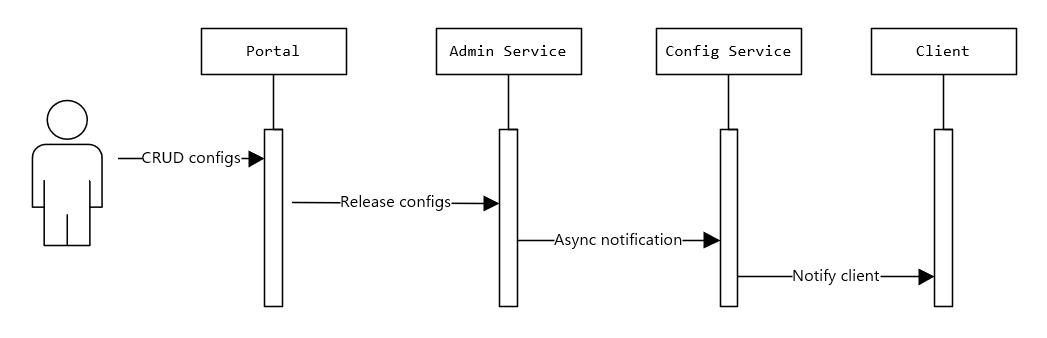


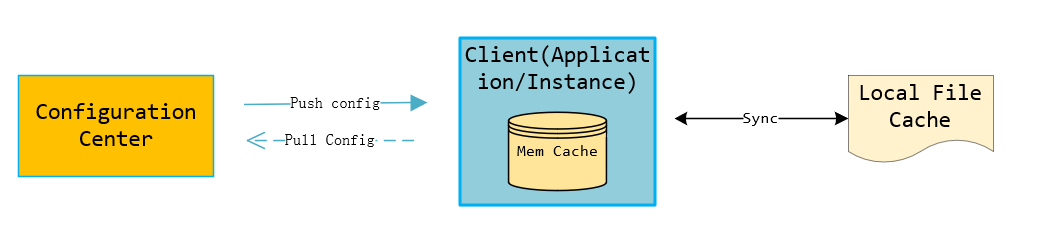
Server Design:

The process of real-time push after configuration release:



1. The user CRUD configurations and publishes on the Portal
2. Portal calls Admin Service API to release configurations.
3. After the Admin Service publishes the configuration, it sends a message to each Config Service
4. After the Config Service receives the message, it notifies the corresponding client

Client Design



1. The client and server maintain a long connection, allowing the client to receive configuration updates as soon as they are pushed (implemented through HTTP Long Polling).
2. The client also periodically pulls the latest configuration for the application from the configuration center server.

2.1 This is a fallback mechanism to prevent configuration from not being updated in case the push mechanism fails.

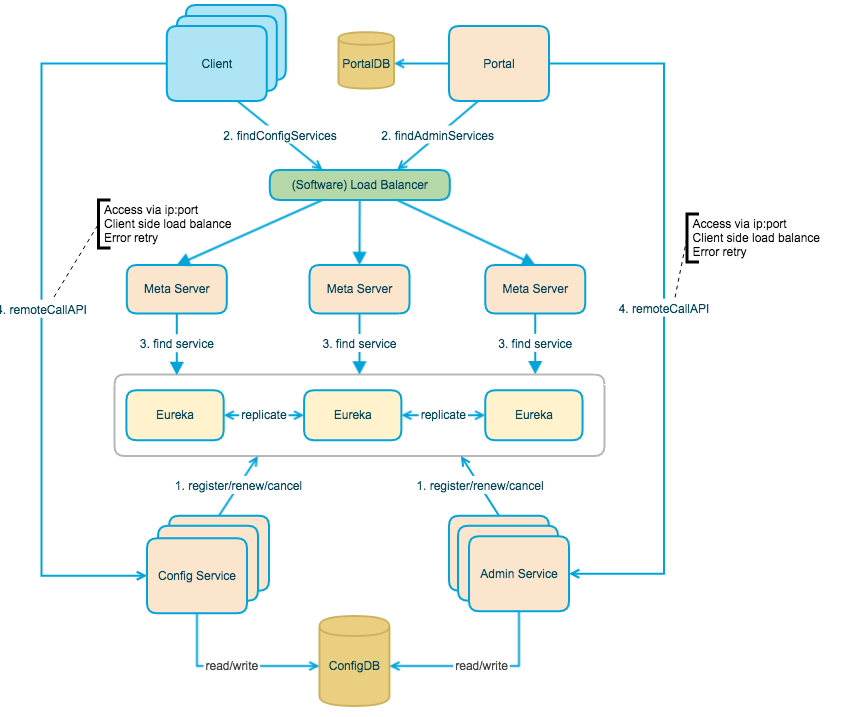
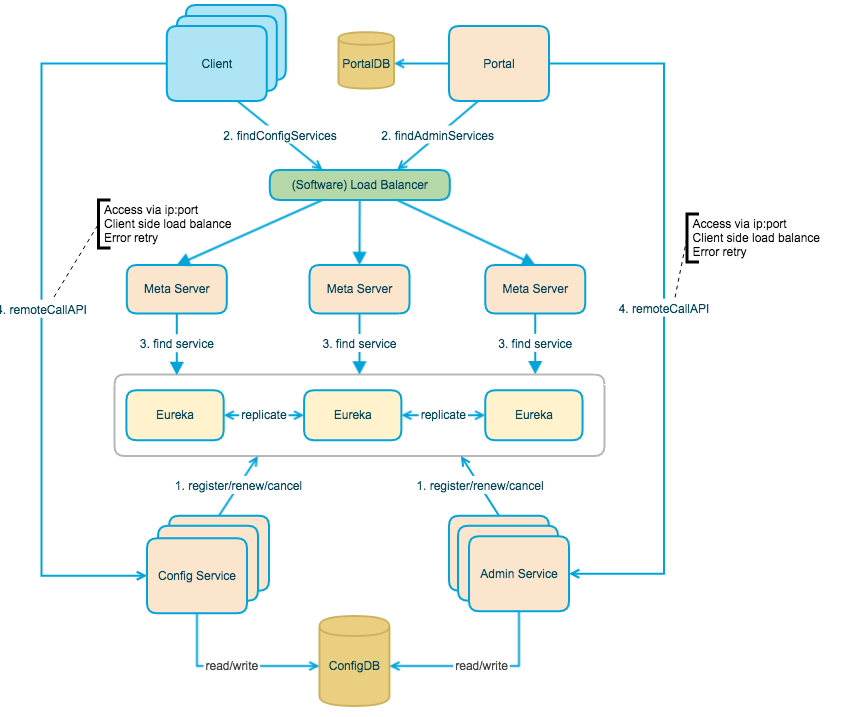
2.2 The client's periodic pull reports the local version

2.3 The default pull frequency is every 5 minutes

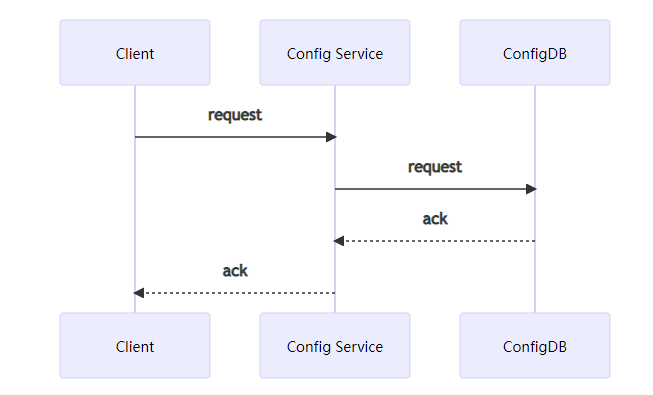
3. After obtaining the latest configuration for the application from the configuration center server, the client stores it in memory.

4. The client caches a copy of the configuration obtained from the server in the local file system. In case the server is unavailable or the network is down, the client can still recover the configuration from the local cache.

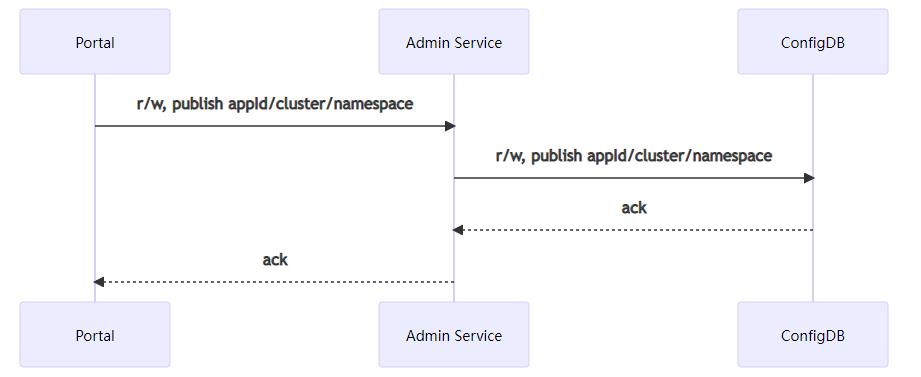
5. Applications can obtain the latest configuration from the Apollo client and subscribe to configuration update notifications

.上图简要描述了Apollo的总体设计，我们可以从下往上看：

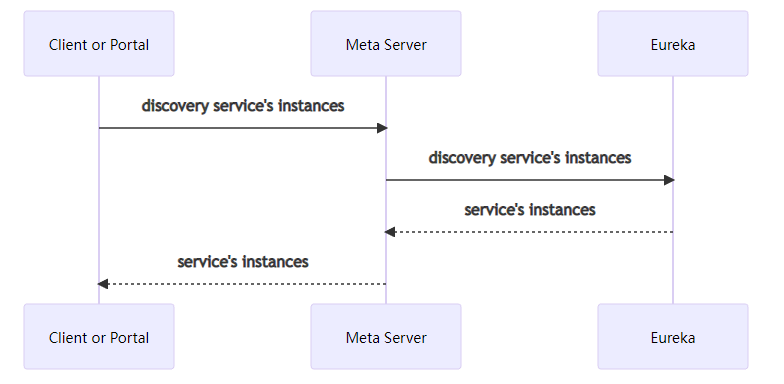
* Config Service提供配置的读取、推送等功能，服务对象是Apollo客户端



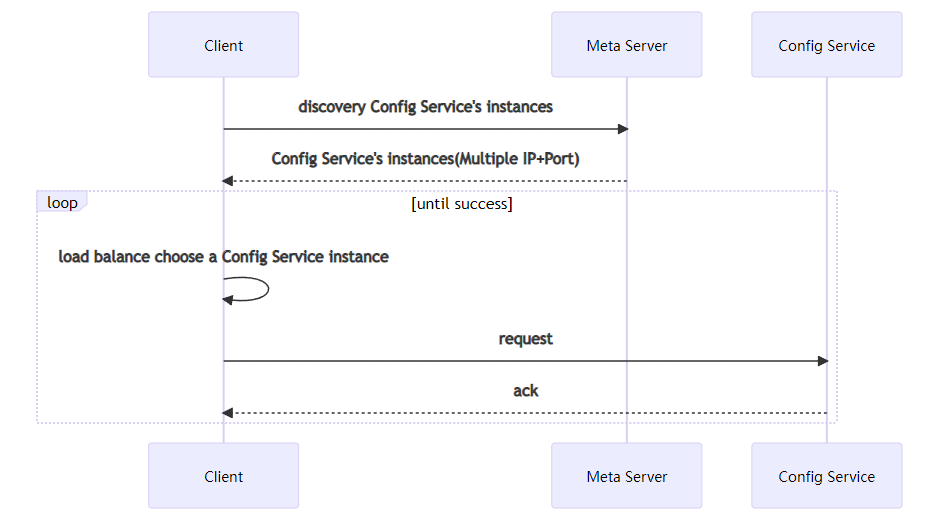
Admin Service提供配置的修改、发布等功能，服务对象是Apollo Portal（管理界面



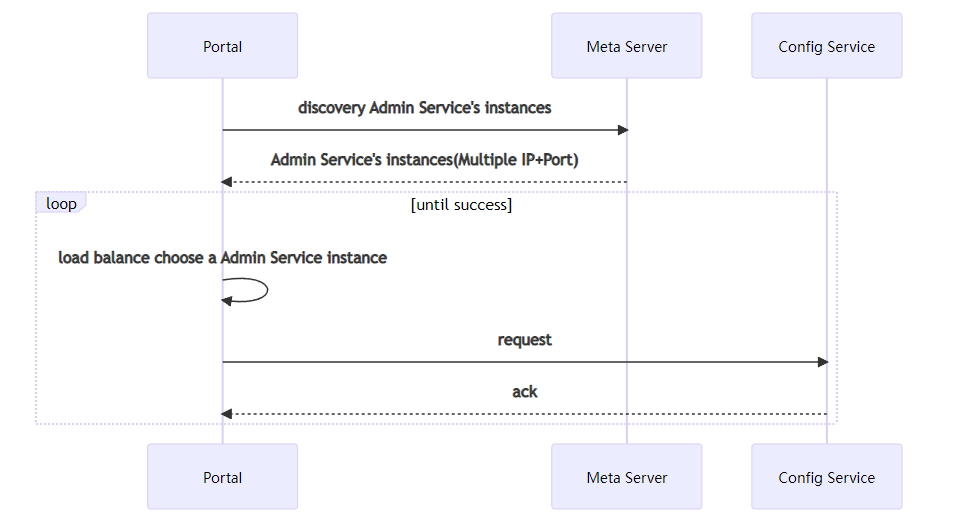
* Config Service和Admin Service都是多实例、无状态部署，所以需要将自己注册到Eureka中并保持心跳
* 在Eureka之上我们架了一层Meta Server用于封装Eureka的服务发现接口



* Client通过域名访问Meta Server获取Config Service服务列表（IP+Port），而后直接通过IP+Port访问服务，同时在Client侧会做load balance、错误重试



* Portal通过域名访问Meta Server获取Admin Service服务列表（IP+Port），而后直接通过IP+Port访问服务，同时在Portal侧会做load balance、错误重试



* 为了简化部署，我们实际上会把Config Service、Eureka和Meta Server三个逻辑角色部署在同一个JVM进程中

