CHINA · BEIJING

# 基于 k8s 原生扩展的机器学习平台引擎 ML Engine

褚向阳

小米 高级软件工程师

SPEAKER

# INTRODUCE

## 褚向阳 高级软件工程师

- 2013 年毕业后加入红帽软件，吸收开源文化，接触 OpenStack 和 IaaS 平台相关技术。
- 2015 年底开始加入容器云创业公司数人云，参与打造容器化的 PaaS 平台。
- 2018 年从京东广告部加入小米人工智能部，负责小米机器学习平台的建设，重点支持各个框架的分布式训练，订制优化 K8s 调度，努力提高平台用户体验的同时保证集群利用率。
- 持续关注 Kubeflow 社区及性能优化相关开源项目发展。

TABLE OF

# CONTENTS 大纲

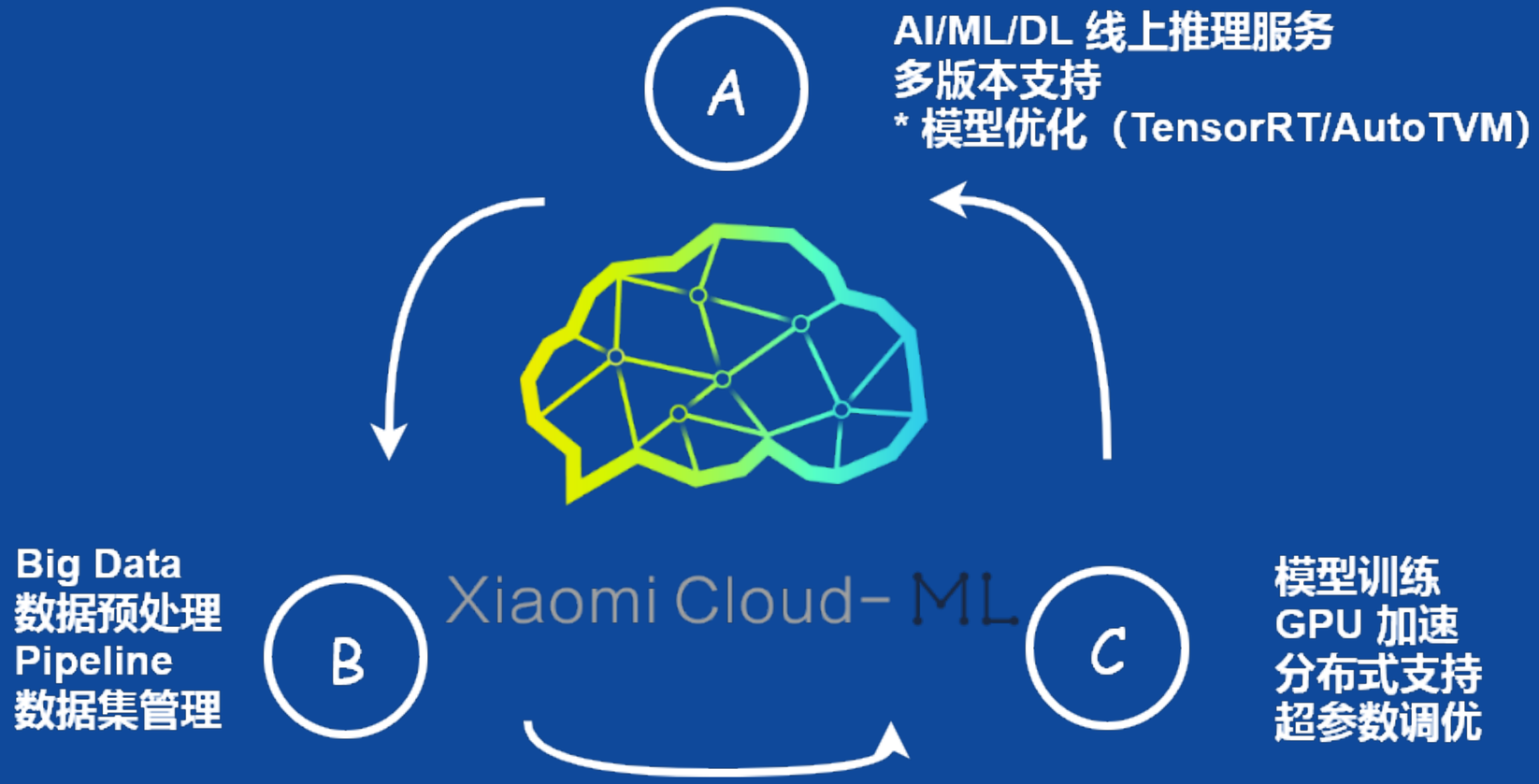
- 小米 CloudML 机器学习平台简介
- ML Engine 架构设计演进
- ML Engine 对多框架的分布式训练支持详解
- 未来发展方向和具体工作

TABLE OF

# CONTENTS 大纲

- 小米 CloudML 机器学习平台简介
- ML Engine 架构设计演进
- ML Engine 对多框架的分布式训练支持详解
- 未来发展方向和具体工作

# CloudML & ABC


A
AI/ML/DL 线上推理服务
多版本支持
* 模型优化（TensorRT/AutoTVM）
Big Data
数据预处理
Pipeline
数据集管理
B
Xiaomi Cloud-ML
C
模型训练
GPU 加速
分布式支持
超参数调优

# 平台全景


应用业务层
图像
NLP
声学
搜索推荐
MIot
平台接入层
多租户
Pipeline
训练任务
模型服务
日志
监控告警
计量计费
模型框架层
Tensorflow
Pytorch
Caffe
Kaldi
定制
资源管理层
K8S
FDS
HDFS
CEPH
基础设施层
金山云
阿里云
独立集群

# 平台全景


核心逻辑

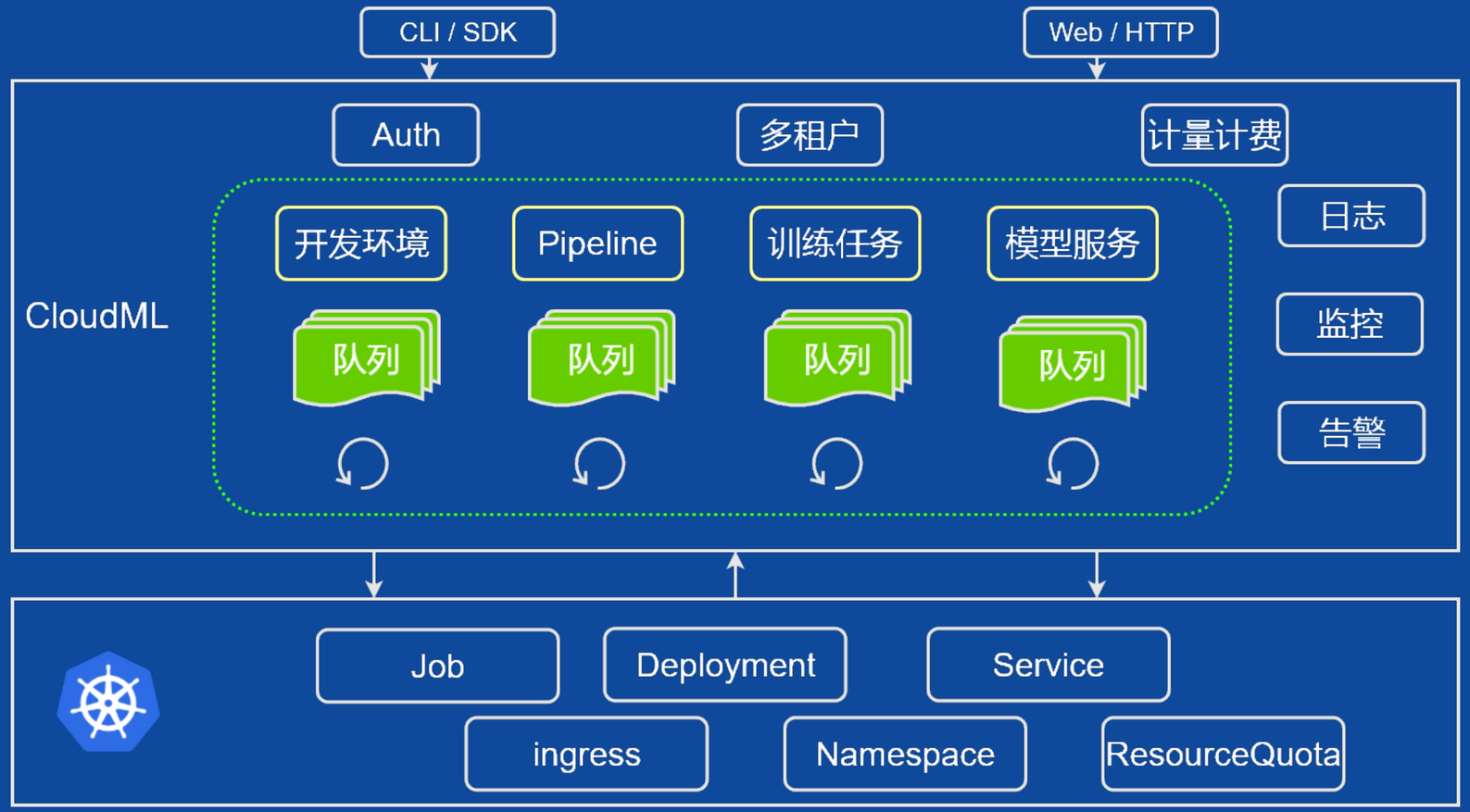
ML Engine

TABLE OF

# CONTENTS 大纲

- 小米 CloudML 机器学习平台简介
- **ML Engine 架构设计演进**
- ML Engine 对多框架的分布式训练支持详解
- 未来发展方向和具体工作

# CloudML 架构


CLI / SDK
Web / HTTP
Auth
多租户
计量计费
CloudML
开发环境
Pipeline
训练任务
模型服务
队列
队列
队列
队列
日志
监控
告警
Job
Deployment
Service
ingress
Namespace
ResourceQuota

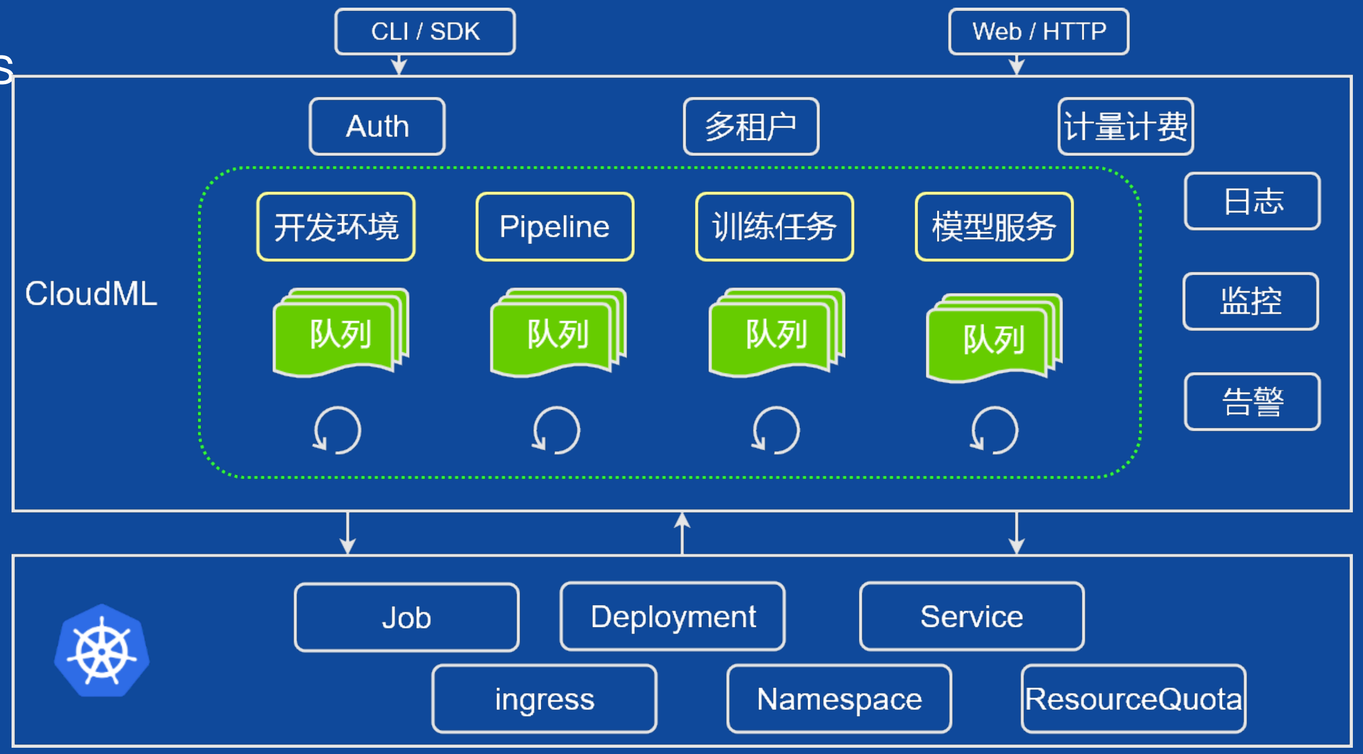
# 架构和挑战

机器学习平台架构

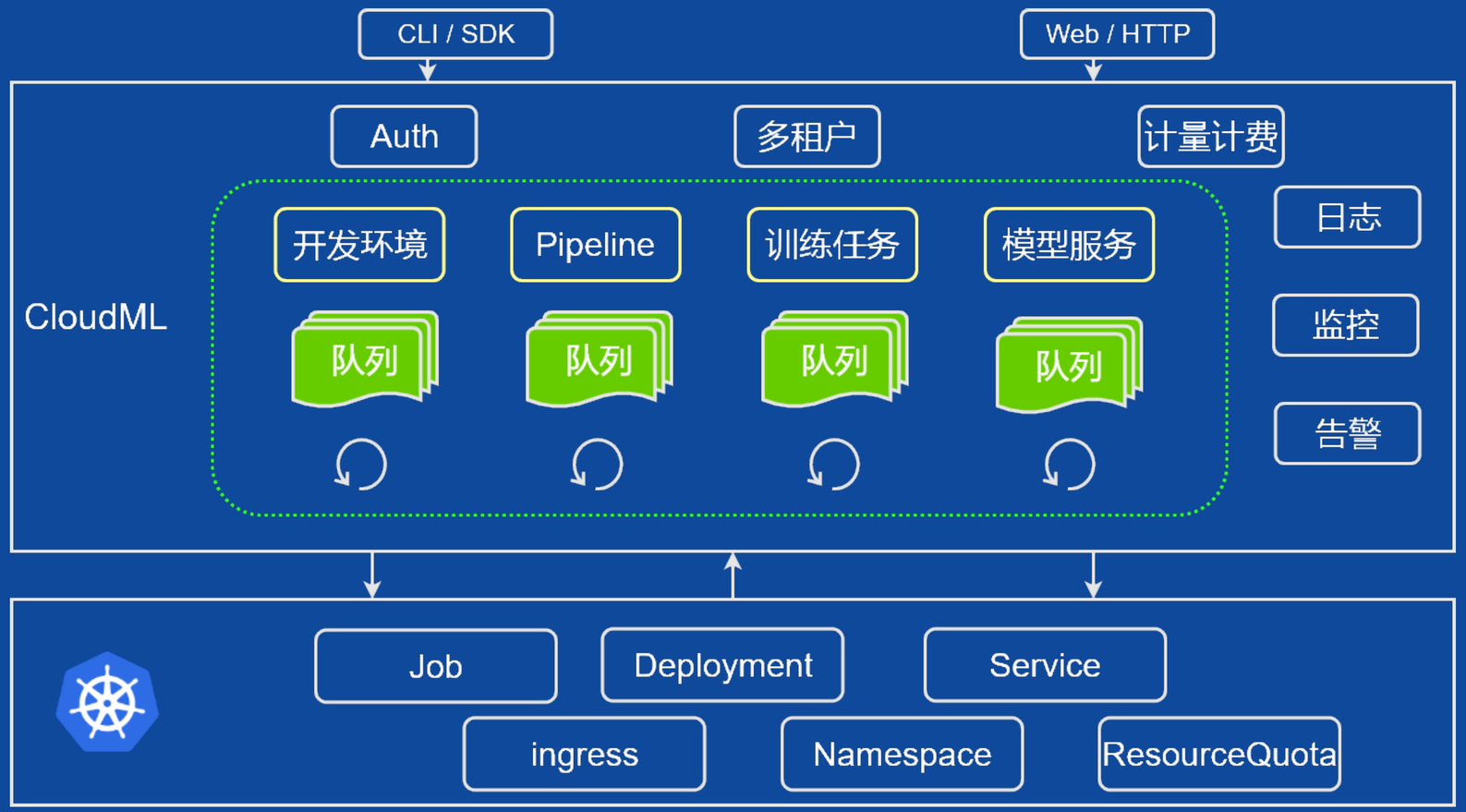
- 分布式训练任务 –> 一组 Job/Pod + SVC
- 模型服务 –> Deployment + SVC + Ingress

挑战：

- 队列、状态同步
- 生命周期管理
- 调度策略



# CloudML 架构




ML Engine

?

The easiest way to install Kubernetes is to get someone else to do it for you.

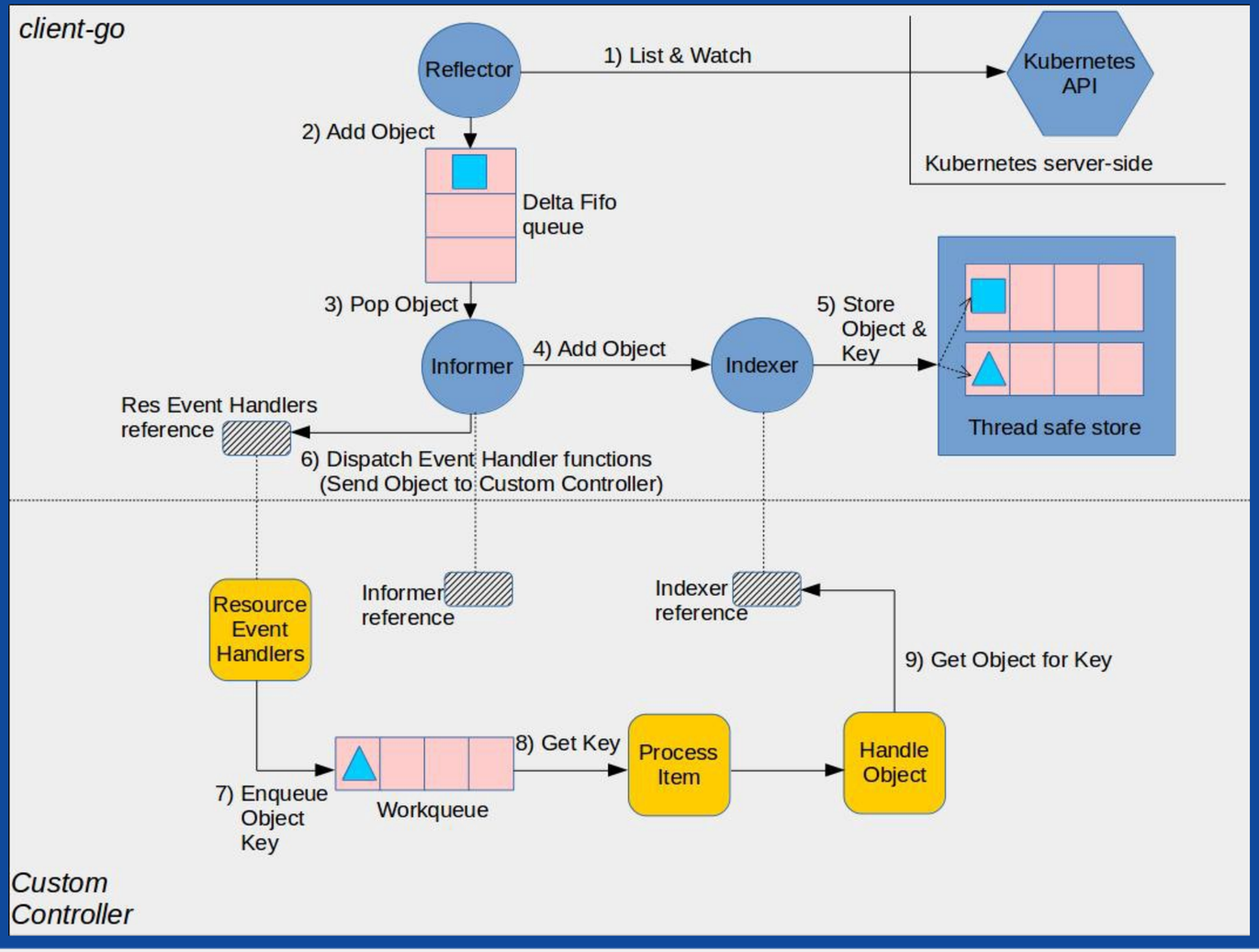
#LISA19, @jpetazzo

The easiest way to Build a
AI /ML platform is to let k8s
to do it for you.

#ArchSummit19, me

# K8S 扩展资源定义

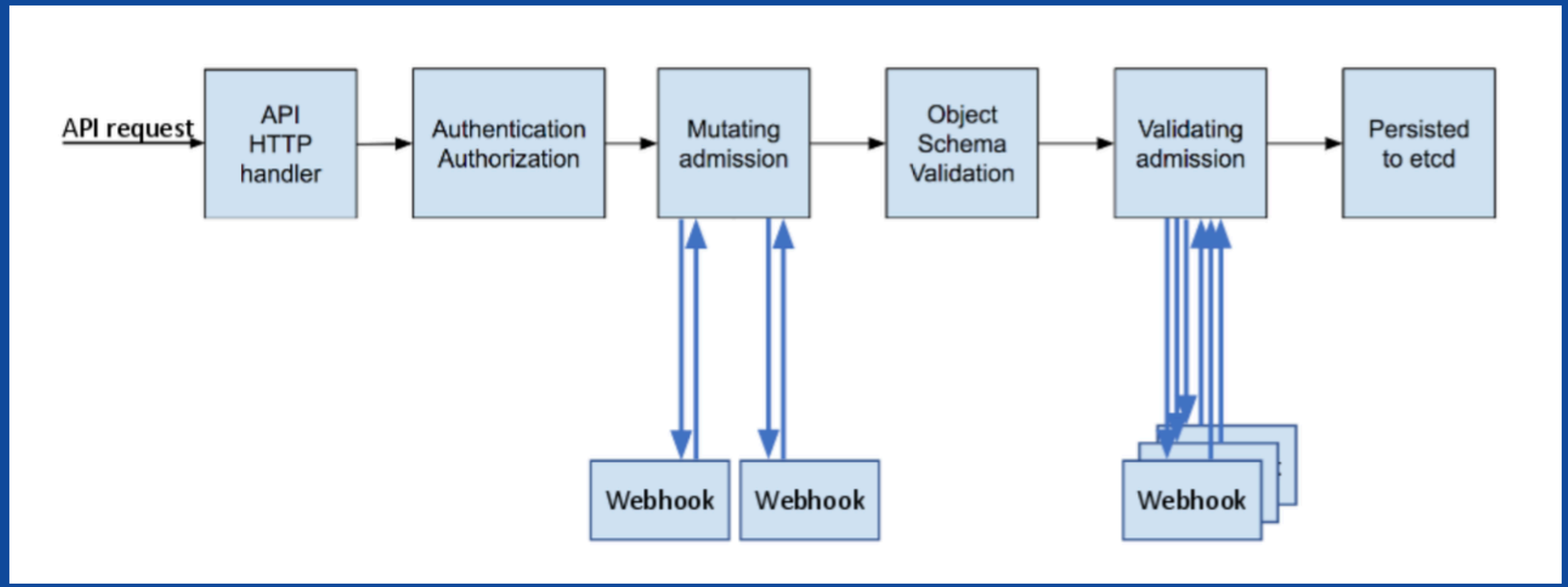
CRD & Controller




https://medium.com/@cloudark/kubernetes-custom-controllers-b6c7d0668fdf

# Mutating & Validating

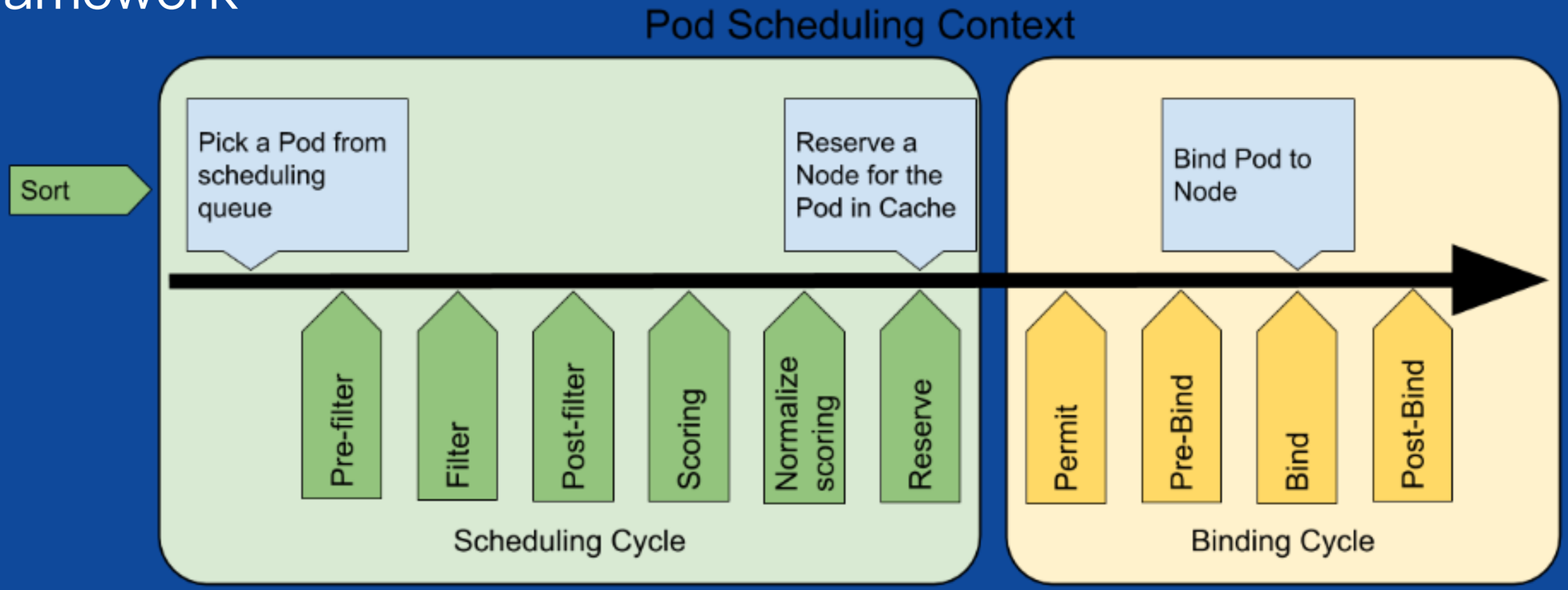
Admission webbooks




https://kubernetes.io/blog/2019/03/21/a-guide-to-kubernetes-admission-controllers/

# 调度优化

Scheduling framework




https://kubernetes.io/docs/concepts/configuration/scheduling-framework/

# 扩展工具包

Kube-builder

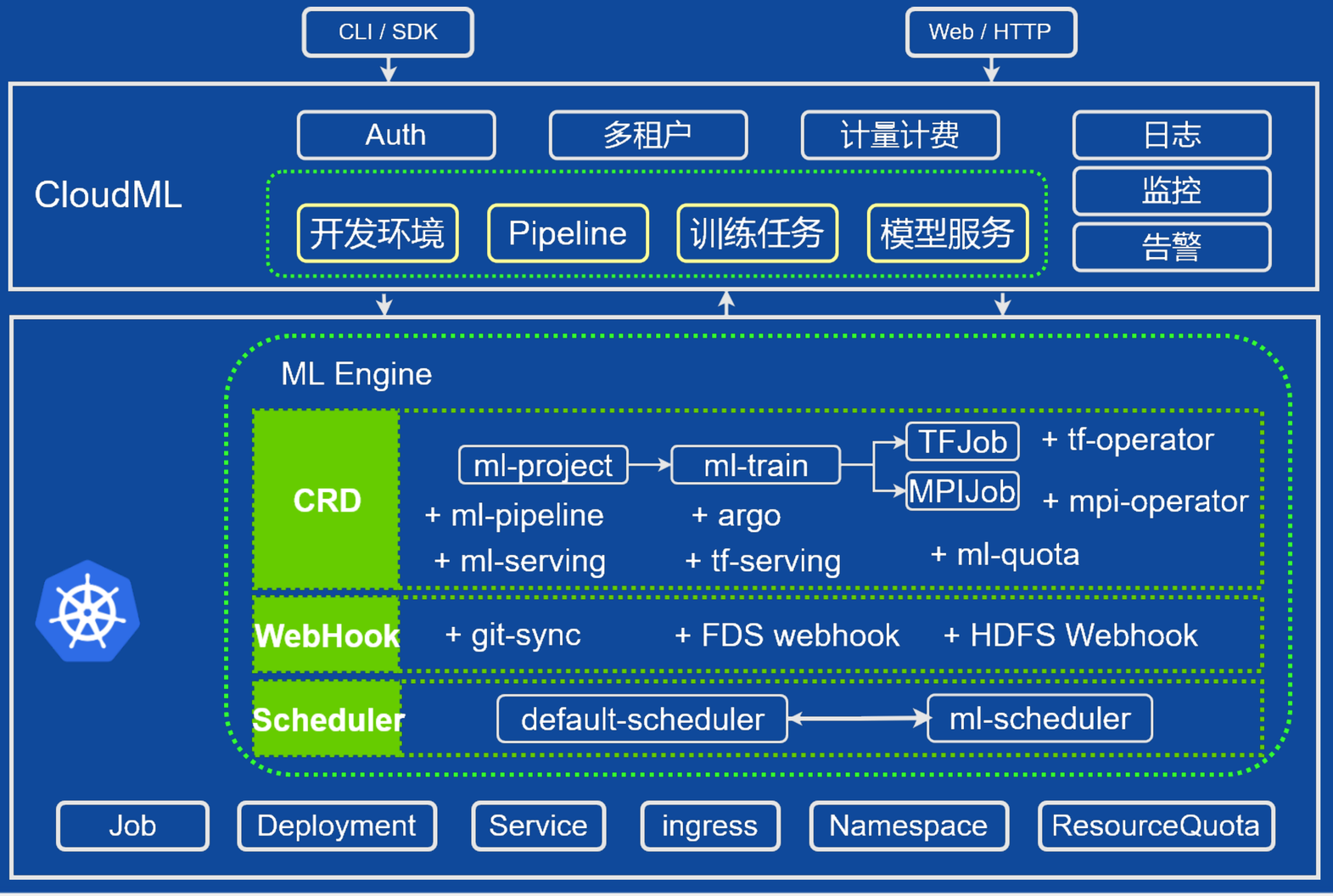
- https://github.com/kubernetes-sigs/kubebuilder

Operator Framework/Operator SDK

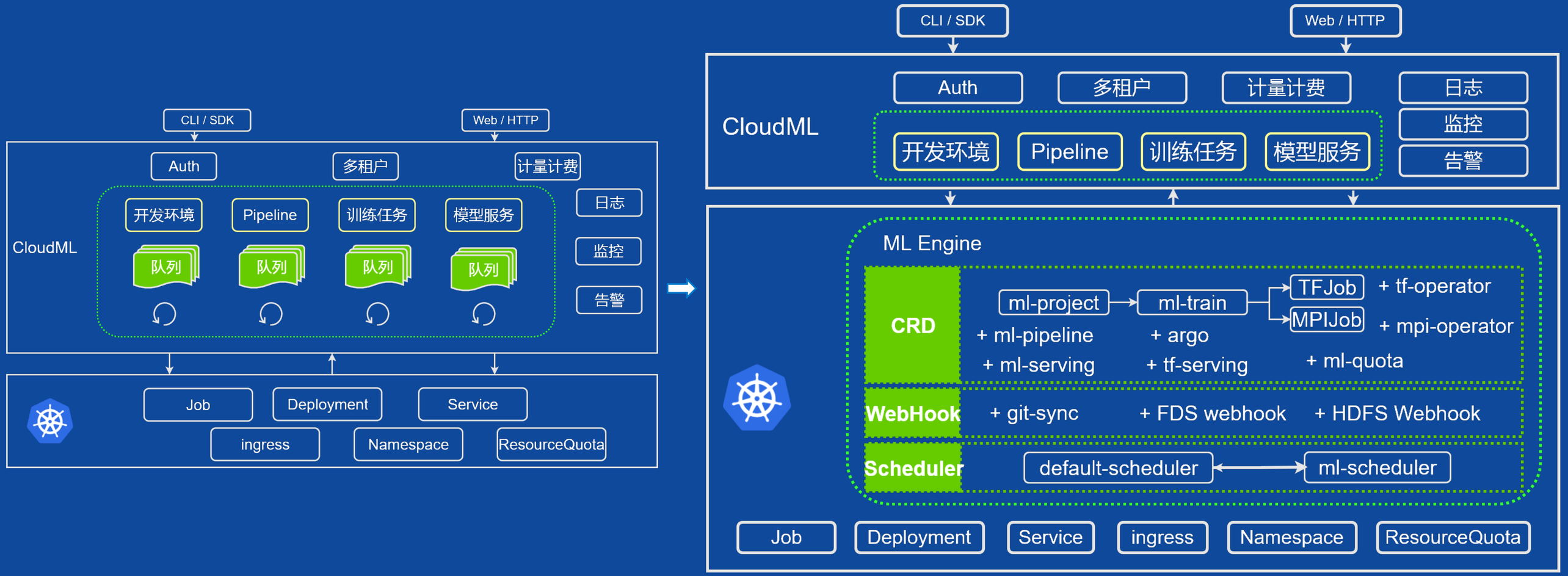
- https://github.com/operator-framework/operator-sdk

```
$ # Initialize Go modules
$ go mod init demo.kubebuilder.io
go: creating new go.mod: module demo.kubebuilder.io
$ # Let's initialize the project
$ kubebuilder init --domain tutorial.kubebuilder.io
go get sigs.k8s.io/controller-runtime@v0.2.2
go mod tidy
Running make...
make
$HOME/go/bin/controller-gen object:headerFile=./hack/boilerplate.go.txt paths="./..."
go fmt ./...
go vet ./...
go build -o bin/manager main.go
```

# ML Engine 最终架构



# 架构对比


CLI / SDK
Web / HTTP
Auth
多租户
计量计费
日志
监控
告警
CloudML
开发环境
Pipeline
训练任务
模型服务
队列
队列
队列
队列
Job
Deployment
Service
ingress
Namespace
ResourceQuota
CLI / SDK
Web / HTTP
Auth
多租户
计量计费
日志
监控
告警
CloudML
开发环境
Pipeline
训练任务
模型服务
ML Engine
CRD
ml-project
ml-train
TFJob
+ tf-operator
MPIJob
+ mpi-operator
+ ml-pipeline
+ argo
+ ml-serving
+ tf-serving
+ ml-quota
WebHook
+ git-sync
+ FDS webhook
+ HDFS Webhook
Scheduler
default-scheduler
ml-scheduler
Job
Deployment
Service
ingress
Namespace
ResourceQuota

# ML Engine 架构设计演进

ML Engine 新版架构总结

- 机器学习平台业务的合理抽象 CRD
- 针对机器学习任务特性的定制调度器
- 公共的 validating/mutating 逻辑
- 提供统一的对外接口

TABLE OF

# CONTENTS 大纲

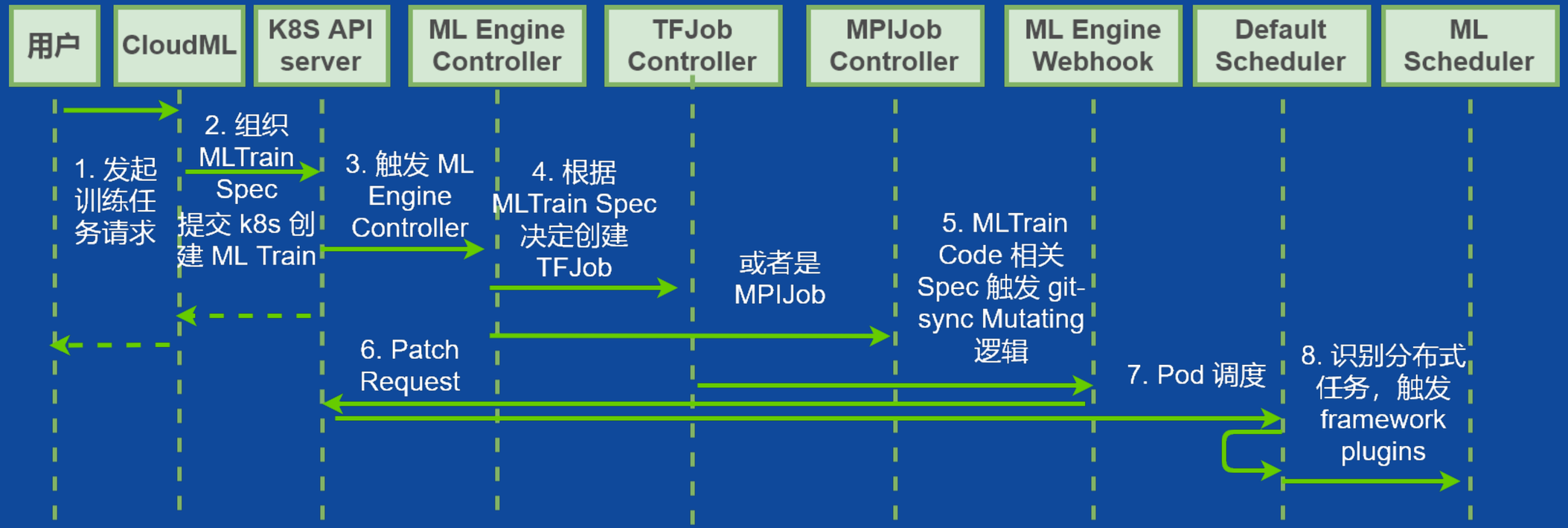
- 小米 CloudML 机器学习平台简介
- ML Engine 架构设计演进
- **ML Engine 对多框架的分布式训练支持详解**
- 未来发展方向和具体工作

# ML Engine 训练任务需求

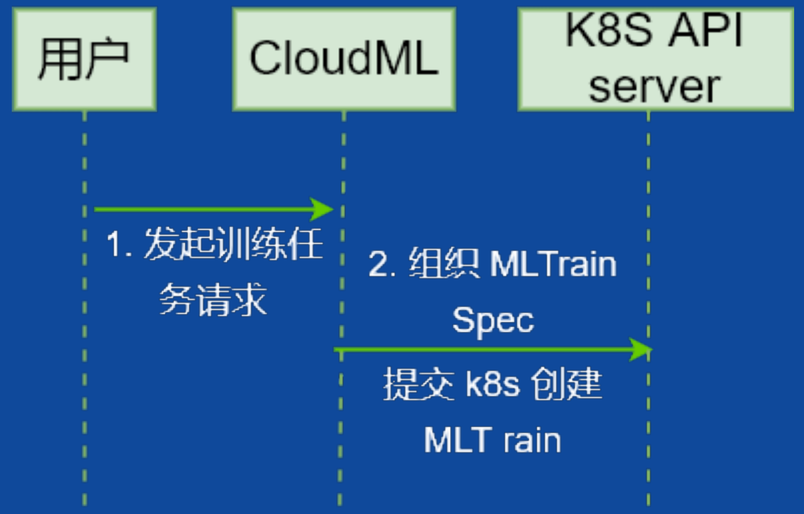
核心用例：

- 用户给定训练代码及启动命令（可选：定义数据及产出物）
- 支持选择不同的机器学习框架及运行时环境（CPU/GPU）
- 下发集群，需要支持分布式启动训练，且需要遵循一定的调度优化策略

# ML Engine 训练任务流程



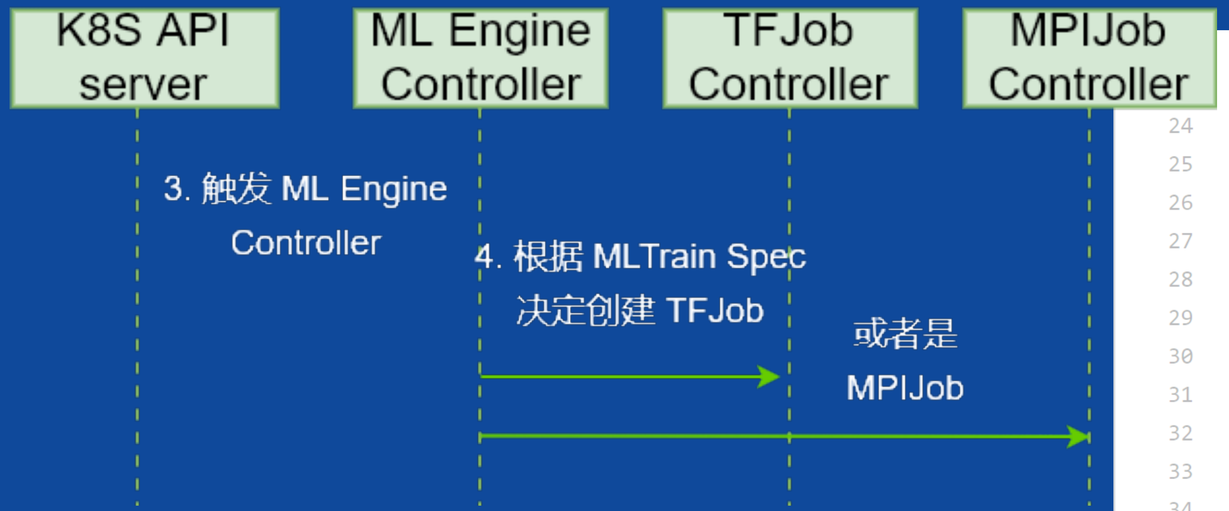
# ML Engine 训练任务定义




```go
// TrainSpec defines the desired state of Train
type TrainSpec struct {
        // INSERT ADDITIONAL SPEC FIELDS - desired state of cluster
        // Important: Run "make" to regenerate code after modifying this file
        ClusterSpec map[TrainRoleName]TrainRoleSpec `json:"cluster_spec,omitempty"`

        // Framework like: tf, mpi, pytorch, default: tf
        // +optional
        Framework string `json
        // +optional
        Image string `json:"im
        // +optional
        Command []string `json
        // +optional
        Args []string `json:"a

        // +optional
        Env []corev1.EnvVar `j
}
```

```go
// TrainRoleSpec defines the desired state of Train Cluster Role
// e.g. chief, worker
type TrainRoleSpec struct {
        Count int32 `json:"count"`
        // +optional
        Image     string                      `json:"image,omitempty"`
        Resources corev1.ResourceRequirements `json:"resources,omitempty"`
        // TODO(xychu): add role level cmd and args support
        //// +optional
        //Command []string `json:"command,omitempty"`
        //// +optional
        //Args []string `json:"args,omitempty"`

        // +optional
        Env []corev1.EnvVar `json:"env,omitempty"`

        // NOTE(xychu): use affinity to support node selector and other placement requir
        // +optional
        Affinity *corev1.Affinity
}
```

# ML Engine 训练任务定义




kubeflow/tf–operator

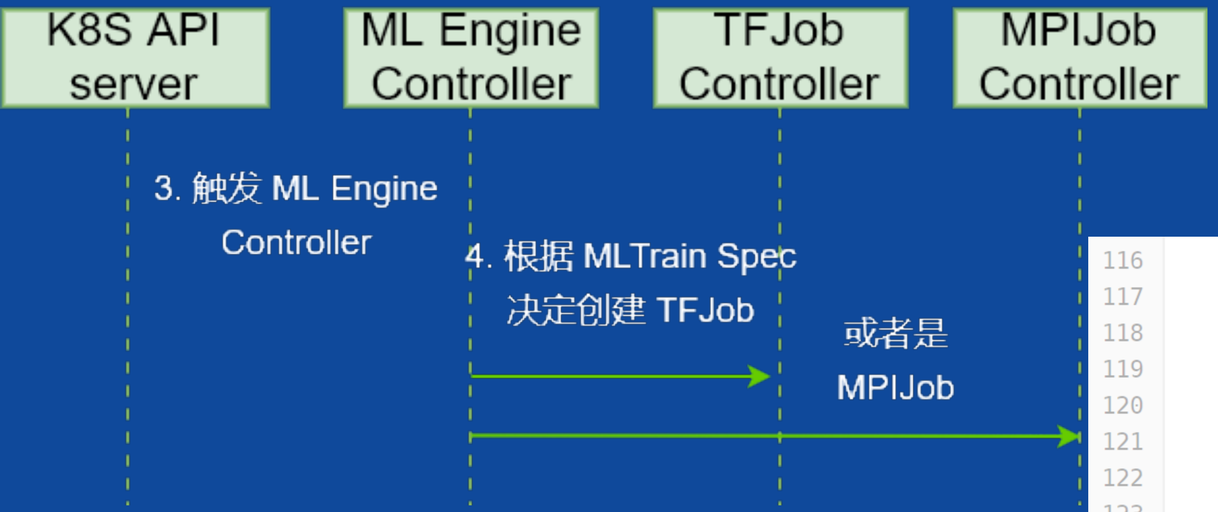
kubeflow/mpi–operator

…

```
// +genclient
// +k8s:deepcopy-gen:interfaces=k8s.io/apimachinery/pkg/runtime.Object
// +resource:path=tfjob

// Represents a TFJob resource.
type TFJob struct {
        // Standard Kubernetes type
        metav1.TypeMeta `json:",inli

        // Standard Kubernetes objec
        metav1.ObjectMeta `json:"met

        // Specification of the desi
        Spec TFJobSpec `json:"spec,c

        // Most recently observed st
        // Read-only (modified by th
        Status common.JobStatus `jsc
}
```

```
// TFJobSpec is a desired state description of the TFJob.
type TFJobSpec struct {
        // Specifies the duration (in seconds) since startTime during which the job ca
        // before it is terminated. Must be a positive integer.
        // This setting applies only to pods where restartPolicy is OnFailure or Alway
        // +optional
        ActiveDeadlineSeconds *int64 `json:"activeDeadlineSeconds,omitempty"`

        // Number of retries before marking this job as failed.
        // +optional
        BackoffLimit *int32 `json:"backoffLimit,omitempty"`

        // Defines the policy for cleaning up pods after the TFJob completes.
        // Defaults to Running.
        CleanPodPolicy *common.CleanPodPolicy `json:"cleanPodPolicy,omitempty"`

        // Defines the TTL for cleaning up finished TFJobs (temporary
        // before kubernetes adds the cleanup controller).
        // It may take extra ReconcilePeriod seconds for the cleanup, since
        // reconcile gets called periodically.
        // Defaults to infinite.
        TTLSecondsAfterFinished *int32 `json:"ttlSecondsAfterFinished,omitempty"`

        // A map of TFReplicaType (type) to ReplicaSpec (value). Specifies the TF clus
        // For example,
        //   {
        //     "PS": ReplicaSpec,
        //     "Worker": ReplicaSpec,
        //   }
        TFReplicaSpecs map[TFReplicaType]*common.ReplicaSpec `json:"tfReplicaSpecs"`
}
```

# ML Engine 多框架支持



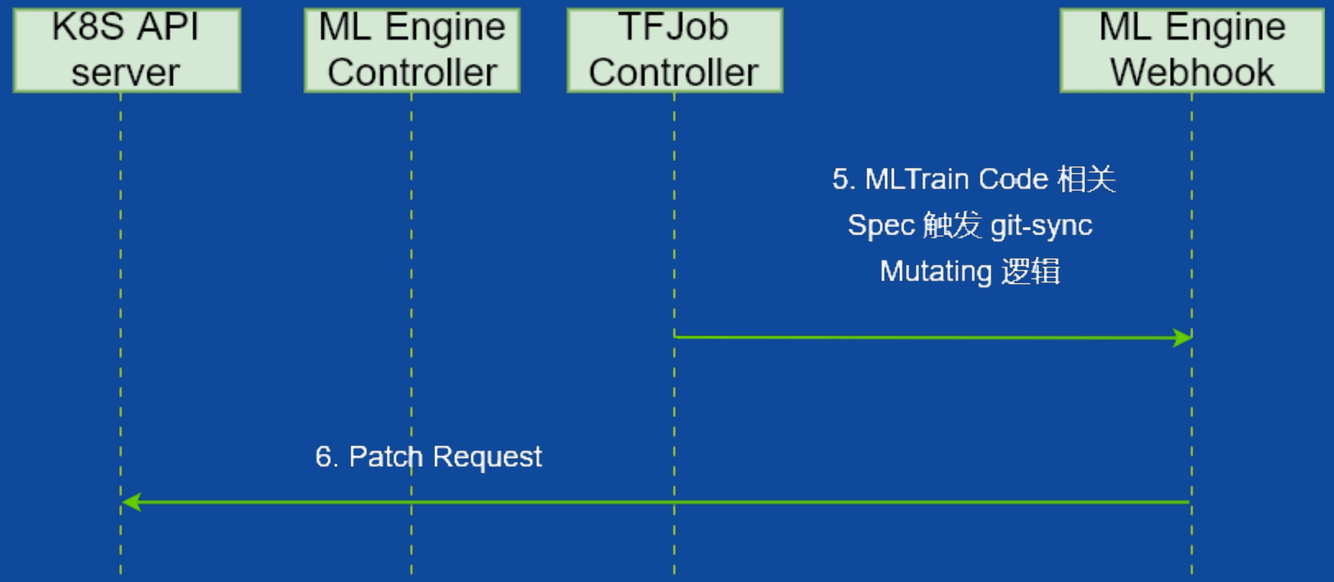

插件化设计支持

```
116         // TODO(user): Change this to be the object type created by your controller
117         // Define the desired TrainJob object
118         if handler, ok := HandlerMap[instance.Spec.Framework]; ok {
119                 reconcileResult, err := handler.Handle(instance, r.Client, r.scheme)
120                 if err != nil {
121                         return reconcileResult, err
122                 }
123                 // Call Update status
124                 err = r.Status().Update(context.Background(), instance)
125                 if err != nil {
126                         return reconcile.Result{}, err
127                 }
128                 return reconcileResult, nil
129         } else {
130                 msg := fmt.Sprintf("Unsupported Train Framework found %s\n", instance.Spec.Framewor
131                 log.Info(msg, "handlers", HandlerMap)
132                 err = errors.NewBadRequest(msg)
133                 return reconcile.Result{}, err
        }
}
```

解决问题：

- 用户接口层统一
- 各个框架状态统一

# Train 代码拉取




Mutating Webhook

- git-sync

https://github.com/kubernetes/git-sync

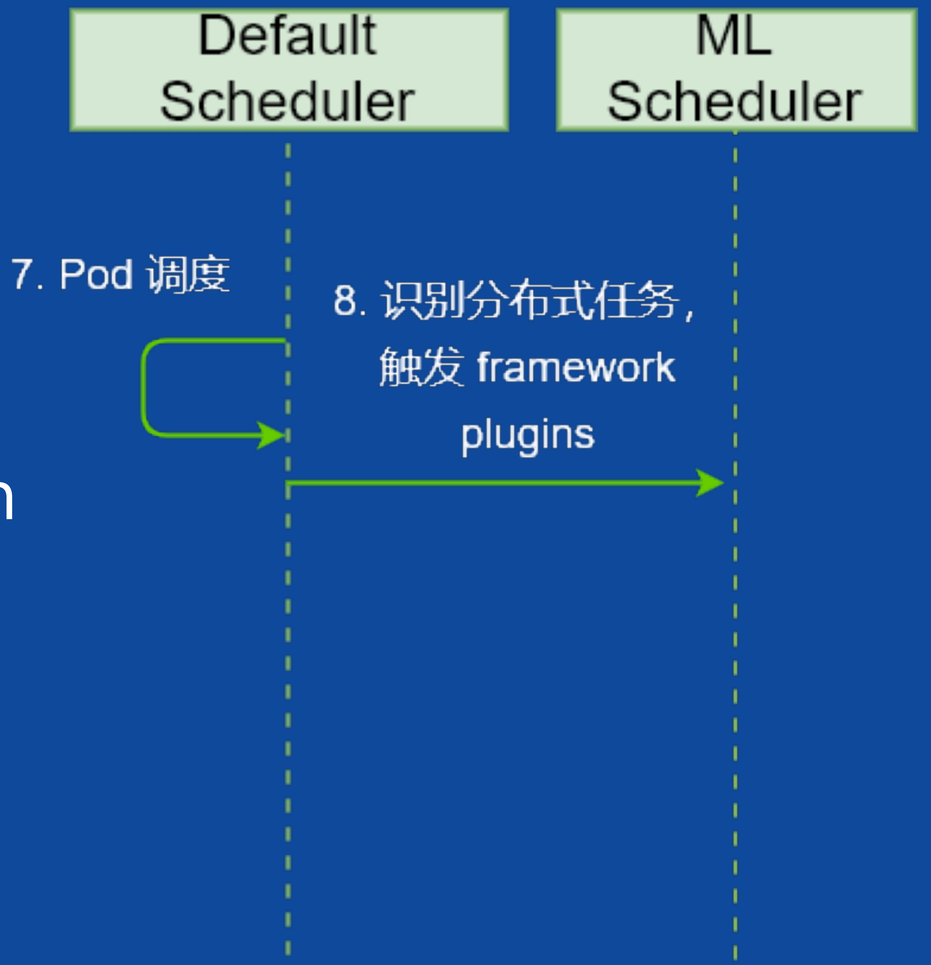
```
# ...
containers:
- name: git-sync
  image: k8s.gcr.io/git-sync:v9.3.76
  args:
   - "-ssh"
   - "-repo=git@github.com:foo/bar"
   - "-dest=bar"
   - "-branch=master"
  volumeMounts:
  - name: git-secret
    mountPath: /etc/git-secret
  securityContext:
    runAsUser: 65533 # git-sync user
# ...
```

# ML Engine 调度优化

GPU Mosted Requested Priority

Distributed Coscheduling Framework plugin

Affinity/Anti–Affinity



# 心得总结

优点：

- K8s 原生的扩展性
- 服务的可用性保证
- 服务的易维护性

缺点：

- admission webhook 双刃剑，建议设置过滤条件
- 多框架支持，各个框架不同版本之间的 golang 依赖

TABLE OF

# CONTENTS 大纲

- 小米 CloudML 机器学习平台简介
- ML Engine 架构设计演进
- ML Engine 对多框架的分布式训练支持详解
- **未来发展方向和具体工作**

# 未来发展方向和具体工作

功能加强

- 考虑将计量/计费也抽象成 CRD
- 提供更方便的数据管理逻辑
- 更丰富的模型服务管理能力

时机成熟后会积极尝试开源

欢迎加入~