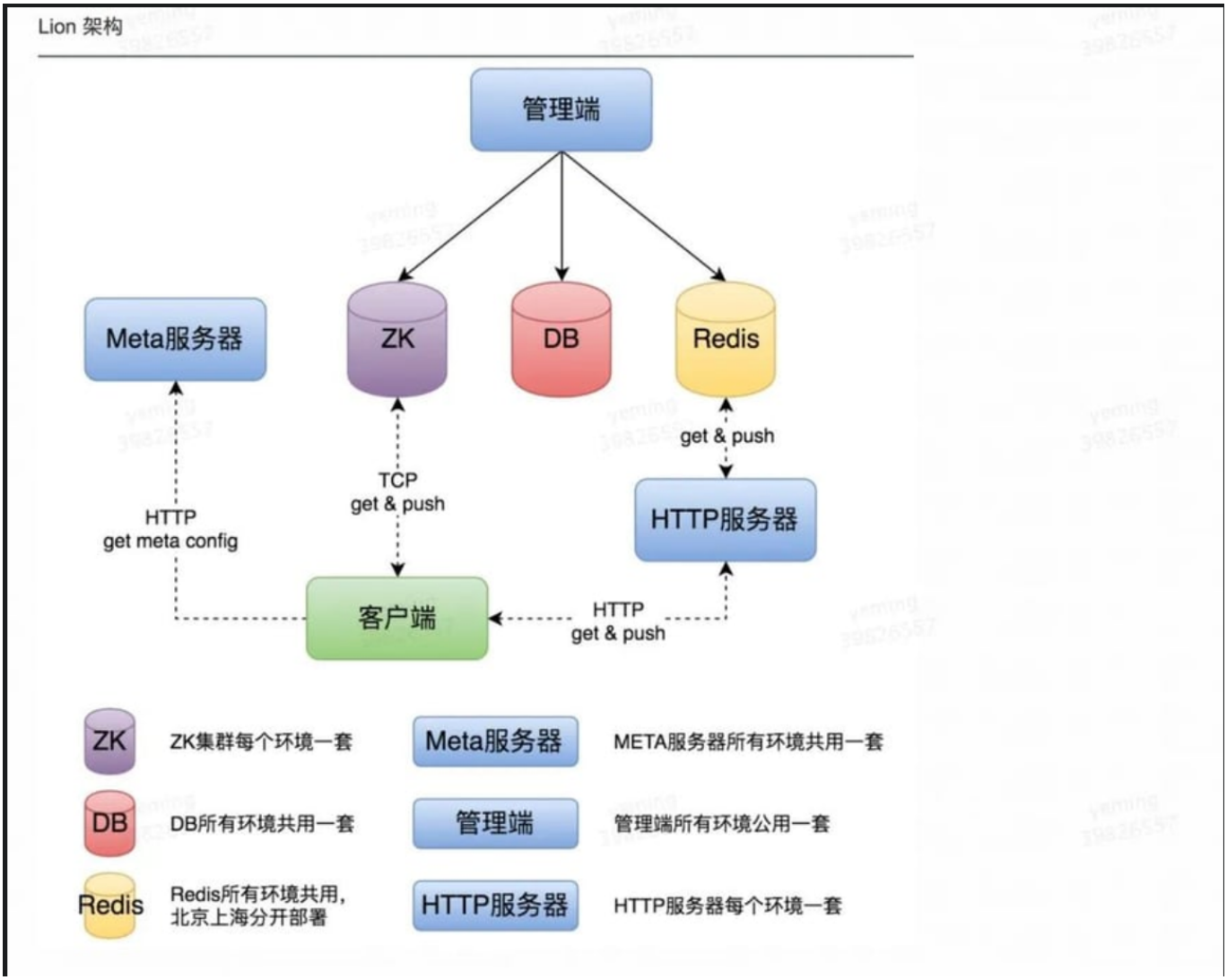
[TOC]

lion配置

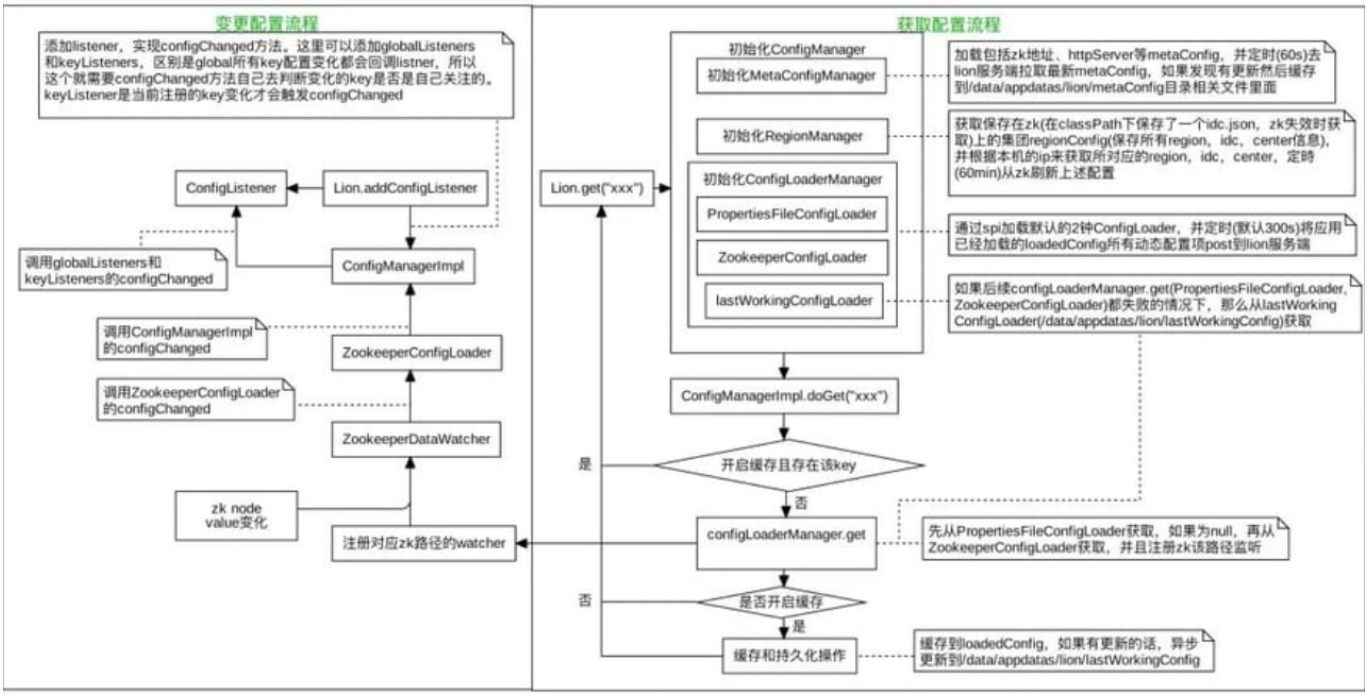
配置中心?

> 一个配置中心必须要做到能动态的获取配置参数，并且当配置发生变更了，能及时准确无误的获取最新的配置。 分布式配置中心(Spring Cloud Config)，在分布式系统中，由于服务数量巨多，为了方便服务配置文件统一管理，实时更新，所以需要分布式配置中心组件。

lion的架构图



获取配置和更新配置



获取配置：

- 本地缓存:先判断是否允许从本地缓存获取，如果可以，从缓存获取，缓存没有的话就从本地配置文件获取
- 本地配置文件:默认文件名applicationContext.properties，没有的话，就从zk获取
- zk:从zk获取，然后更新本地缓存和/data/appdatas/lion/lastWorkingConfig，并注册一个对该zk node的监听watcher。当然如果zk也失败， 就从机器/data/appdatas/lion/lastWorkingConfig获取，另外Lion后台会启动一些线程池来异步的同步或者上传数据。

更新配置

- 开始在Lion.get(“xxx”)的时候，如果最终是通过获取zk node value，则会注册一个对该node监听的watcher。当该节点的value发生变化时候，会更新该key对应的本地缓存和/data/appdatas/lion/lastWorkingConfig，最终会调用客户端Lion.addConfigListener添加的listener的configChanged方法。

lion的高可用

数据冗余

配置信息显然是很重要的，作为一个分布式配置中心，保存了那么多应用的配置信息，所以数据必须要持久化。我们配置信息不光存在zk上，上面也提到了，每个应用会在自己的/data/appdatas/lion/lastWorkingConfig下面保存应用用到的所有的配置，有多少个key，就有多少个对应的文件名，文件的内容就是key的value。另外后台会开启一个线程池lion-upload-stat-thread:定期把应用使用到的所有配置同步到lion-server端，估计可以作为zk挂了，本地应用磁盘挂了的另外一种备份吧

本地缓存

为了提高性能，减少不必要的访问zk，应用会缓存所有使用的key值，当然也不必担心数据不一致的情况。每次zk节点变化通知到客户端，客户端都会更新本地缓存

推拉结合

配置在发生变更时候，我们需要能及时的获取变更后的配置。目前是依赖zk的push，我们知道zk对网络比较敏感，有可能会发生zk的值变了，但是客户端没有收到通知。Lion客户端也会开启一个线程来定时主动去从zk pull最新的值。

lion的使用