

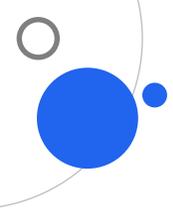
CNCF网络研讨会

- cloud-provider-subproject-baiducloud

百度智能云 周倜、叶子超、孙天元、王城程、杨帆



- 早期Kubernetes中关于云厂商的代码是深度耦合的在系统中的，当云厂商代码逻辑需要更新时，不得不等待Kubernetes自身的迭代周期，而Kubernetes代码也变得非常臃肿。
- 在这个背景下Tim Hockins提出了external cloud provider的解决方案，力图让所有云厂商相关逻辑通过controller的方式与原生的Kubernetes进行拆解，这就启动了SIG Cloud Provider。
- 在SIG Cloud Provider的带动下，Kubernetes的release 与云厂商的代码逻辑进行了解耦，Kubernetes的binary变得更小，对于Cloud Provider的迭代速度也有了明显的提升。



SIG Cloud Provider做什么

- 目前SIG Cloud Provider广义的关注领域包括：
 - > 云厂商IaaS层资源基于Kubernetes的生命周期管理
 - > 云厂商与Kubernetes的交互接口扩展
 - > 云厂商中立的测试框架，文档以及工具
 - > 云厂商与Kuberentes的交互API标准制定

BaiduCloud为什么加入SIG Cloud Provider

- SIG Cloud Provider始终坚持推动Kubernetes云厂商接口的标准化工作，通过技术来优化云厂商和Kubernetes的连接。
- 而百度云云原生团队2019年在Kubernetes上游主库贡献量排名全球前十，中国前三，以实际行动推动Kubernetes云厂商接口的落地与实践。

k8s.io/cloud-provider-alibaba



k8s.io/cloud-provider-aws



k8s.io/cloud-provider-azure



k8s.io/cloud-provider-baidu



k8s.io/cloud-provider-cloudstack



k8s.io/cloud-provider-gcp



k8s.io/cloud-provider-digitalocean



k8s.io/cloud-provider-huawei



k8s.io/cloud-provider-ibm



k8s.io/cloud-provider-openstack



k8s.io/cloud-provider-tencent



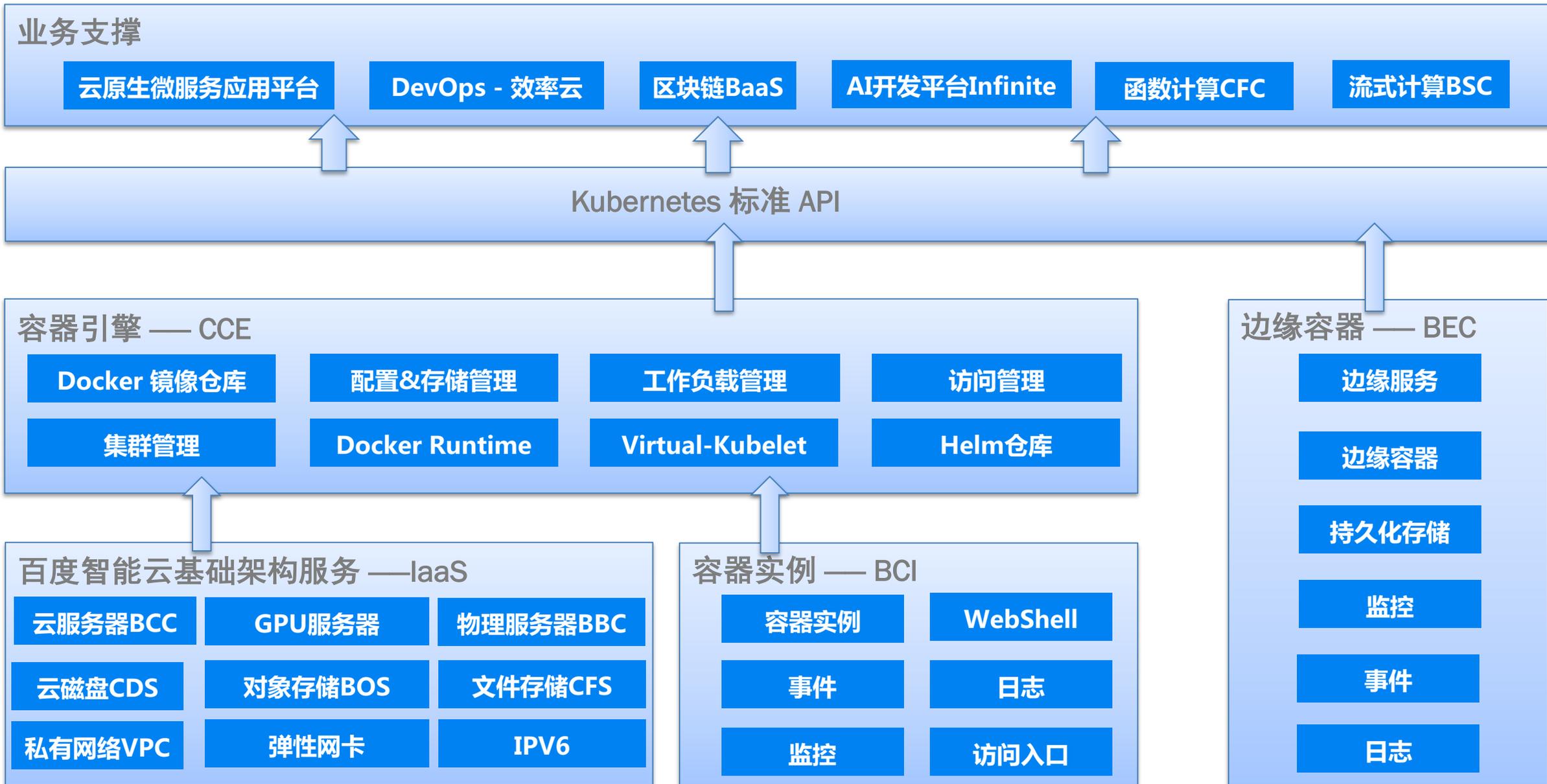
k8s.io/cloud-provider-vmware



Show 50 entries

#	Company	Commits
1	Google	2318
	*independent	2062
2	Red Hat	598
3	VMware	455
4	BoCloud	326
5	Microsoft	217
6	Huawei	209
7	NEC	191
8	Intel	188
9	IBM	153
10	Baidu	99
11	ZTE Corporation	87
12	Futurewei	86
13	DaoCloud	80
14	SUSE	76
15	Nvidia	61
16	Tencent	60
17	Cloudbase Solutions	57
18	Amazon	52
19	Fujitsu	52

(数据来源: 截止2019年12月31日)



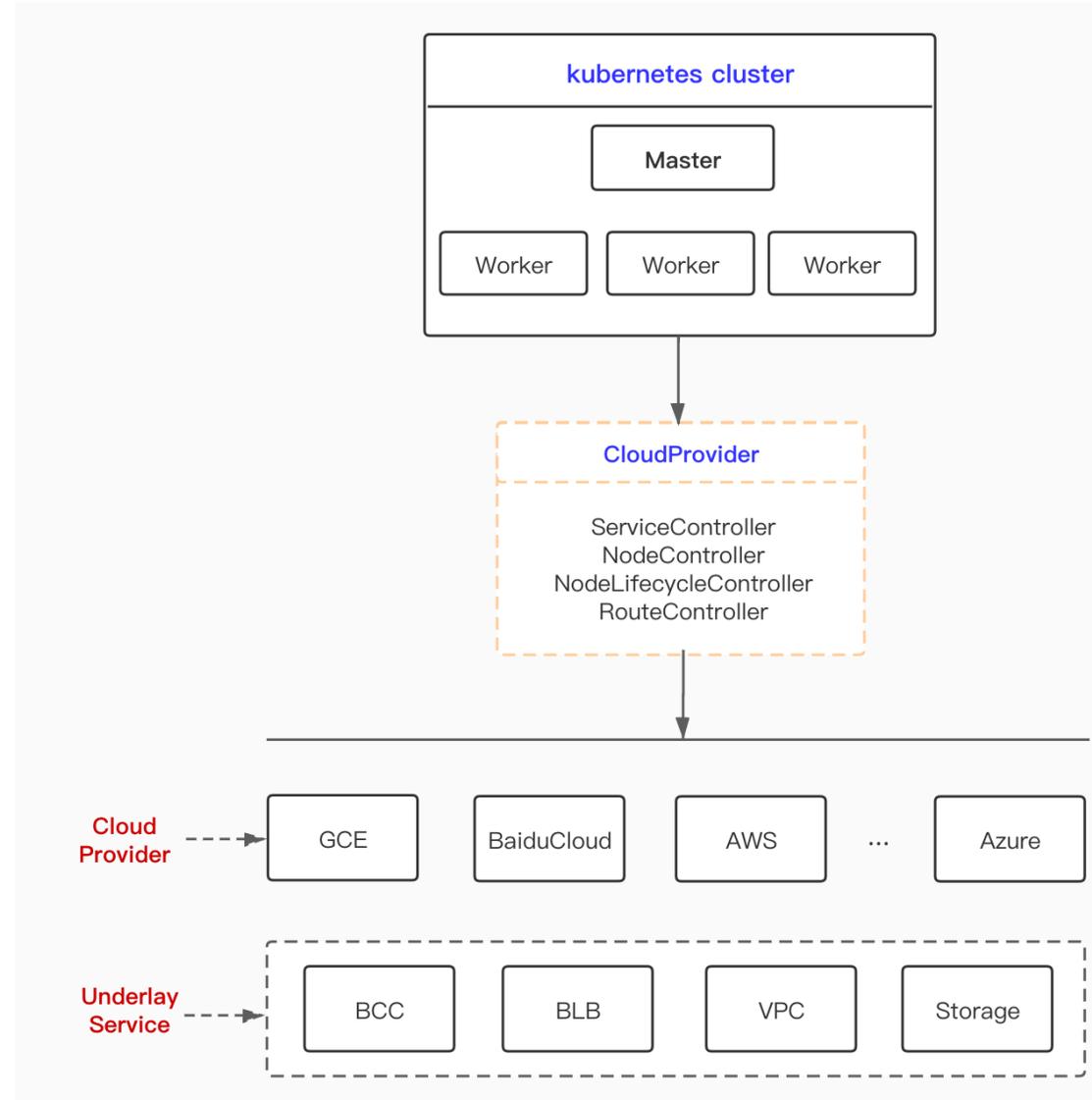
目前BaiduCloud参与建设的开源组件

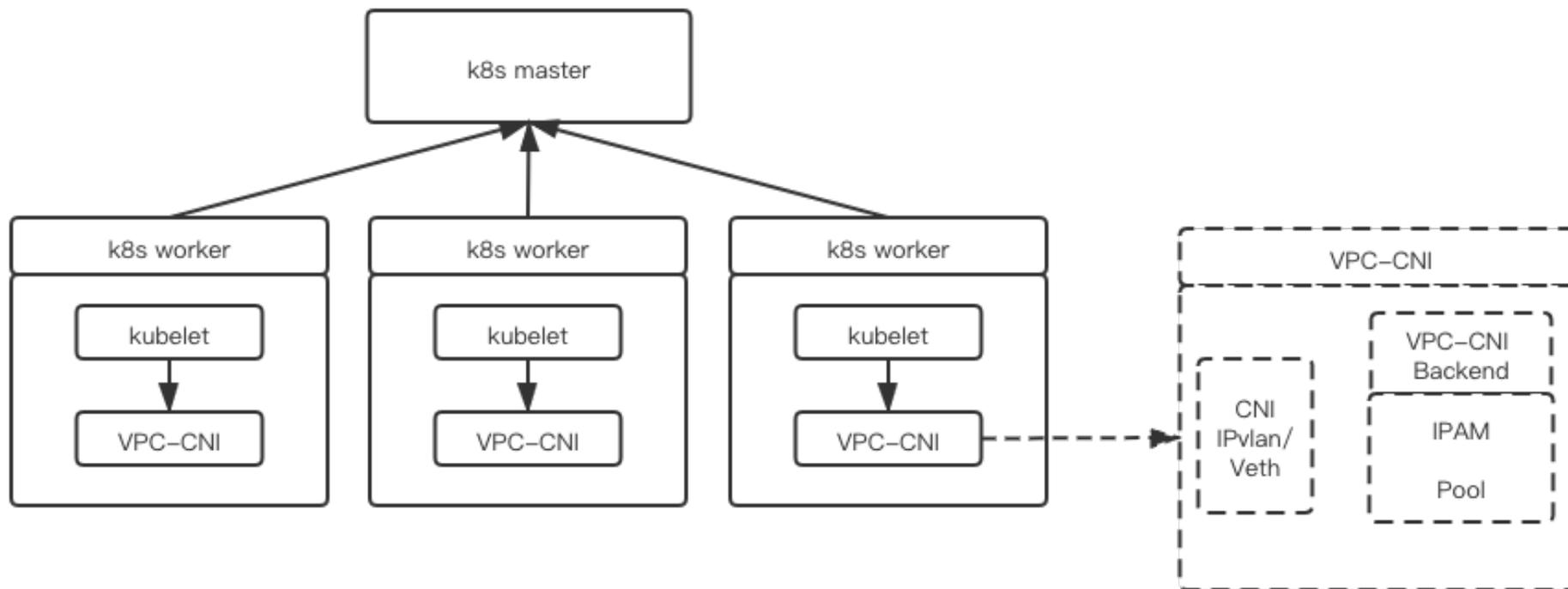
- Cloud Controller Manager : cloud-provider-baiducloud
- <https://github.com/kubernetes-sigs/cloud-provider-baiducloud>
- 集群生命周期管理 : cluster-api-baiducloud
- <https://github.com/kubernetes-sigs/cluster-api#provider-implementations>
- 集群弹性 : cluster-autoscaler-baiducloud
- <https://github.com/kubernetes/autoscaler/tree/master/cluster-autoscaler/cloudprovider/baiducloud>
- 存储 : csi-driver-bos, csi-driver-cds
- 网络 : 7层负载均衡bfe(cncf sandbox), vpc-cni
- 微服务 : spring-cloud-formula
- AI : paddle-on-k8s, edl, volcano

Cloud Controller Manager

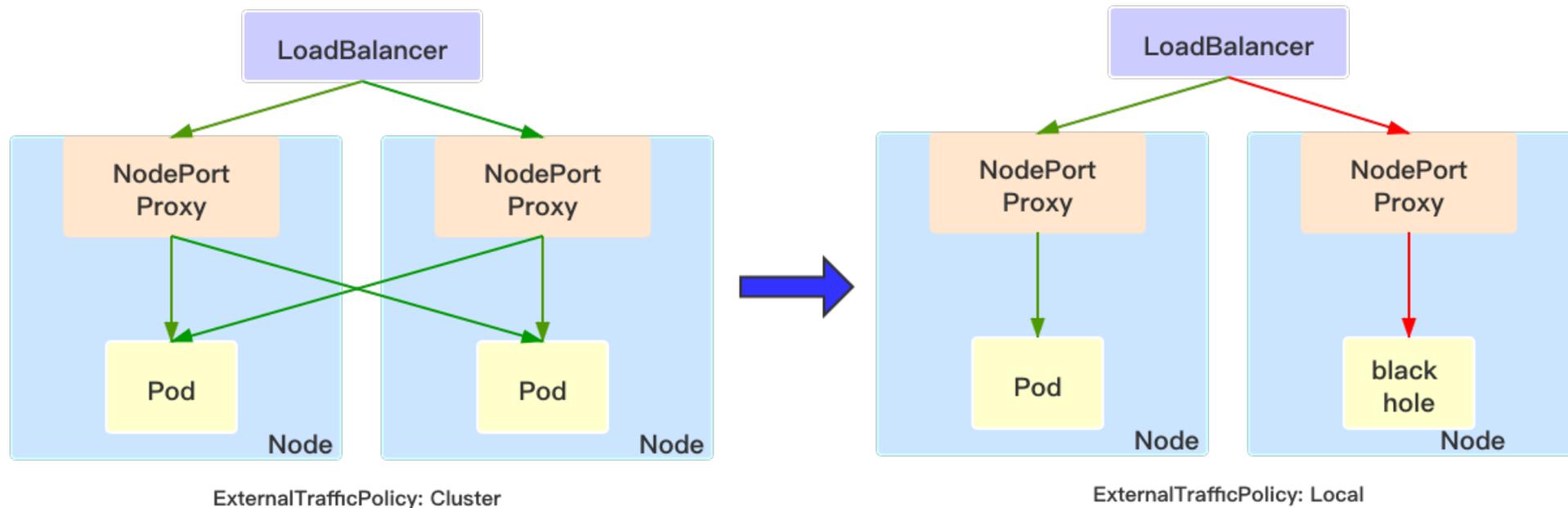
CloudController 指 K8s 的 cloud-controller-manager 组件(简称 CCM), 提供 Kubernetes 与各个云厂商基础服务的对接能力 (包含网络负载均衡, VPC 路由, DNS 等)。CCM 实现了 k8s 中的 4 个控制器 :

- NodeController/NodeLifecycleController :
- 管理计算节点的生命周期以及元信息。如 Region、Type、是否 shutdown 等
- RouteController :
- 管理集群内的路由信息。动态为节点配置实例路由, 连通 Pod 网络
- ServiceController :
- 管理负载均衡, 动态为 k8s LoadBalancer 类型配置云上的负载均衡器

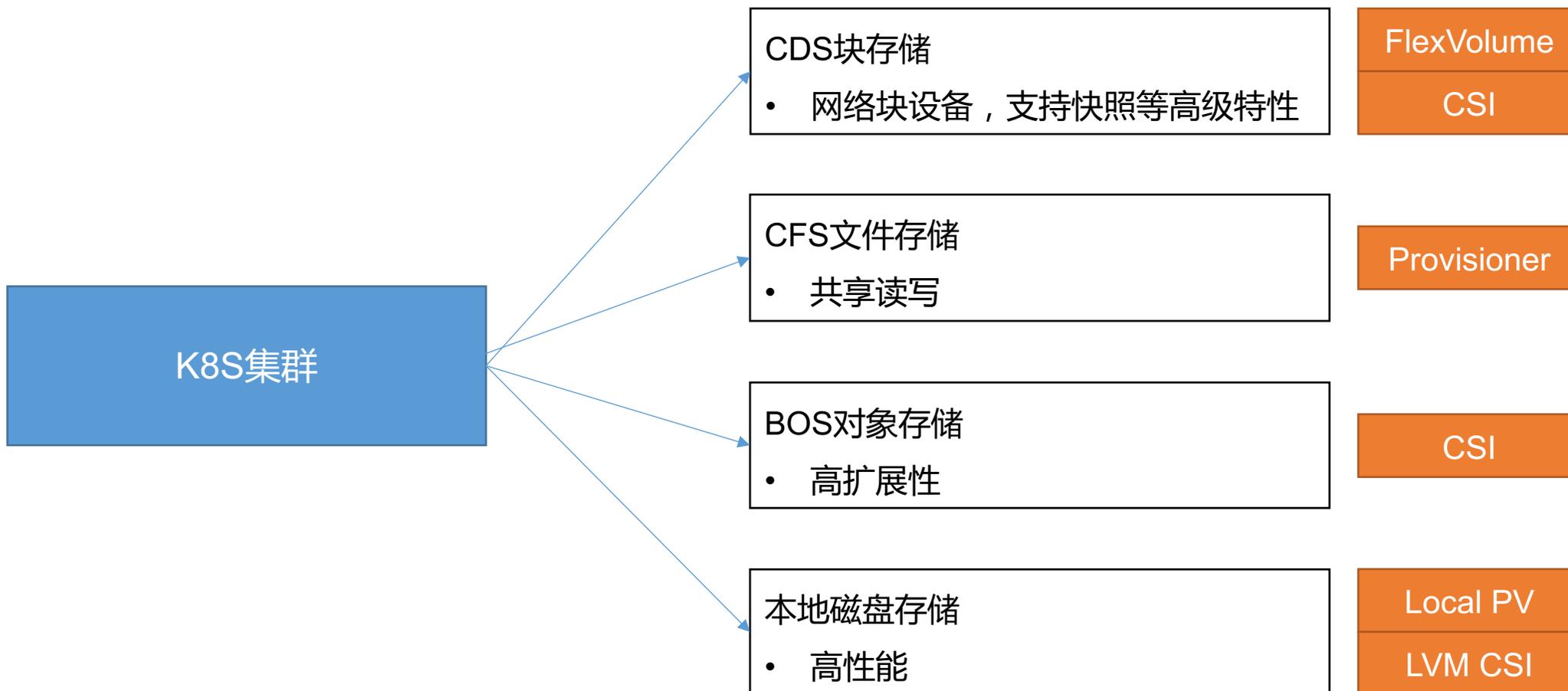




- 基于百度云弹性网卡以及辅助 IP，实现容器网络和 VPC 网络打平
- 提供 VPC 路由、弹性网卡+辅助 IP 等多种网络模式
- IPvlan 实现的高性能网络方案



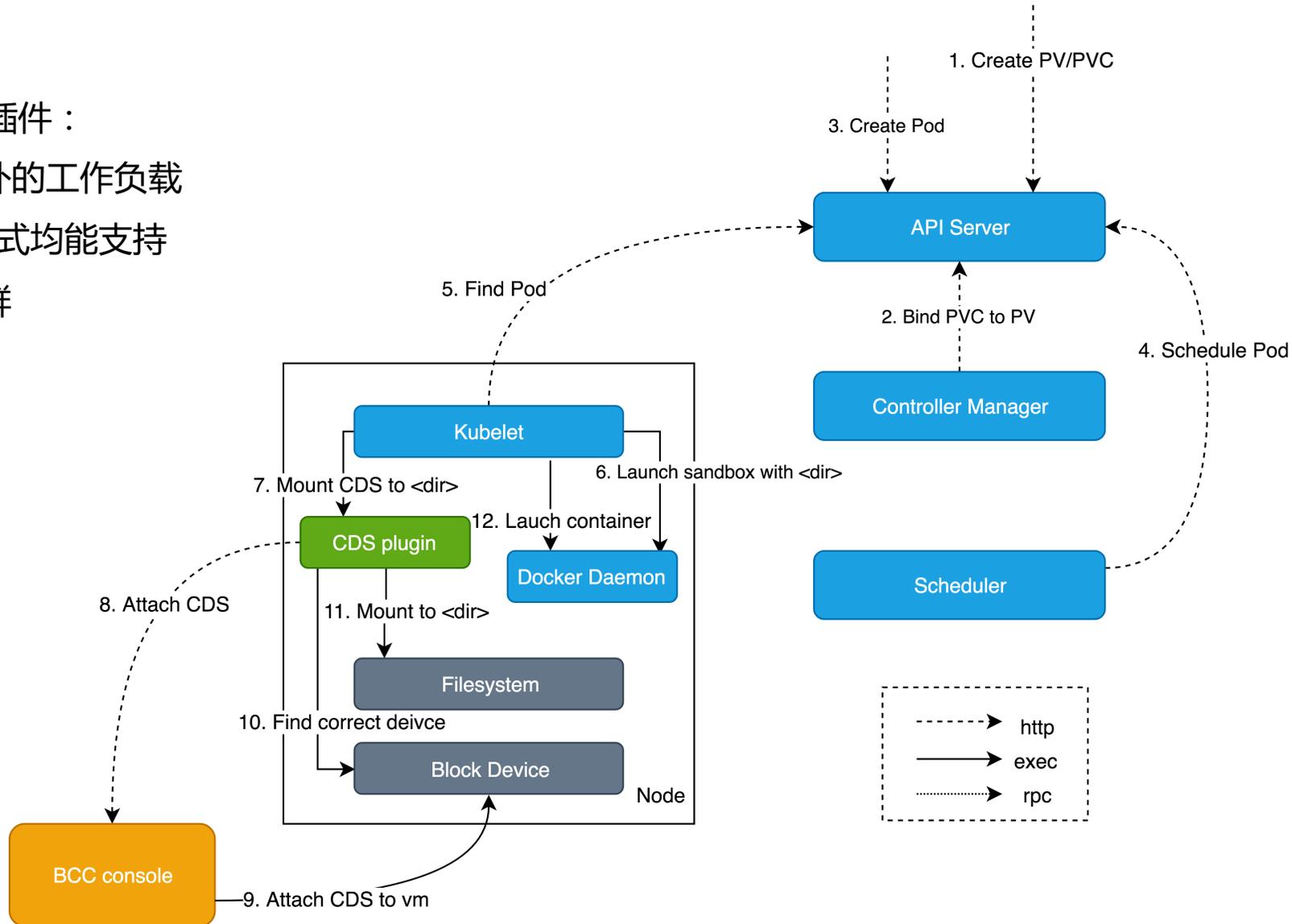
- Baidu CCM 中，通过 k8s List-Watch 机制，对 ExternalTrafficPolicy=Local 的 Service 实现负载均衡器的 Real Server 随 Pod 漂移而动态变更，保留 Real Client IP，减少网络转发。
- 通过 annotations 实现了固定 EIP、指定 BLB 等功能，方便用户使用。



CDS存储 (FlexVolume)

CDS FlexVolume插件：

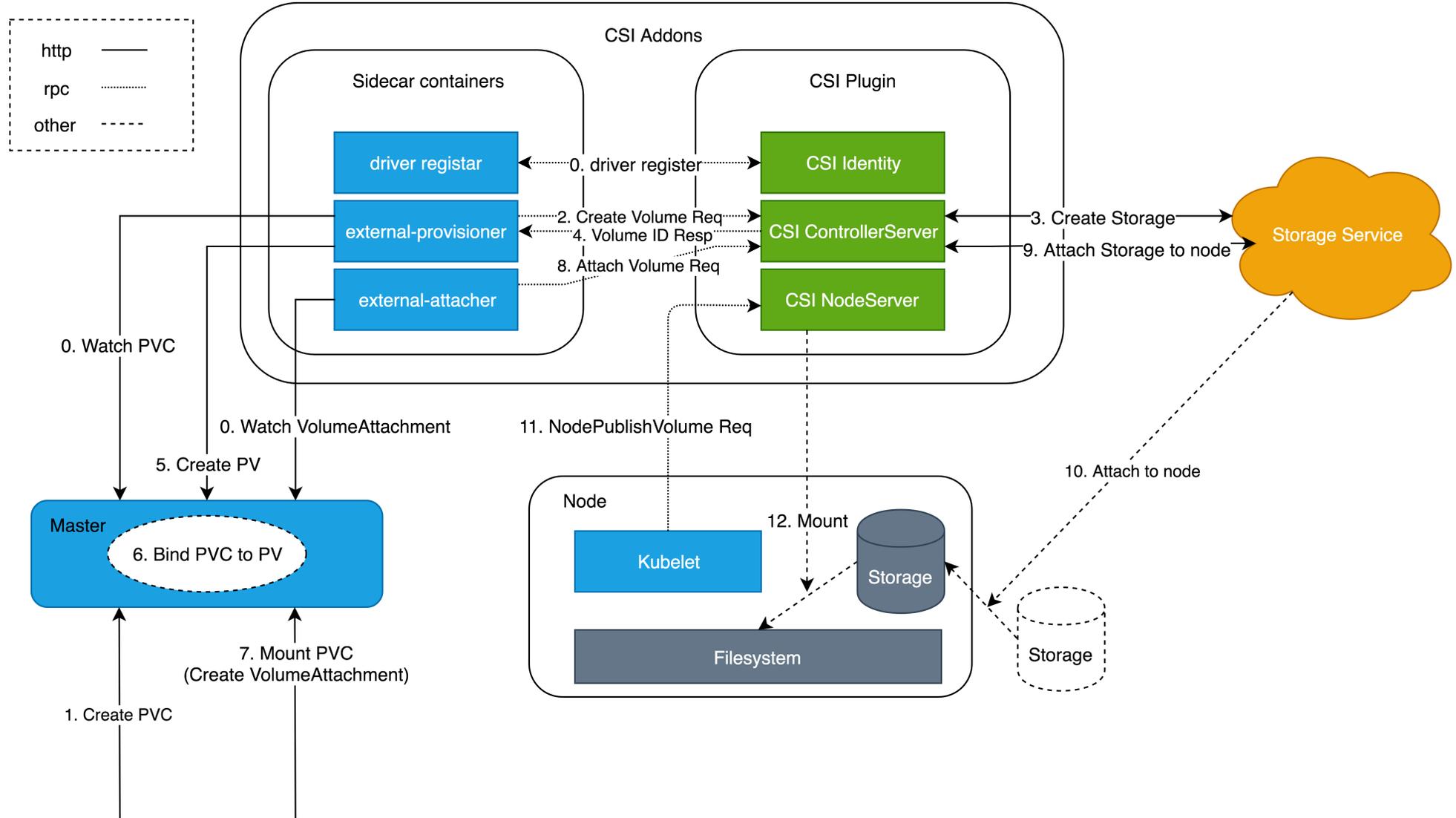
- 集群内没有额外的工作负载
- volume/PVC模式均能支持
- 支持老版本集群

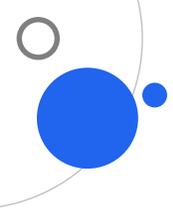


CDS存储 (CSI)

CDS CSI插件：

- 支持动态创建和销毁
- 支持多可用区拓扑
- 支持磁盘快照

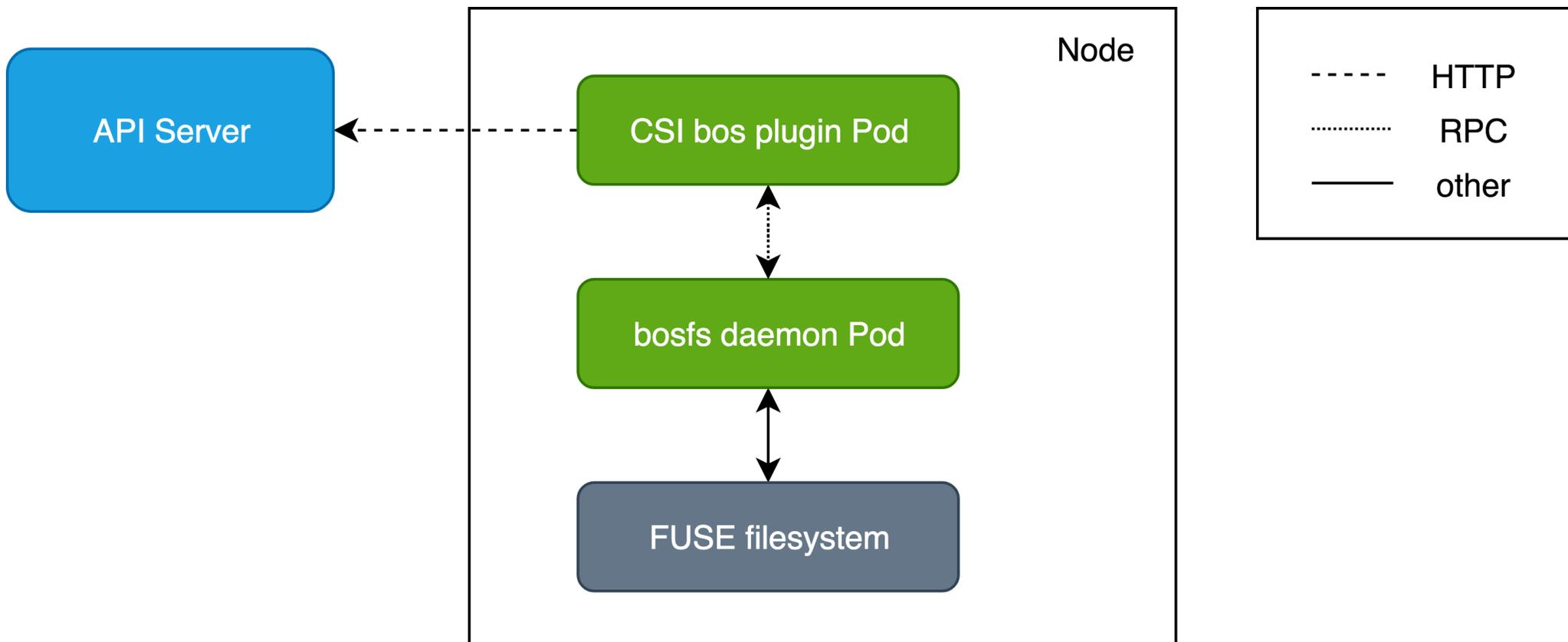




BOS存储

BOS对象存储：

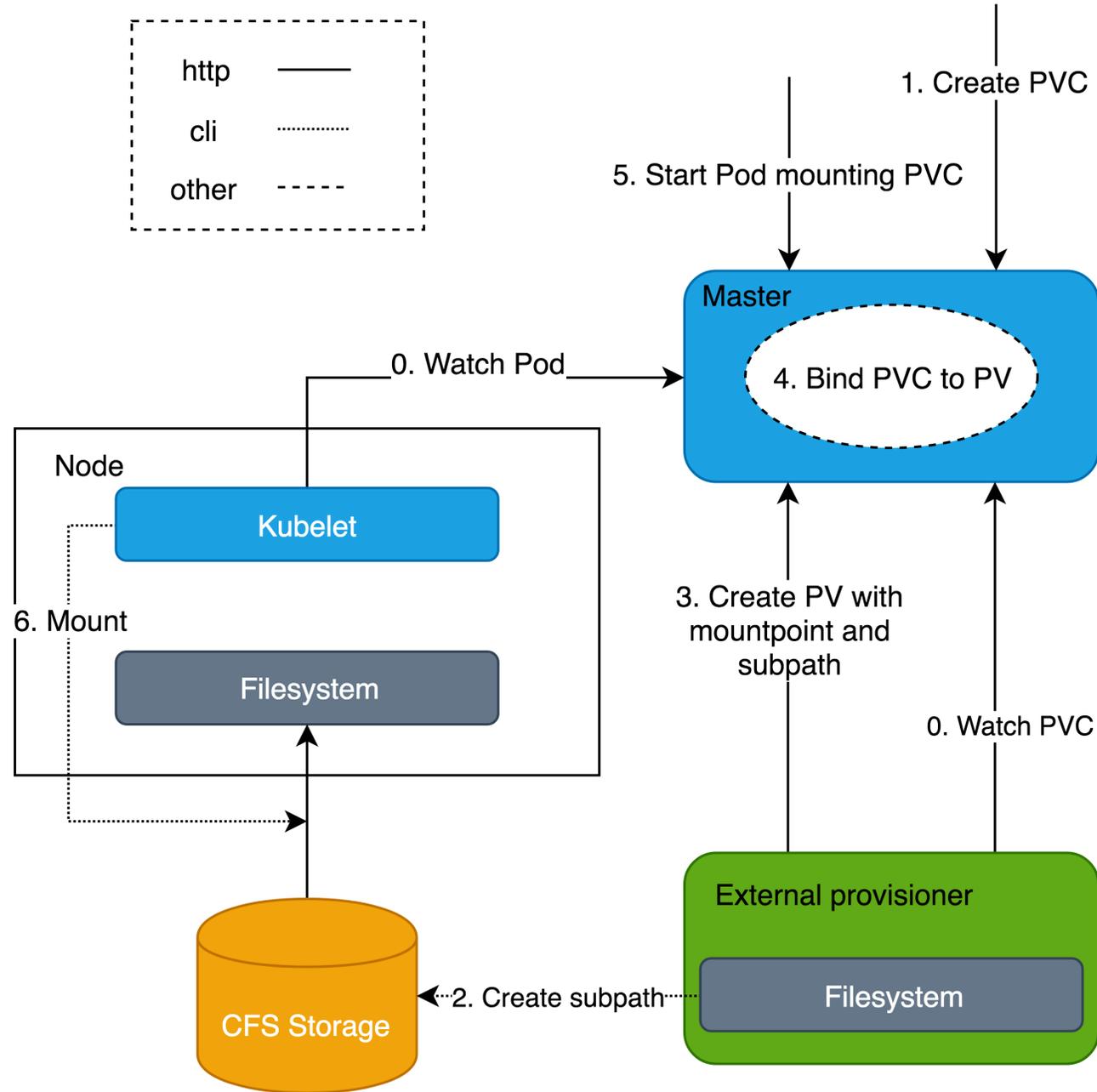
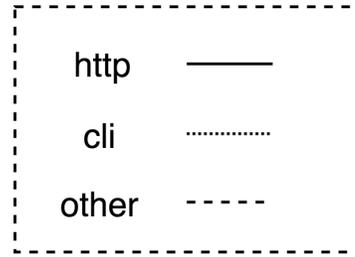
- 高扩展性，不限制用户和VPC
- 多种访问方式



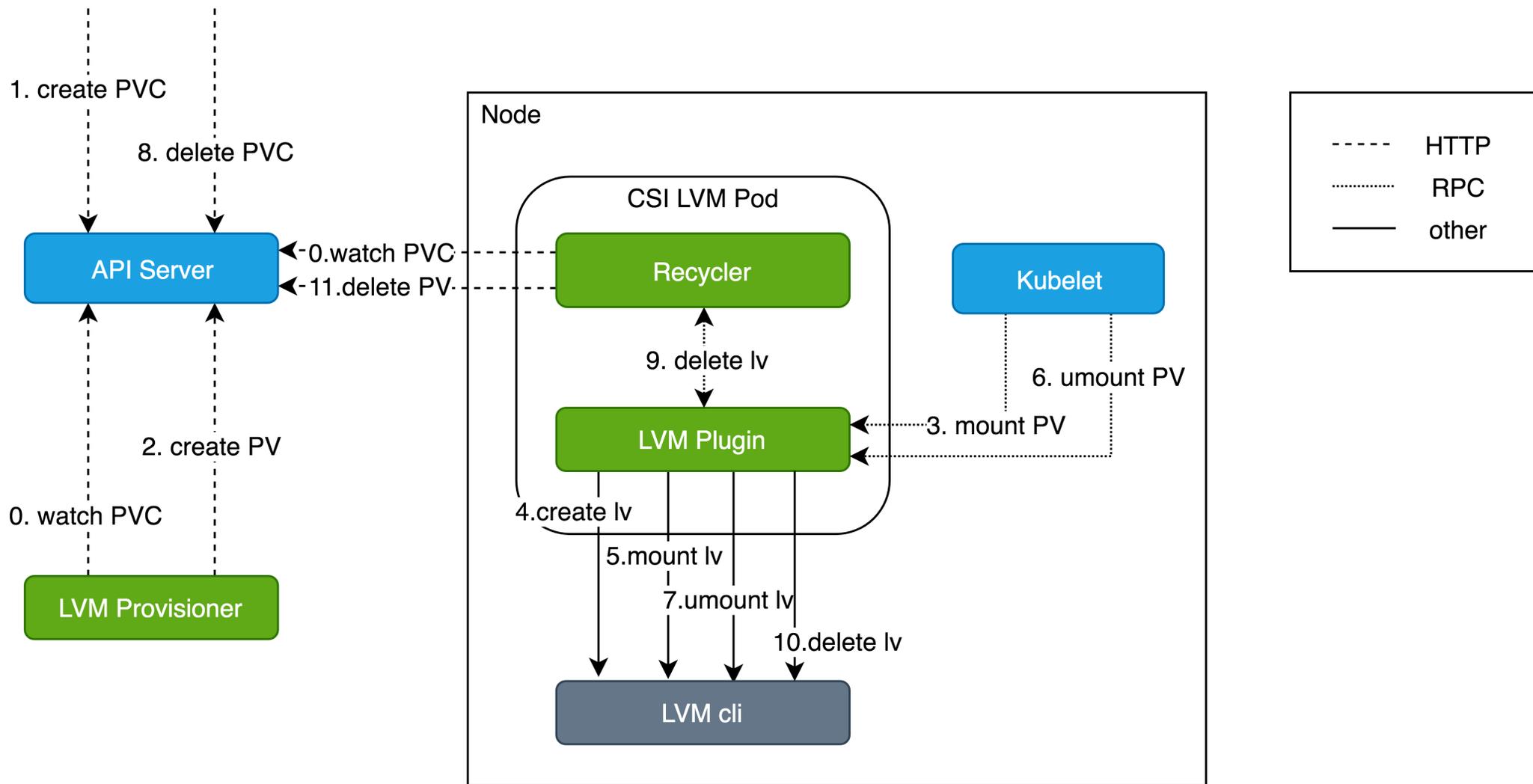
CFS存储

CFS文件存储：

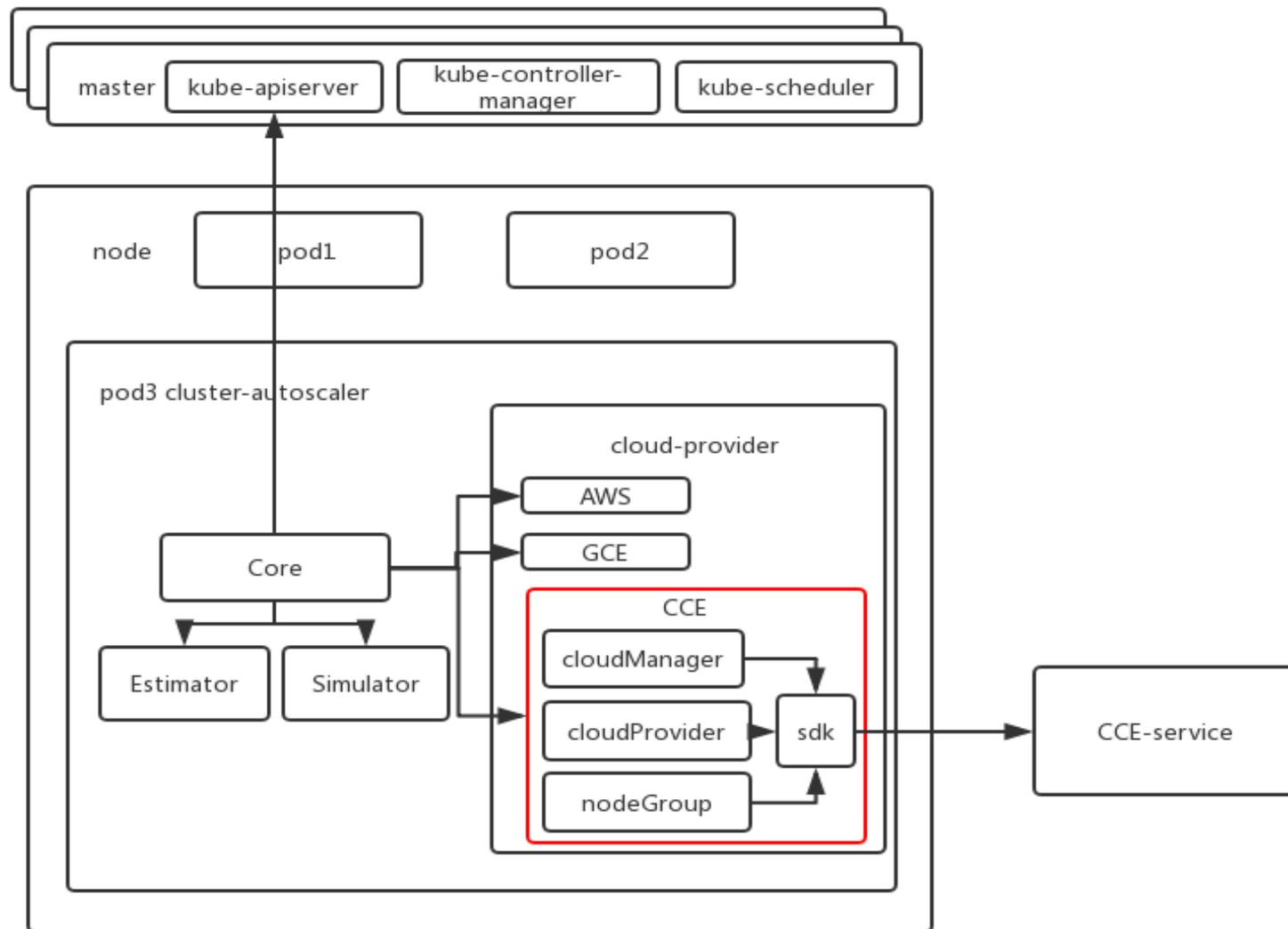
- 适用于共享场景下的文件存储
- 兼容NFS协议
- 通过External Provisioner支持动态创建

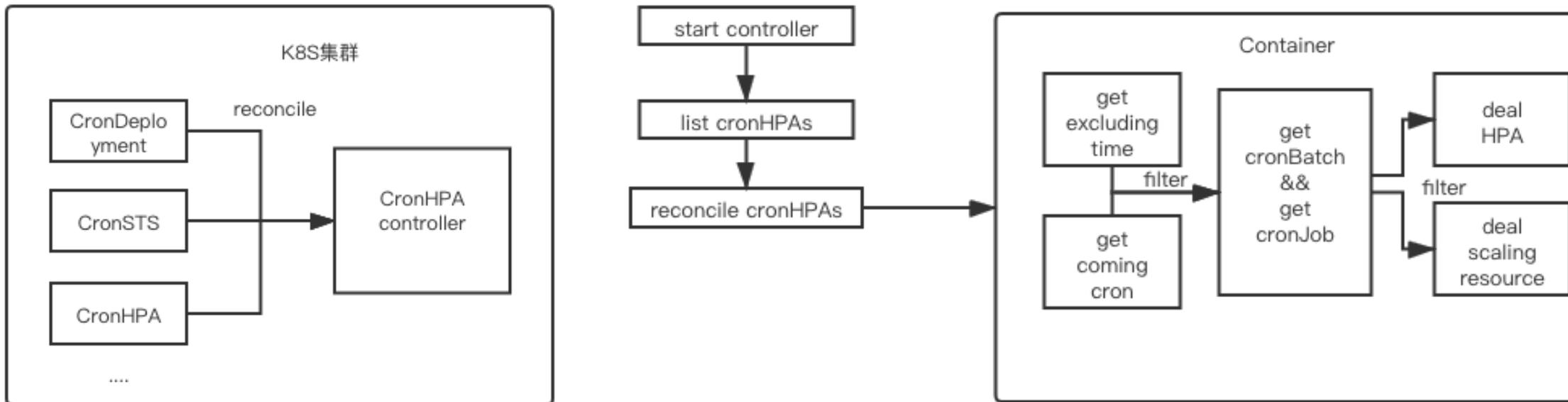
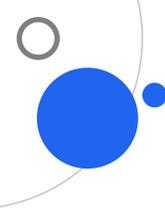


本地存储：提供Local PV和LVM CSI存储插件两种本地磁盘的使用模式，适用于不同使用场景。

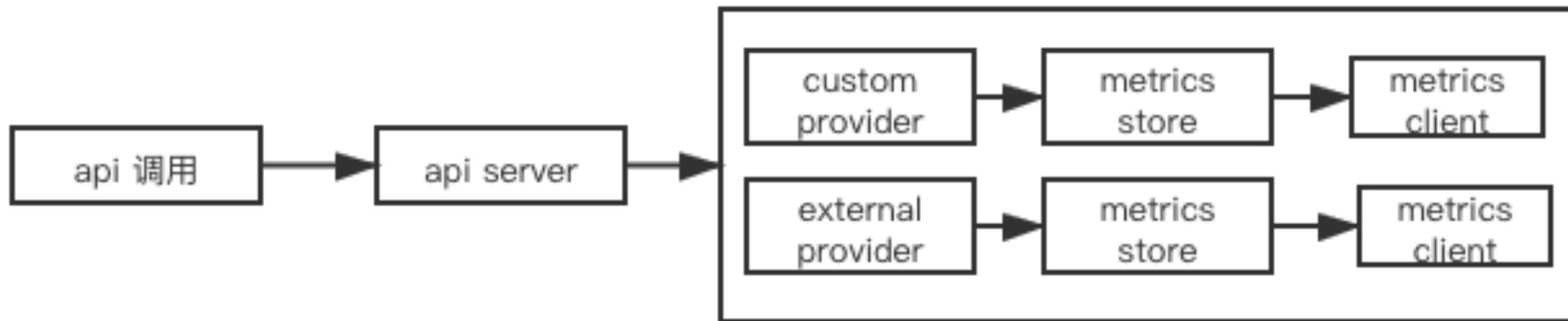


- Cluster-Autoscaler可以弹性伸缩kubernetes集群，根据部署的应用所请求的资源量来动态的伸缩集群
- 支持多种扩容策略、缩容设置
- 支持多伸缩组、GPU实例、竞价实例的伸缩
- 支持数分钟内伸缩上百节点
- CCE控制台界面操作
- Cluster-Autoscaler组件实时监控，集群伸缩情况实时掌控



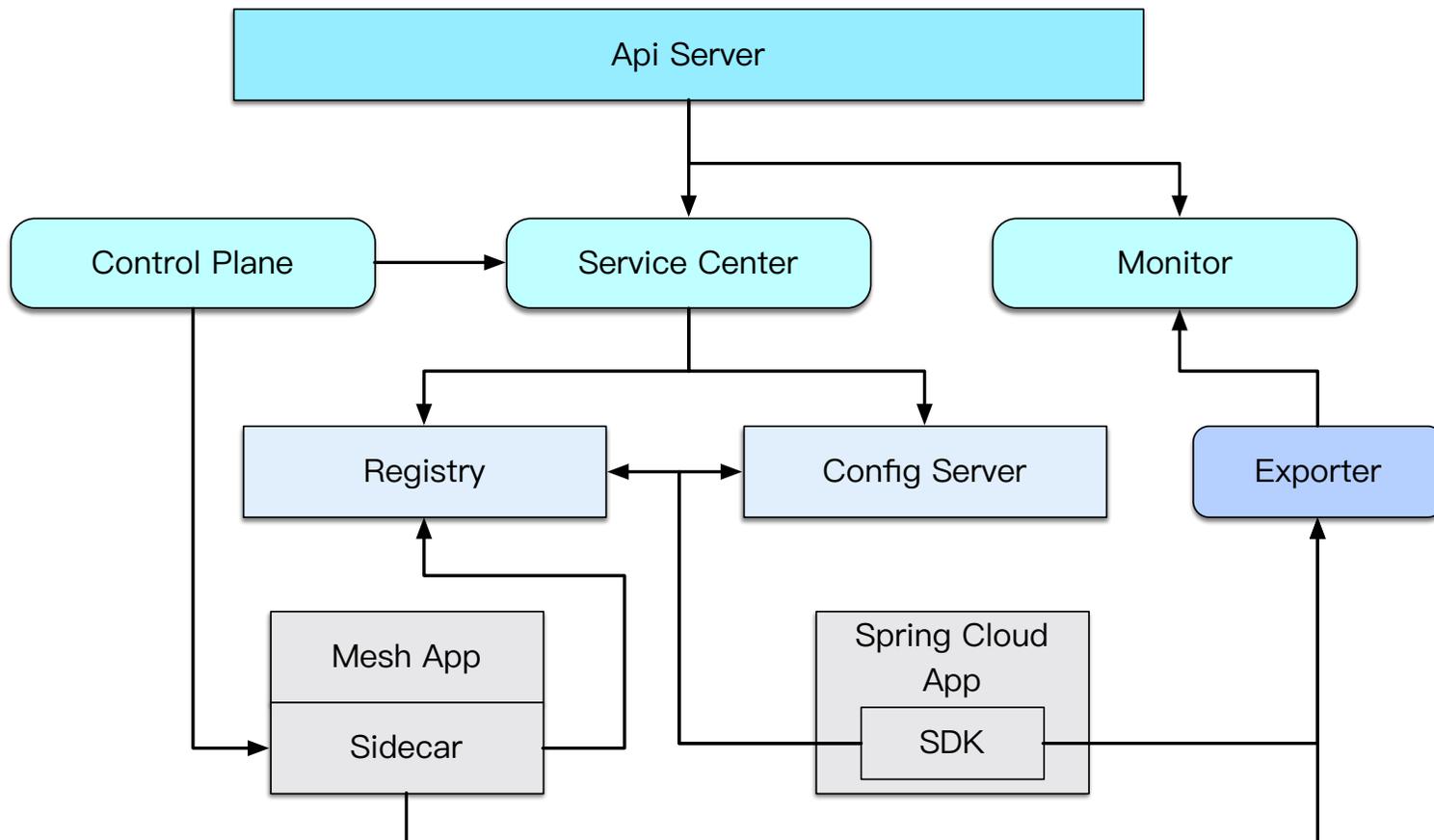


- 当出现Cron任务与HPA任务冲突时，controller会维持部署的最大的副本数，并让部署持续处于最大副本的状态
- CronHPA 可以根据用户设置的cron任务，周期、定时伸缩部署的副本数
- 支持与HPA的协同工作，极大扩展了弹性伸缩的应用场景
- 支持日期过滤功能
- 支持指定任务执行一次



- 集群原生支持基于CPU、MEM指标HPA
- 支持磁盘、网络、blb等其他指标的HPA就需要Metrics-adapter
- 该adapter基于社区的开源解决方案[custom-metrics-apiserver](#)
- 目前CCE只支持custom metrics的HPA，基于pod的磁盘、网络指标进行HPA

- Spring Cloud 业务低成本接入
- Service Mesh 业务零成本接入
- 异构微服务架构互访
- 服务治理能力丰富
- 链路追踪与服务拓扑能力兼容
- 容器集群+物理机/虚拟机集群





路由

- 基于Spring Cloud Ribbon实现
- 兼容 RestTemplate、AsyncRestTemplate及 OpenFeign
- 多种负载均衡策略
- 权重路由及优先级路由
- 自定义标签路由
- 多种容错策略(failfast/failover/failresend)



限流

- 基于Resilience4j RateLimiter实现
- 采用令牌桶算法
- 按标签限流
- 自定义限流策略/算法
- 采用Hystrix客户端限流



配置

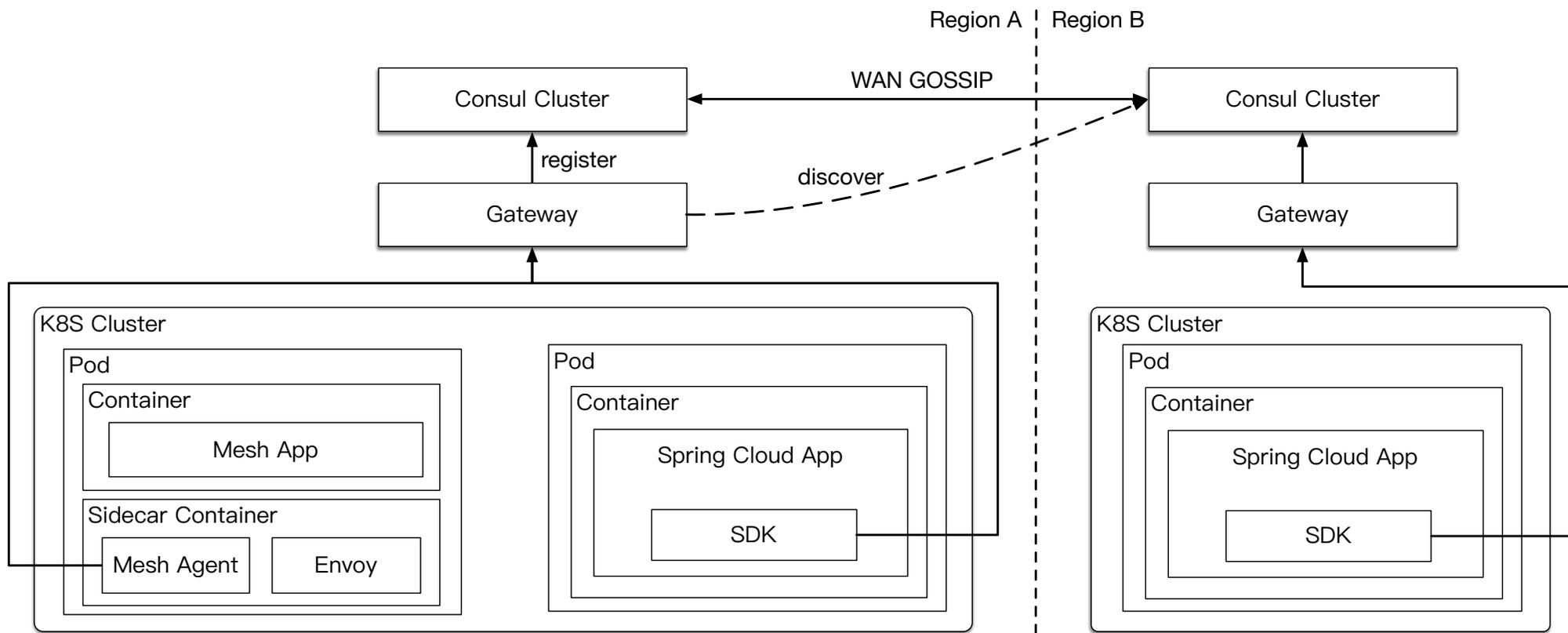
- 基于Spring Cloud Consul Config实现
- 实现HTTP Long Polling 的配置热更新
- 扩展Consul Token的安全认证机制
- 按优先级生效
- 配置灰度发布
- 微服务配置本地调试



熔断

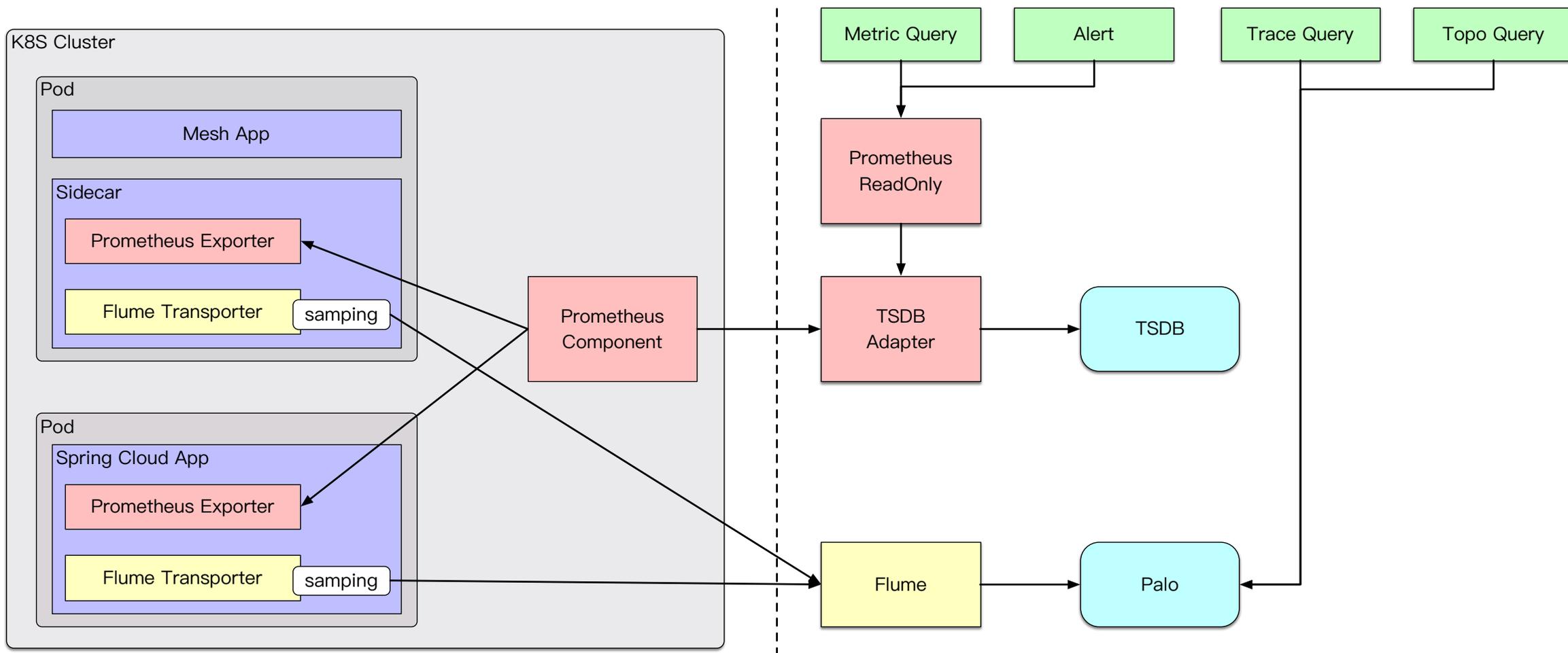
- 基于Resilience4j CircuitBreaker实现
- 符合Spring Cloud Circuit Breaker规范
- 支持RestTemplate和OpenFeign
- 方法维度、服务维度熔断
- 手动熔断
- 无侵入服务熔断

Spring Cloud Formula : <https://github.com/baidu/spring-cloud-formula>



- 注册中心多地域部署，每个地域一套集群
- 注册/心跳请求只访问同地域注册中心，Mesh应用由Agent完成注册心跳
- 支持跨地域服务发现，配合优先级路由功能支持业务多地域高可用

- Spring Cloud应用与Mesh应用拓扑及调用链打通，同时能够支持采样
- 服务指标数据通过集群内Prometheus组件拉取并存储到时序数据库，查询效率高





THANK YOU

CLOUD.BAIDU.COM