



PolarisMesh开发者例会

2021-11-16

- 北极星项目介绍
- 社区的晋升机制介绍
- Roadmap介绍
- 如何进行北极星开发
- Q&A

北极星项目介绍

截止2021年9月，在线节点超过

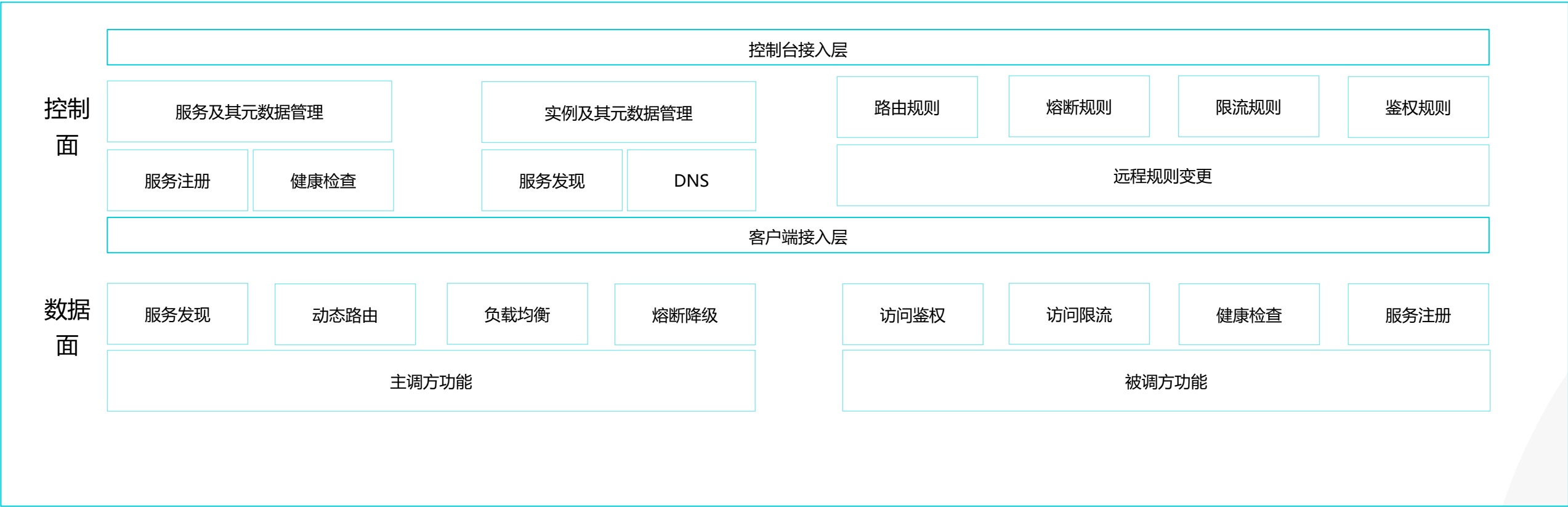
5,000,000⁺

- 每日服务调用量超过30万亿
- 接口的调用成功率超过99.999%
- 覆盖腾讯内部90%以上的业务部门



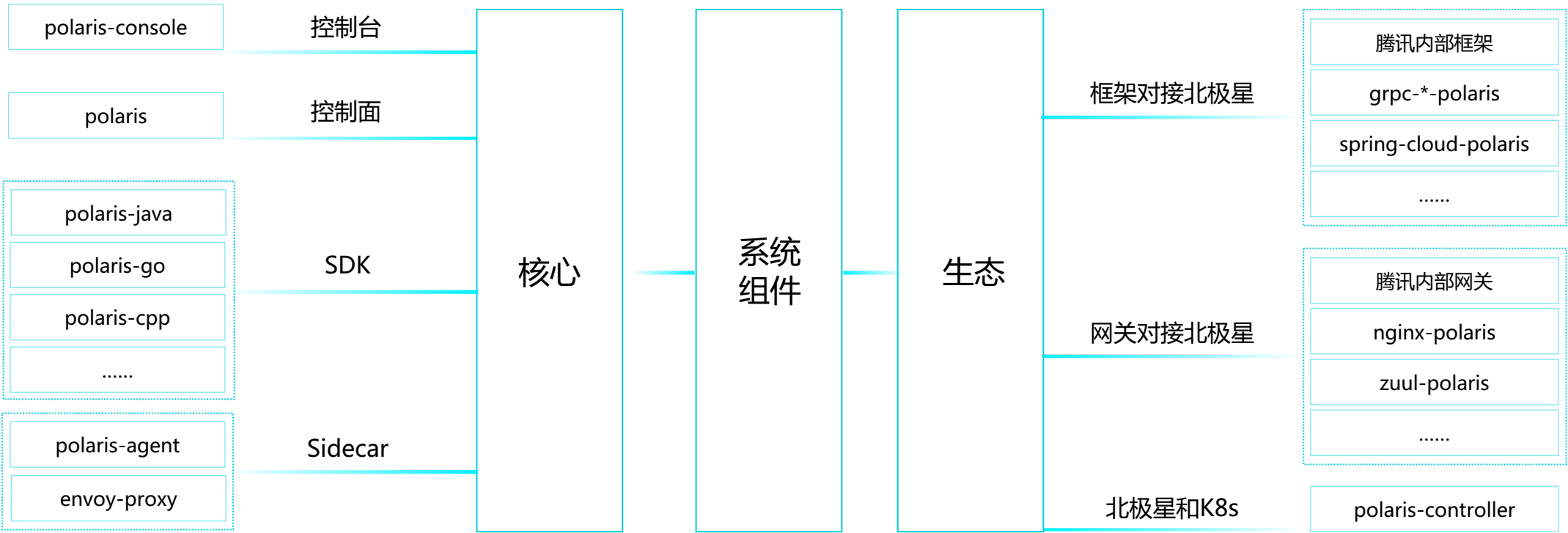
北极星以服务注册中心为基础，扩展了服务治理功能以及相应的控制面

- 基础：服务发现、服务注册、健康检查、DNS
- 扩展：流量调度（动态路由、负载均衡）、熔断降级（实例、接口、服务三级熔断）、访问控制（限流、鉴权）



北极星的系统组件分为核心组件和生态组件两大类

- 核心：控制台、控制面、数据面（SDK和Sidecar两种形态）
- 生态：用于对接框架、网关和Kubernetes，帮助北极星融入周边生态



核心亮点：支持大量微服务开发技术栈

功能特性		多语言SDK	Sidecar	gRPC/Spring Cloud等框架集成		HTTP	DNS
基础功能	服务发现	支持	支持	支持		支持	支持
	服务注册	支持	支持	支持		支持	-
	健康检查	支持	支持	支持		支持	-
流量调度	动态路由	支持	支持	支持		-	-
	负载均衡	支持	支持	支持		-	-
熔断降级	熔断	支持	支持	支持		-	-
	降级	支持	支持	支持		-	-
访问控制	访问限流	支持	支持	支持		-	-
	访问鉴权	支持	支持	支持		-	-
可观测性			路由/熔断/降级/限流/鉴权监控			-	-

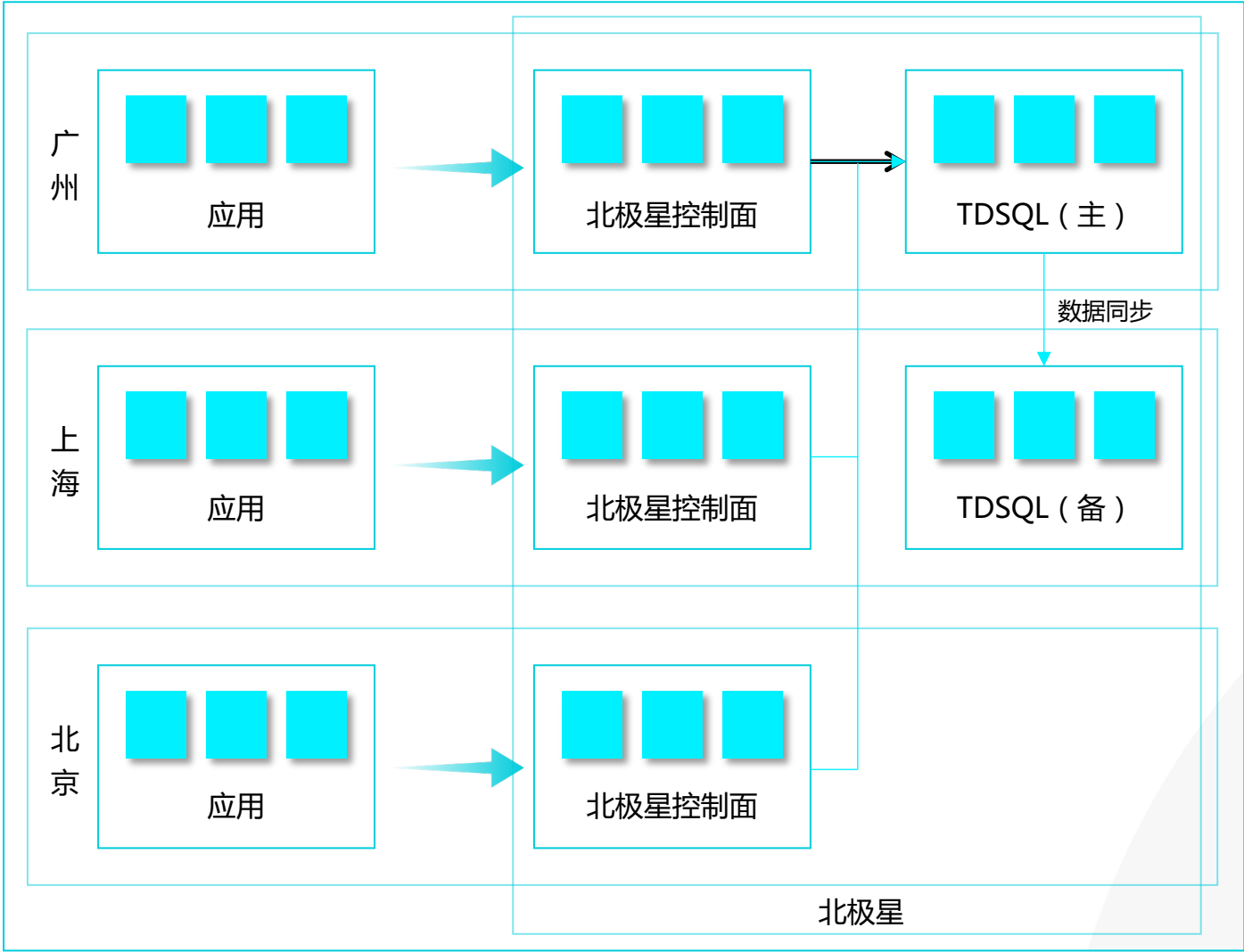
核心亮点：高扩展性、高可用性

高扩展性

- 北极星服务端支持计算存储分离、控制面无状态，可以随着接入节点的增加平行扩展，轻松支持百万节点
- 北极星服务端支持数据分片，并发性能和存储容量也可以平行扩展

可高可用性

- 从数据面到控制面，均有内存和磁盘数据缓存。即使存储层故障，也不影响存量数据的现网服务
- 控制面支持跨城多中心部署，TDSQL主备集群分别同城三中心部署



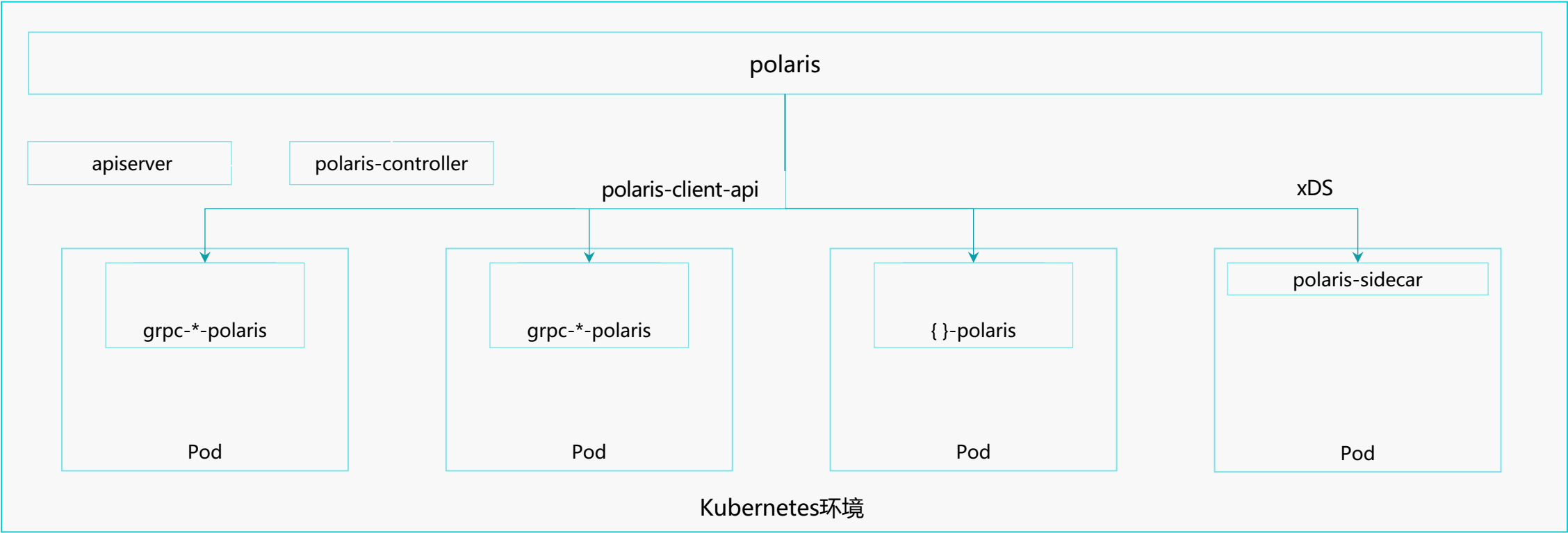
核心亮点：统一VM和K8s、框架和网络的微服务体系

云原生

- 支持K8s服务和网格Sidecar的自动注入
- 支持多K8s集群的全局服务发现和治理

联通性

- 打通K8s和框架的服务发现和治理体系
- 打通K8s和虚拟机的服务发现和治理体系



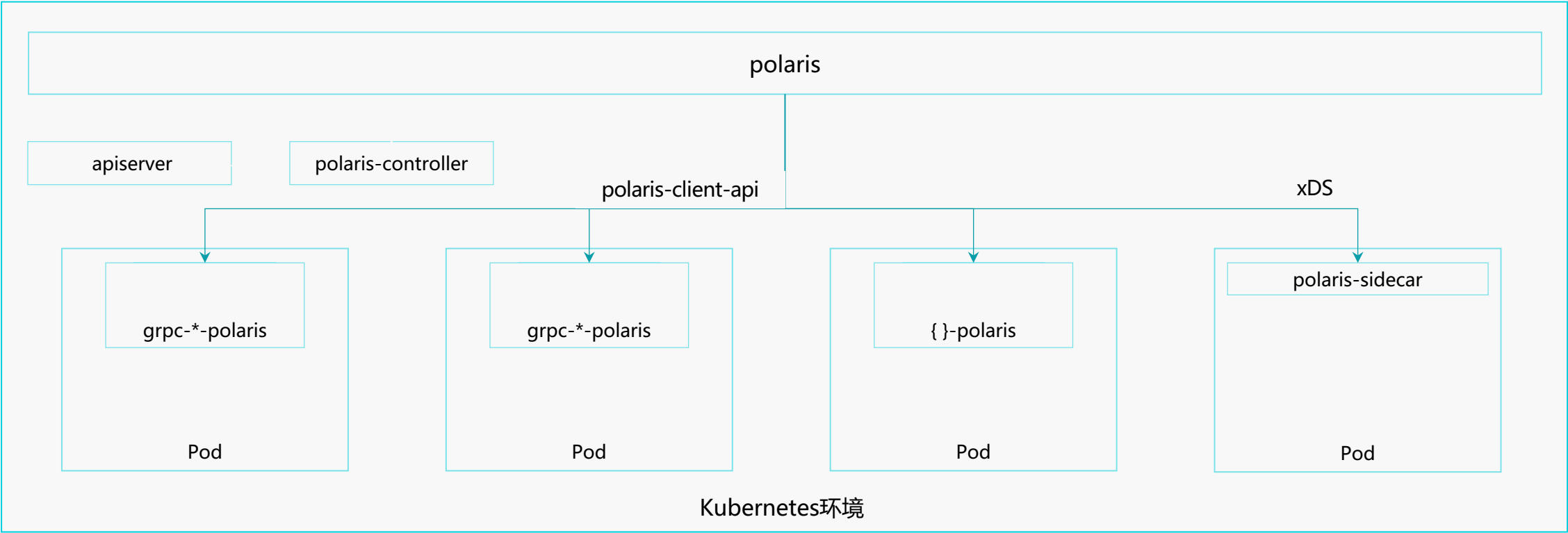
核心亮点：统一VM和K8s、框架和网络的微服务体系

云原生

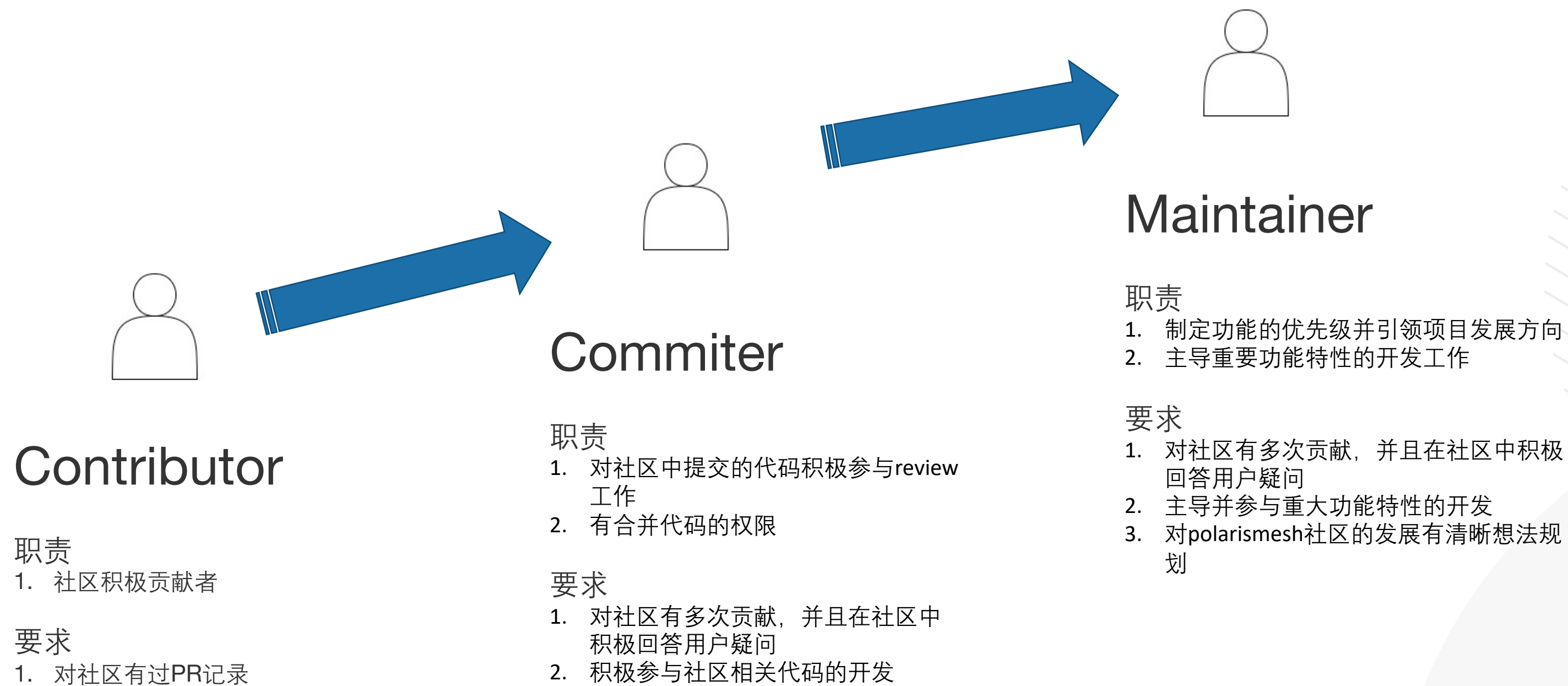
- 支持K8s服务和网格Sidecar的自动注入
- 支持多K8s集群的全局服务发现和治理

联通性

- 打通K8s和框架的服务发现和治理体系
- 打通K8s和虚拟机的服务发现和治理体系



社区的晋升机制介绍



如何成为Contributor

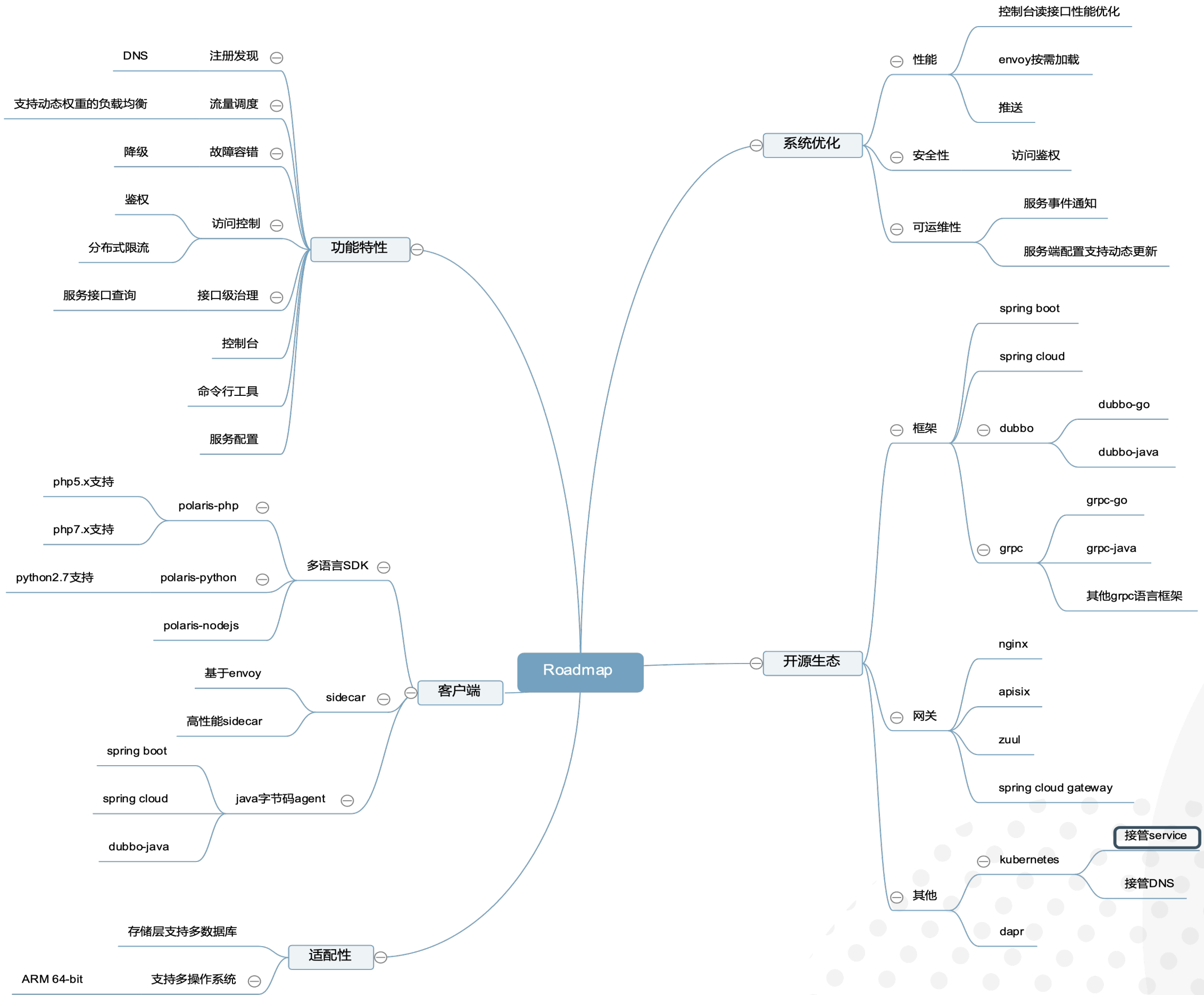
- 1.参与过polarismesh下的任意项目，只要提交过PR并得到merge即可成为contributor

如何成为Committer

- 1.首先要成为contributor角色
- 2.提交 10 个 Pull Request 并得到merge，即可获取被提名资格（代码类PR）
- 3.由maintainer发起committer提名申请，并在社区得到其余至少1个Maintainer同意以及2 ~ 3个committer同意
 1. 推举社区contributor成为committer需要的信息
 - 1.contributor对自己社区工作的简单总结
 - 2.历来参与的PR以及ISSUE列表

1. 首先要成为committer角色
2. 至少超过一年时间在社区有非常高的活跃度
 - a. 在某一子项目有突出的Owner意识以及较好的推动该子项目发展的意识和规划
 - b. 积极参与社区同学问题解答
 - c. 在社区有较高的影响力
3. 由maintainer发起并在社区得到其余至少2个Maintainer同意以及2 ~ 3个committer同意
 - a. 推举社区committer成为mainatiner需要的信息
 - I. commiter对自己社区工作的简单总结
 - II. 自己在PolarisMesh某些子项目的主要贡献
 - III. 历来参与的PR以及ISSUE列表

Roadmap介绍



每月初确定版本功能范围， 每月底发布迭代版本

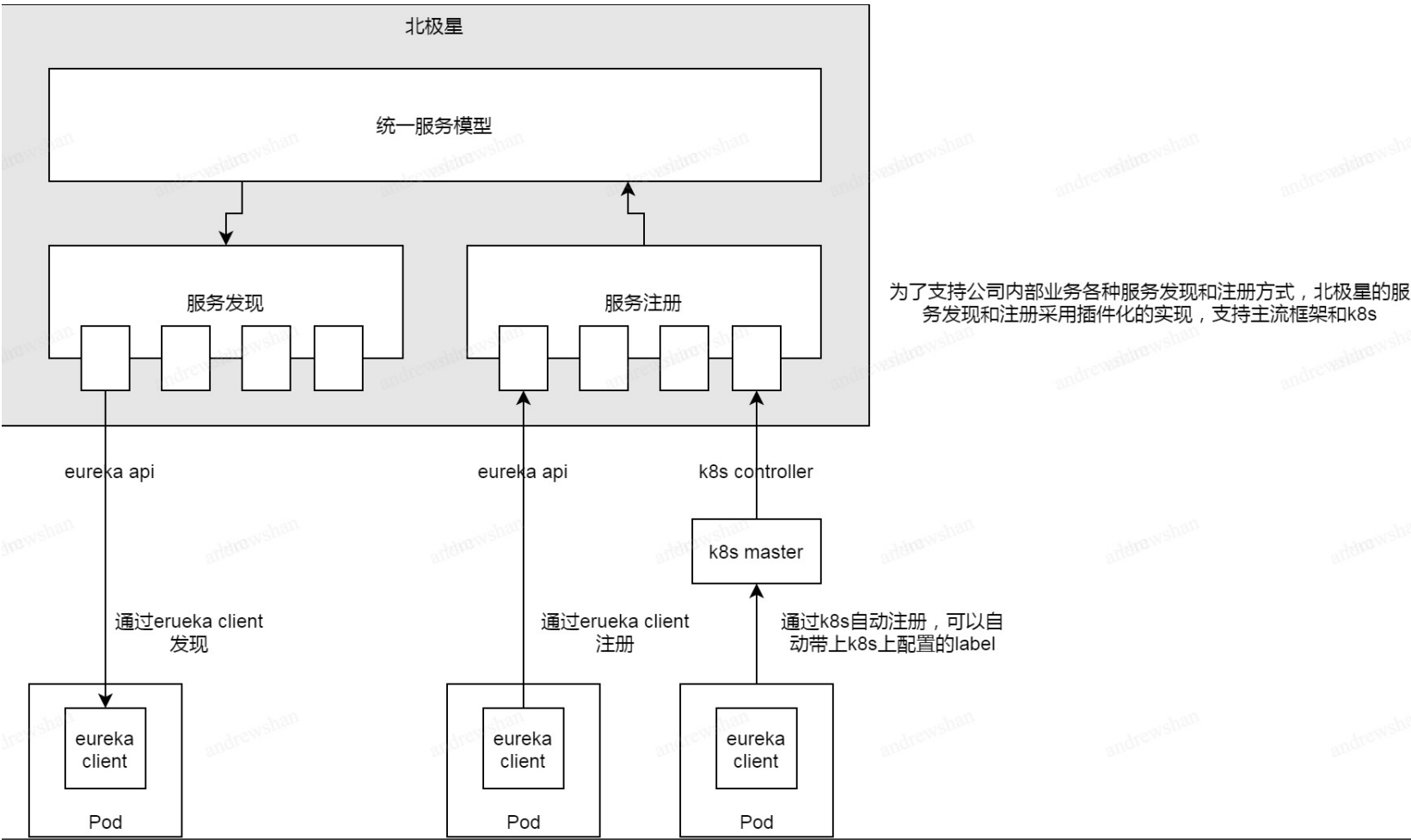
时间	版本号	规划特性
2021-11	v1.4.0	系统优化： 控制台读接口优化 服务事件通知 客户端： 支持python 5.x版本接入 控制台： 支持命名空间管理
2021-12	v1.5.0	规划中，欢迎大家认领

如何进行北极星开发

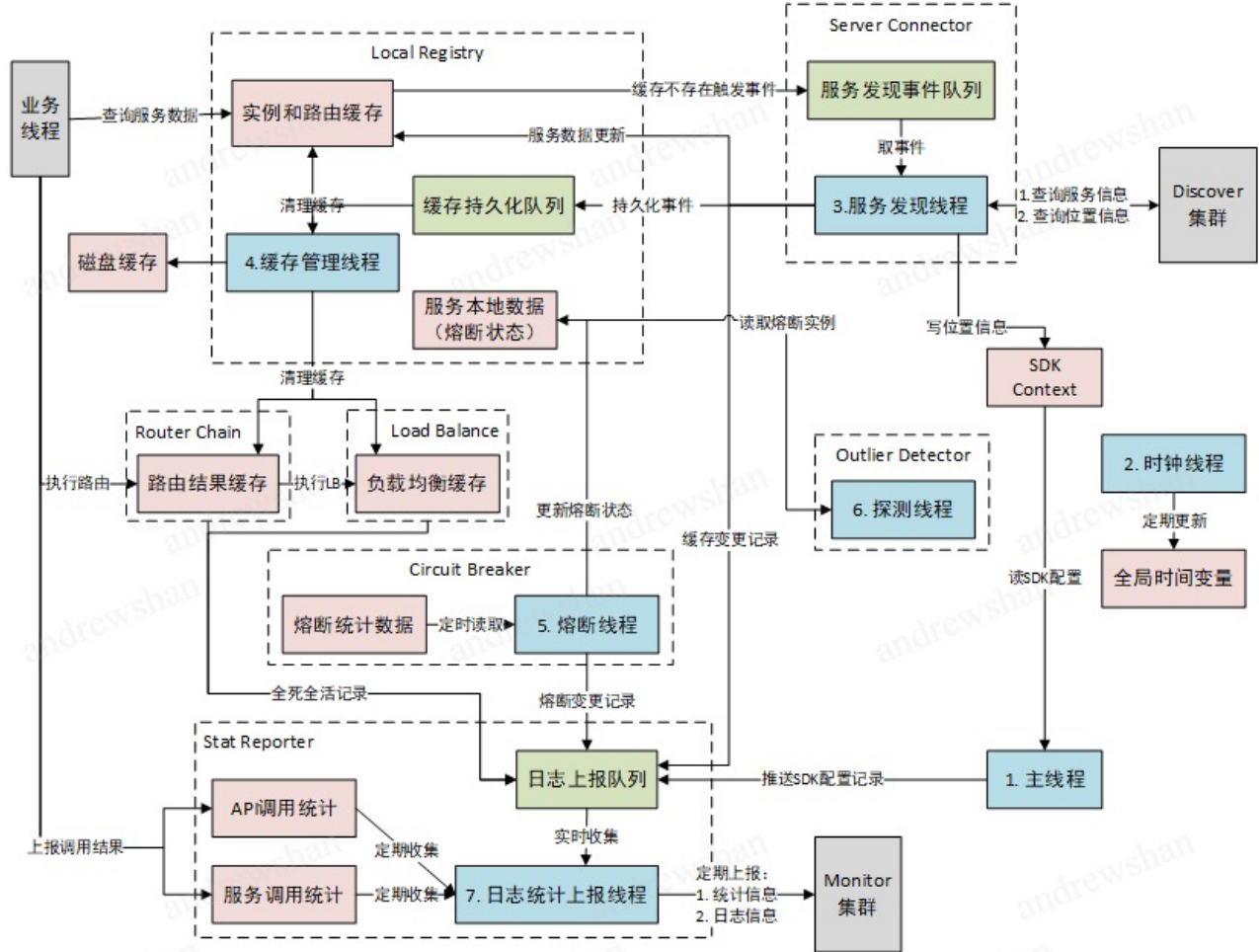
Github源码地址：<https://github.com/PolarisMesh>

组件	功能	代码库路径
polaris	北极星控制面server	https://github.com/polarismesh/polaris
polaris-console	北极星web控制台	https://github.com/polarismesh/polaris-console
website	北极星开源官网 polarismesh.cn源码	https://github.com/polarismesh/website
polaris-sidecar	北极星边车组件	https://github.com/polarismesh/polaris-sidecar
polaris-go	北极星GO语言SDK	https://github.com/polarismesh/polaris-go
polaris-java	北极星JAVA语言SDK	https://github.com/polarismesh/polaris-java
polaris-cpp	北极星C++语言SDK	https://github.com/polarismesh/polaris-cpp
polaris-controller	北极星对接k8s的插件	https://github.com/polarismesh/polaris-controller
nginx-polaris	北极星对接nginx的插件	https://github.com/polarismesh/nginx-polaris

服务端插件化架构



客户端插件化架构



```
polaris-plugins
> polaris-plugin-api
> polaris-plugins-cache [flow-cache]
> polaris-plugins-circuitbreaker
> polaris-plugins-connector [connector]
> polaris-plugins-healthchecker
> polaris-plugins-loadbalancer [loadbalancer]
> polaris-plugins-observability
> polaris-plugins-ratelimiter
> polaris-plugins-registry
> polaris-plugins-router
> .flattened-pom.xml
> polaris-plugins.iml
> pom.xml
> polaris-ratelimit
```

为什么要做

- 1.php是世界最好的语言
- 2.让php生态也能体验Polaris强大的服务发现以及治理能力
- 3.内外部对polaris-php sdk的需求

期间过程以及注意点

1. php-zend 框架在不同 php 版本下，一些API的定义是不同的，要支持多php版本的话，这里需要用上条件编译，根据php-zend的版本来选择编译的代码段
2. 尽可能让整个封装层代码简单，不要参杂太多的逻辑在里面
3. 尽可能保持跟Polaris-CPP一样的接口使用体验
4. php有自己的一套内存申请机制，在写插件的时候一定要注意这一块的使用
5. php有自己的一套数据类型，在使用zend将php的数据类型转换到c++时，要注意指针问题，比如zval *以及zval **

Q&A