首先是appId的概念（app包括部门信息，负责人——管理app配置的人）

appId 一对多 namespace

appId 一对多 cluster

cluster：同一个程序在不同的集群（如不同的数据中心），使用不同的配置

注意：1、如果不同的集群使用相同的配置，则没必要创建集群

2、apollo客户端默认读取机器上opt/settings/server.properties ( linux )或 c:\opt\settings\server.properties ( windows ) 文件中的idc属性作为集群 名字。如SHAJQ（金桥数据中心）、SHAOY（欧阳数据中心)

3、在apollo管理界面创建的集群名字，应和客户端的server.properties中的 idc属性对应

namespace：有公共namespace和项目私有namespace之分。namespace有点对应程序中间 不同配置文件的概念。

私有namespace：1、私有的namespace只能被所属的应用获取到

2、创建私有的namespace实现分组管理配置。

3、私有namespace格式可以是xml、yml、yaml、json、txt。opollo-client

可以通过ConfigFile接口来获取非properties格式的namespace的内容。

4、1.3.0及以上版本的opollo-client针对yaml/yml提供了更好的支持。可 以通过ConfigService.getConfig("someNamespace.yml")直接获取Config 对象，也可以通过@EnableApolloConfig("someNamespcace.yml")或 apollo.bootstrap.namespaces=someNamespaces.yml注入yml配置到 spring/spring-boot中。

公有namespace：1、公共的Namespace配置能被任何项目读取到。

2、通过公共namespace可以实现公共组件的配置，或多个应用共享一份 配置的需求。

3、应用可以关联其他的公共namespace，如果想覆盖关联过来的 namespace配置，修改相应的配置。

namespace名称，appId，集群三个确定一个namespace，对应一个namespaceid

配置项item关联到namespaceid上。

AppNamespace表中存放的是应用拥有哪些namespace，是私有还是公有

Namespace表中存放的是 appid关联了哪些namespace

item表（配置参数表）关联Namespace表中的id

一个portal可以管理多套环境

1、增加一套环境就需要多部署一套config service，admin service和相应的configdb

2、portaldb库ServerConfig表apollo.portal.envs增加环境。

在apollo-portal/config/apollo-env.properties文件中定义环境

local.meta=http://localhost:8080

dev.meta=http://localhost:8080

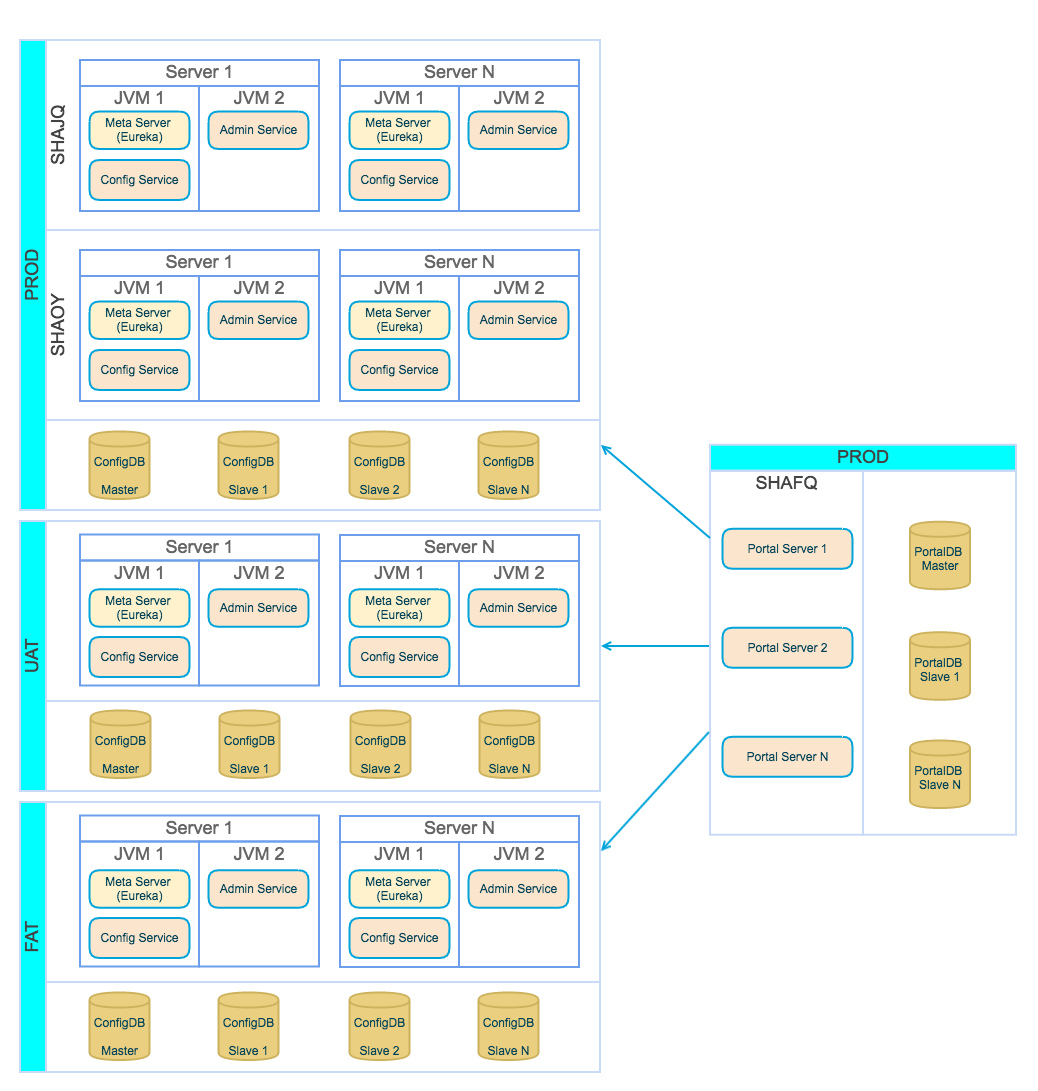
fat.meta=

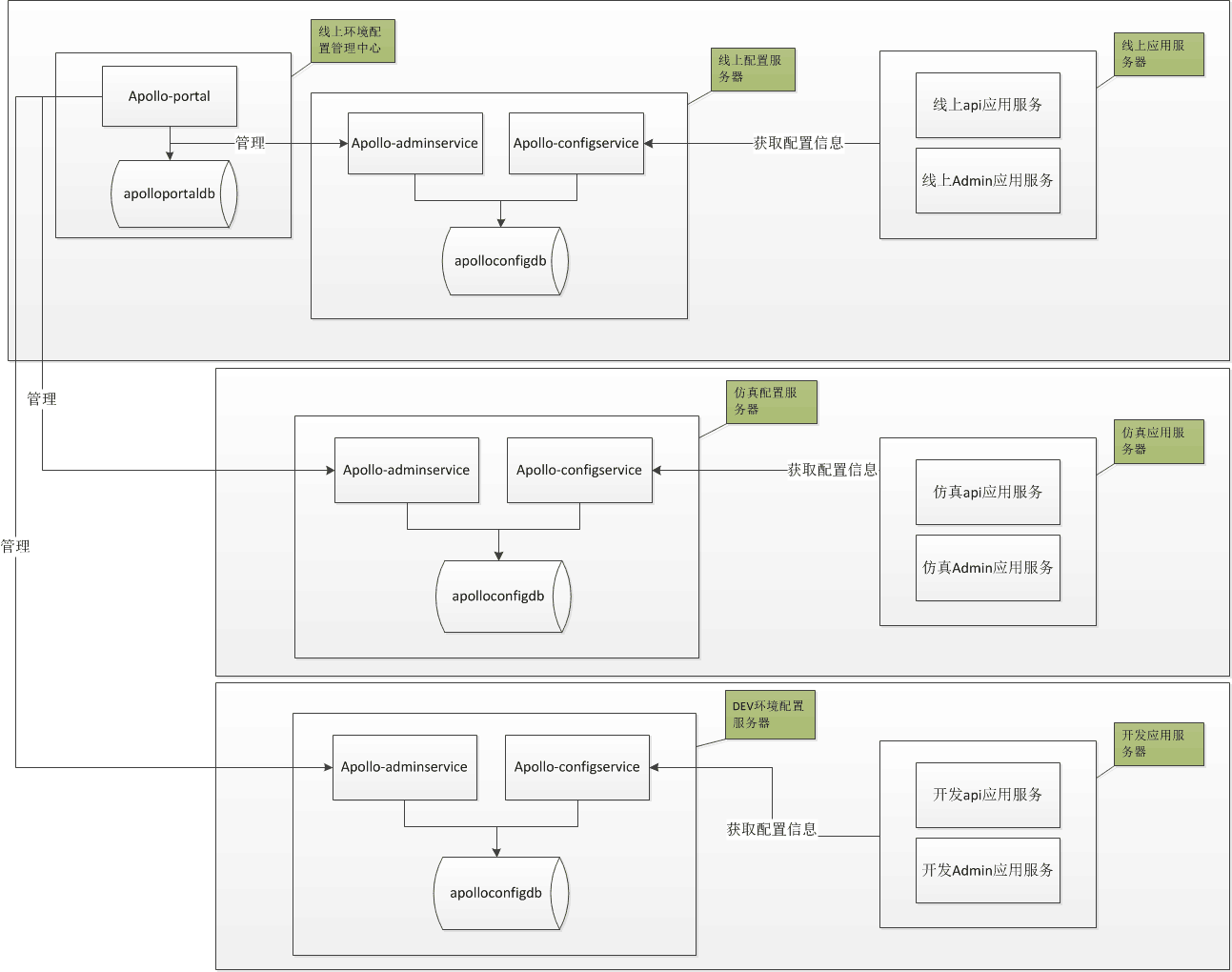
uat.meta=

lpt.meta=${lpt\_meta}

pro.meta=http://localhost:8081

客户端定义环境是在server.properties，或者其他方式的传参数。





portal 是管理端

config，admin两个是后台服务

客户端到meta-server上获取服务信息，直接访问config和admin服务。