟即可，无需实现。</p>
权限管理	<ul style="list-style-type: none"> · 应用和配置需要有查看和编辑的权限控制 · 权限定义： <ul style="list-style-type: none"> ◦ 平台管理员 <ul style="list-style-type: none"> ▪ 超级管理员，具有数据库的操作权限 ◦ 平台成员 <ul style="list-style-type: none"> ▪ 只要注册了都属于普通成员 ◦ 应用负责人 <ul style="list-style-type: none"> ▪ 具备该应用所有操作权限 ◦ 应用管理员 <ul style="list-style-type: none"> ▪ 具有除修改应用负责人/管理员外所有权限 ◦ 应用成员 <ul style="list-style-type: none"> ▪ 可以查看该应用的任何配置的所有内容 ◦ 配置负责人 <ul style="list-style-type: none"> ▪ 具备该配置所有权限 ◦ 配置开发者 <ul style="list-style-type: none"> ▪ 只有修改配置代码和发布权限，不具备修改配置信息等权限 		
安全审计	<ul style="list-style-type: none"> · 操作记录 <ul style="list-style-type: none"> ◦ 任何变更都需要记录下来 		操作记录更全面，更细致

		<ul style="list-style-type: none">可以根据时间，操作人筛选记录		不仅仅包括了发布，还包括相关角色的变更等
服务端服务	python引擎	<ul style="list-style-type: none">实现计算<ul style="list-style-type: none">根据客户端传过来的参数运行管理平台的配置代码，并返回结果		
	接入服务	<ul style="list-style-type: none">提供多种访问方式：<ul style="list-style-type: none">具备http 和 rpc 两种方式可以拉取整个应用配置或者某几个配置的内容		
客户端	数据获取	<ul style="list-style-type: none">提供一个工具，可以读取某个应用的配置<ul style="list-style-type: none">全量配置指定配置		只要模拟客户端发起请求，并且得到正确配置即可。完整的功能可作为附加功能实现。
非功能性要求		每个接口耗时不能超过1秒		

权限表

操作\角色	平台管理员 (多人)	平台成员 (多人)	应用负责人 (一人)	应用管理员 (多人)	应用成员 (多人)	配置负责人 (一人)	配置开发者 (多人)	备注
发布平台公告	✓	✗	✗	✗	✗	✗	✗	
查看应用列表	-	✓	✓	✓	✓	✓	✓	
注册新应用	-	✓	✓	-	-	-	-	<ul style="list-style-type: none">任何人都可以注册新应用，一旦注册成功，默认成为该应用的负责人
查看应用信息	-	✗	✓	✓	✓	✓	✓	
关注/取关	-	✓	✓	✓	✓	✓	✓	

应用								
发布应用公告	-	✗	✓	✓	✗	✗	✗	
修改应用负责人	-	✗	✓	✗	✗	✗	✗	
修改应用管理员	-	✗	✓	✗	✗	✗	✗	
修改应用成员	-	✗	✓	✓	✗	✗	✗	
修改配置负责人	-	✗	✓	✓	✗	✓	✗	配置负责人默认属于应用成员
修改配置开发者	-	✗	✓	✓	✗	✓	✗	配置开发者默认属于应用成员
新建配置	-	✗	✓	✓	✓	✓	-	新建者默认成为配置负责人
删除配置	-	✗	✓	✓	✗	✓	✗	
查看配置	-	✗	✓	✓	✓	✓	✓	
关注/取关配置	-	✗	✓	✓	✓	✓	✓	
修改配置信息	-	✗	✓	✓	✗	✓	✗	
修改配置代码	-	✗	✓	✓	✗	✓	✓	
发起审核	-	✗	✓	✓	✗	✓	✓	
审核通过/不通过	-	✗	✓	✓	✗	✓	✗	
查看操作记录	-	✗	✓	✓	✗	✗	✗	

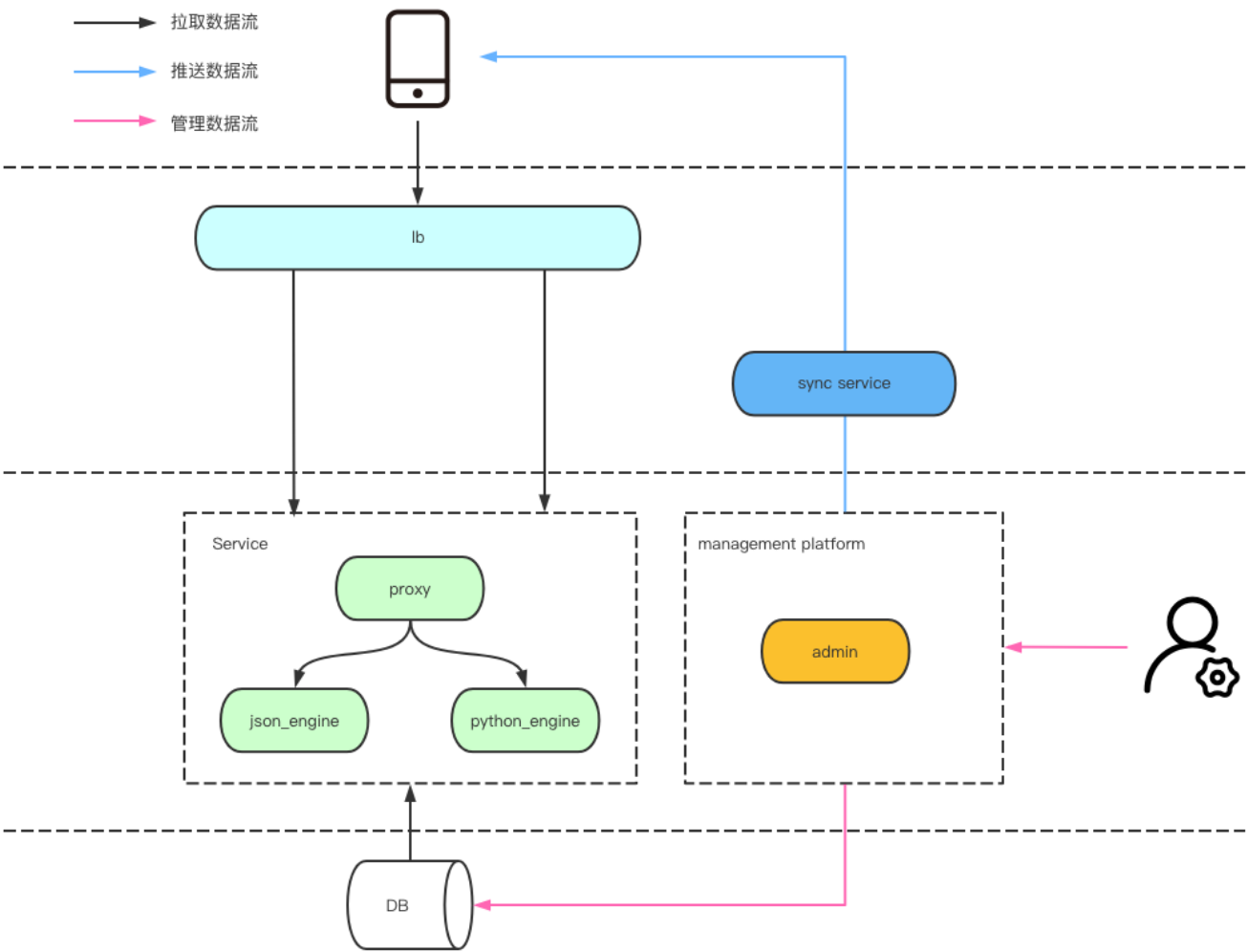
2. 附加功能（可选）

	功能模块	说明	示意图
管理平台	异常熔断	1、发布配置时，监控客户端运行情况，发现异常自动暂	

		停发布或者回滚发布	
服务端服务&客户端	引擎多样化	1、除了支持python代码编辑，可以再支持lua 等其他动态语言的计算引擎、以及json 规则引擎等	
	增量更新	1、客户端实现本地缓存机制，如果更新配置失败，则使用本地缓存，并提供统一访问接口 2、服务端实现增量更新的逻辑，每次只计算有变更的内容，同时客户端维护好本地数据	
	服务降级	1、实现系统自我保护，并最小程度影响客户端使用。提示：可通过缓存+降级+限流实现	
	秒级更新	1、配置有变更时，实时地推送到客户端，并非等客户端轮询拉取	

三、简单设计

示意架构图



架构分层

系统整体架构如图所示，主要包括以下模块和组件

1. **客户端SDK**：负责主动或定时拉取最新配置，维护本地文件，并且处理增量更新、降级策略等逻辑
2. **接入层（lb）**：主要是网关
3. **服务层（service、management Platform）**：分为接入服务、计算引擎
 - 接入服务：参数预处理以及结果合并
 - 计算引擎：包括json、python，但不限于这两种
4. **数据层（DB）**：存储全量的数据，包括应用信息、配置内容等。

数据流

数据流可以分为三种：

1. **管理数据流**：主要是由使用者在管理平台操作产生的行为数据流（包括修改配置，发布配置等），数据最终存储到数据库。
2. **拉取数据流**：由客户端发起，经过网关，到配置服务获取计算后的配置结果。
3. **推送数据流**：由使用者通过管理平台直接把配置内容推送到客户单。（属于附加功能的秒级更新）

注：

1. 管理平台的服务端和获取配置的服务端需分开实现

四、项目QA

1. 客户端是指手机端吗？可以通过网页端或PC端代替吗？
 - 本项目的客户端特制手机端（包括android 和 iphone），基本功能可以使用其他端代替，附加功能需要用手机端
2. 数据提供 @林景博

五、课程预备知识

<https://xie.infoq.cn/article/2593d1a3b9e1e06cac6502c4f>

github.com/sbinet/go-python

github.com/antonmedv/expr

<https://aws.amazon.com/cn/appsync/>

<https://mp.weixin.qq.com/s/WgTV7XcyTmJQkCLlQZi83Q>

六、项目信息

文档链接：[📖 客户端配置系统- 清华合作课程项目](#)

用户代表和助教的联系方式：

助教：

联系人：禹含

手机号：18083799220

邮箱：yuh21@mails.tsinghua.edu.cn

联系人：饶淙元

手机号：13661356081

邮箱：rcy17@mails.tsinghua.edu.cn

企业用户：

[@毛羽](#)

联系人：毛羽

手机号：18762674812

邮箱：maoyu@bytedance.com

[@胡振星](#)

联系人：胡振星

手机号：18640381641

邮箱：huzhenxing@bytedance.com

[@谢豪](#)

联系人：谢豪

手机号：18669051233

邮箱：xiehao_vnu@bytedance.com