拉勾教育
—互联网人实战大学—

《Kubernetes 原理剖析与实战应用》

正范

— 拉勾教育出品 —

拉勾教育
—互联网人实战大学—

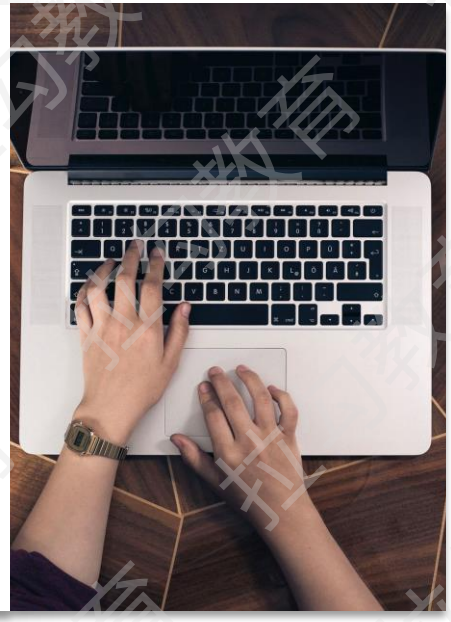
27 | K8s CRD：如何根据需求自定义你的 API？

前言

Kubernetes 中内置的对象定义

如 **Deployment、StatefulSet、Configmap**

可能已经不能满足需求



前言

Kubernetes

自定义对象方式

聚合 API

聚合 API（Aggregation API，AA）

主要目的是方便用户将自己定义的 API 注册到 kube-apiserver 中

并且可以像使用其它内置的 API 一样，通过 APIServer 的 URL 就可以访问和操作

聚合 API


```
--requestheader-client-ca-file=<path to aggregator CA cert>
--requestheader-allowed-names=front-proxy-client
--requestheader-extra-headers-prefix=X-Remote-Extra-
--requestheader-group-headers=X-Remote-Group
--requestheader-username-headers=X-Remote-User
--proxy-client-cert-file=<path to aggregator proxy cert>
--proxy-client-key-file=<path to aggregator proxy key>
```

聚合 API


```
--requestheader-client-ca-file=<path to aggregator CA cert>
--requestheader-allowed-names=front-proxy-client
--requestheader-extra-headers-prefix=X-Remote-Extra-
--requestheader-group-headers=X-Remote-Group
--requestheader-username-headers=X-Remote-User
--proxy-client-cert-file=<path to aggregator proxy cert>
--proxy-client-key-file=<path to aggregator proxy key>
```

https://kubernetes.io/zh/docs/tasks/extend-kubernetes/setup-extension-api-server/

聚合 API

1

Kubernetes apiserver 会判断 --requestheader-client-ca-file 指定的 CA 证书中的 CN 是否是 --requestheader-allowed-names 提供的列表名称之一

如果不是，则该请求被拒绝

如果名称允许，则请求会被转发

聚合 API

1

Kubernetes apiserver 会判断 --requestheader-client-ca-file 指定的 CA 证书中的 CN 是否是 --requestheader-allowed-names 提供的列表名称之一

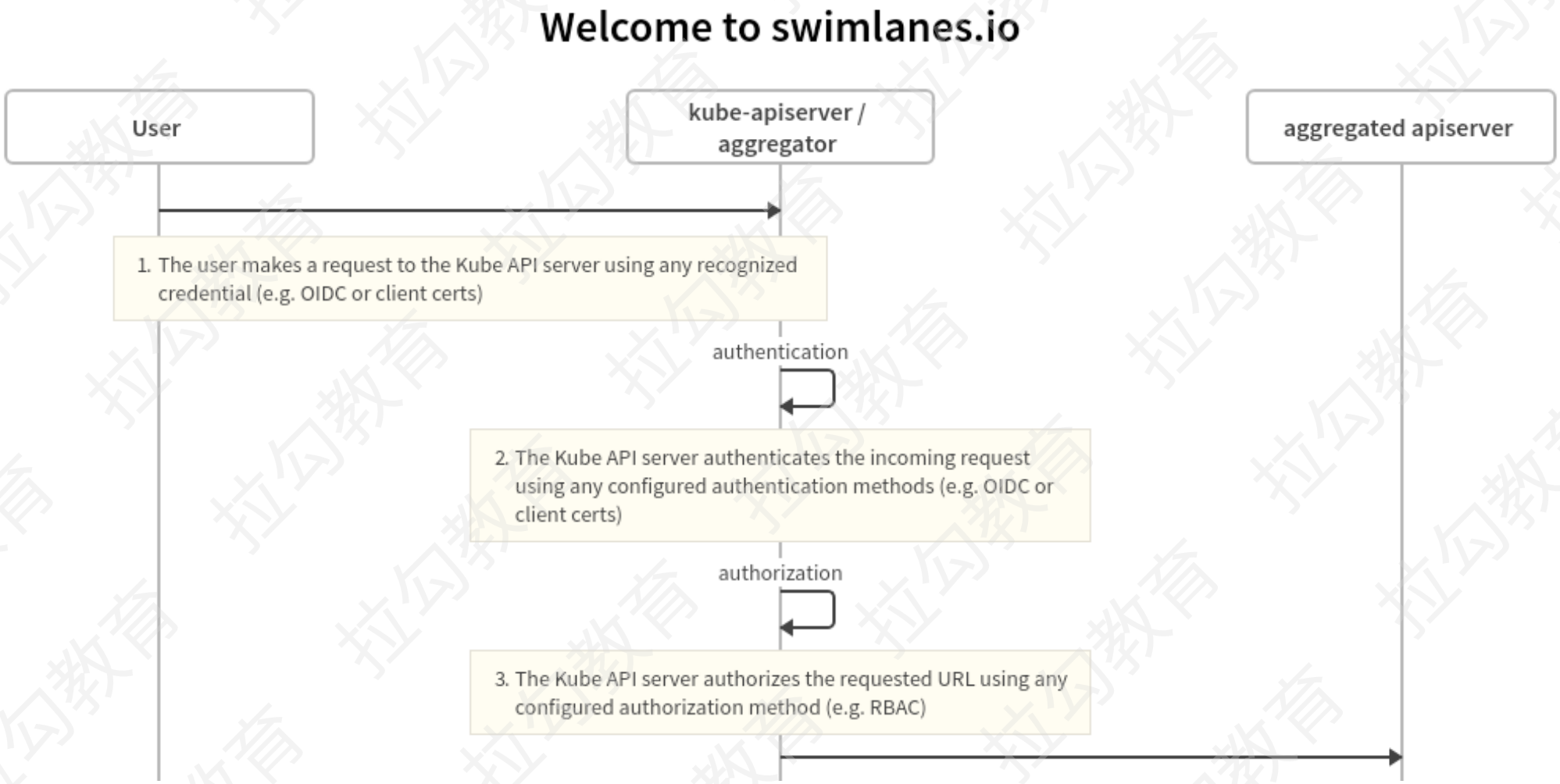
如果不是，则该请求被拒绝

如果名称允许，则请求会被转发

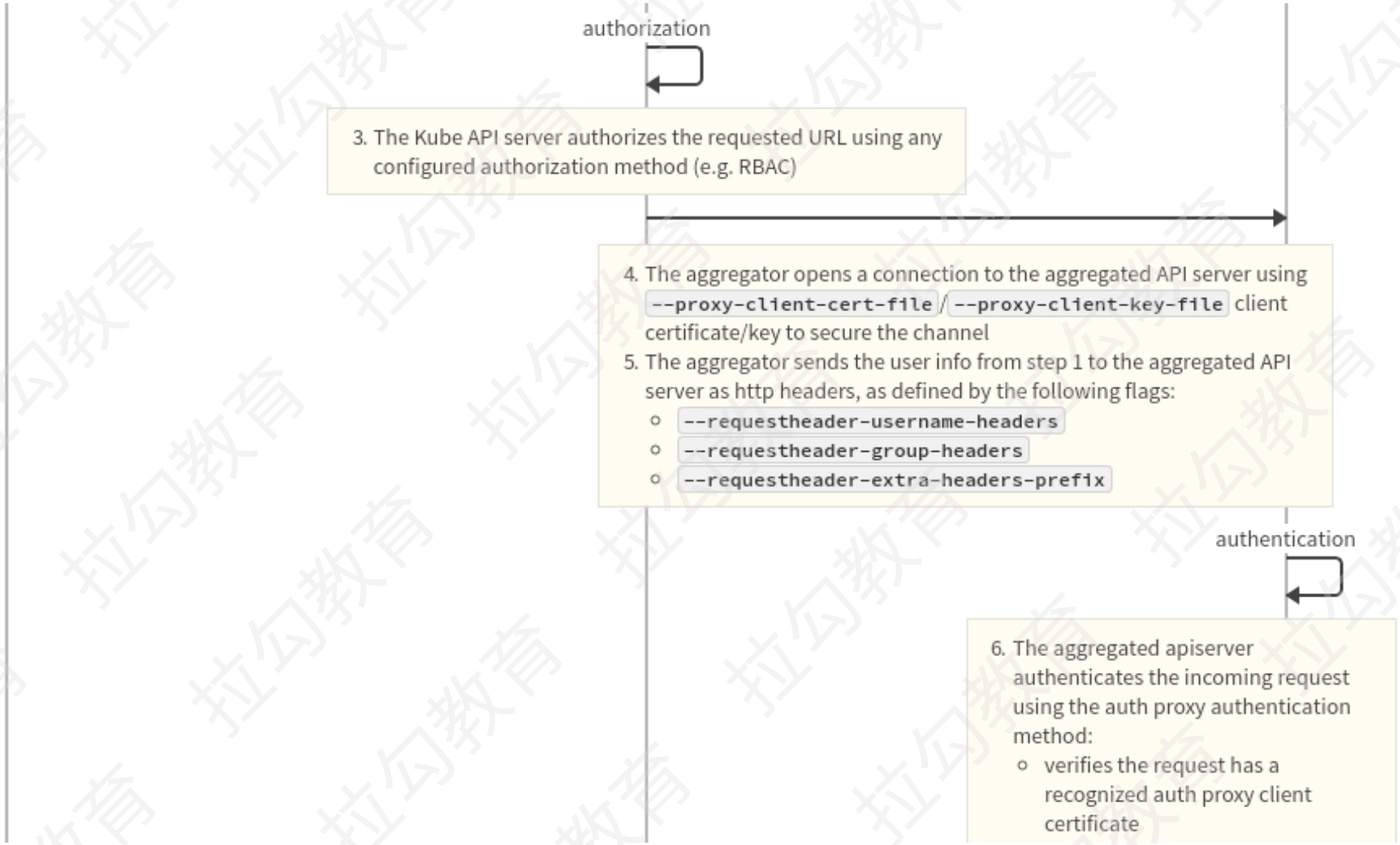
2

Kubernetes apiserver 使用由 --proxy-client-*-file 指定的文件来访问用户的扩展 APIServer

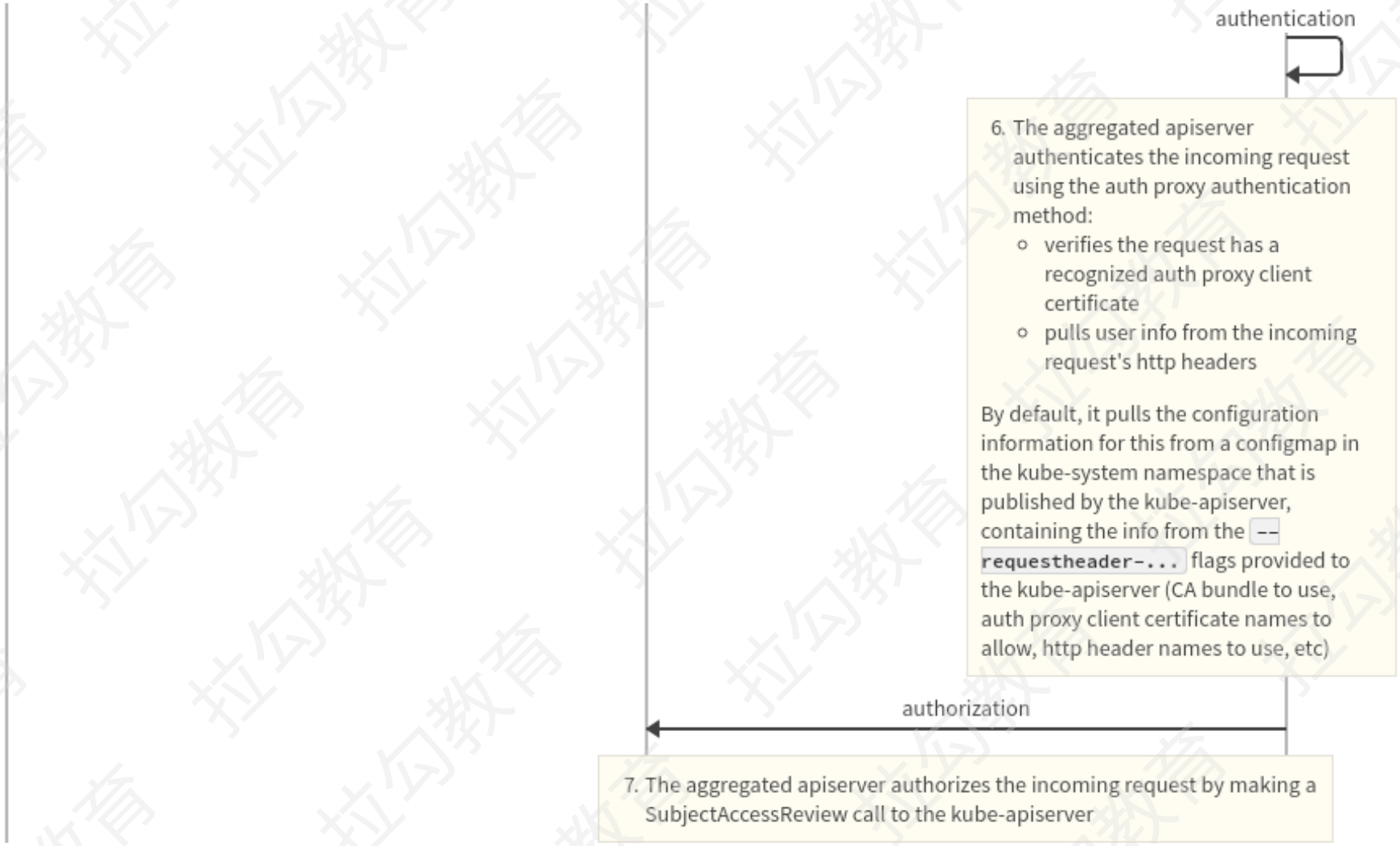
聚合 API



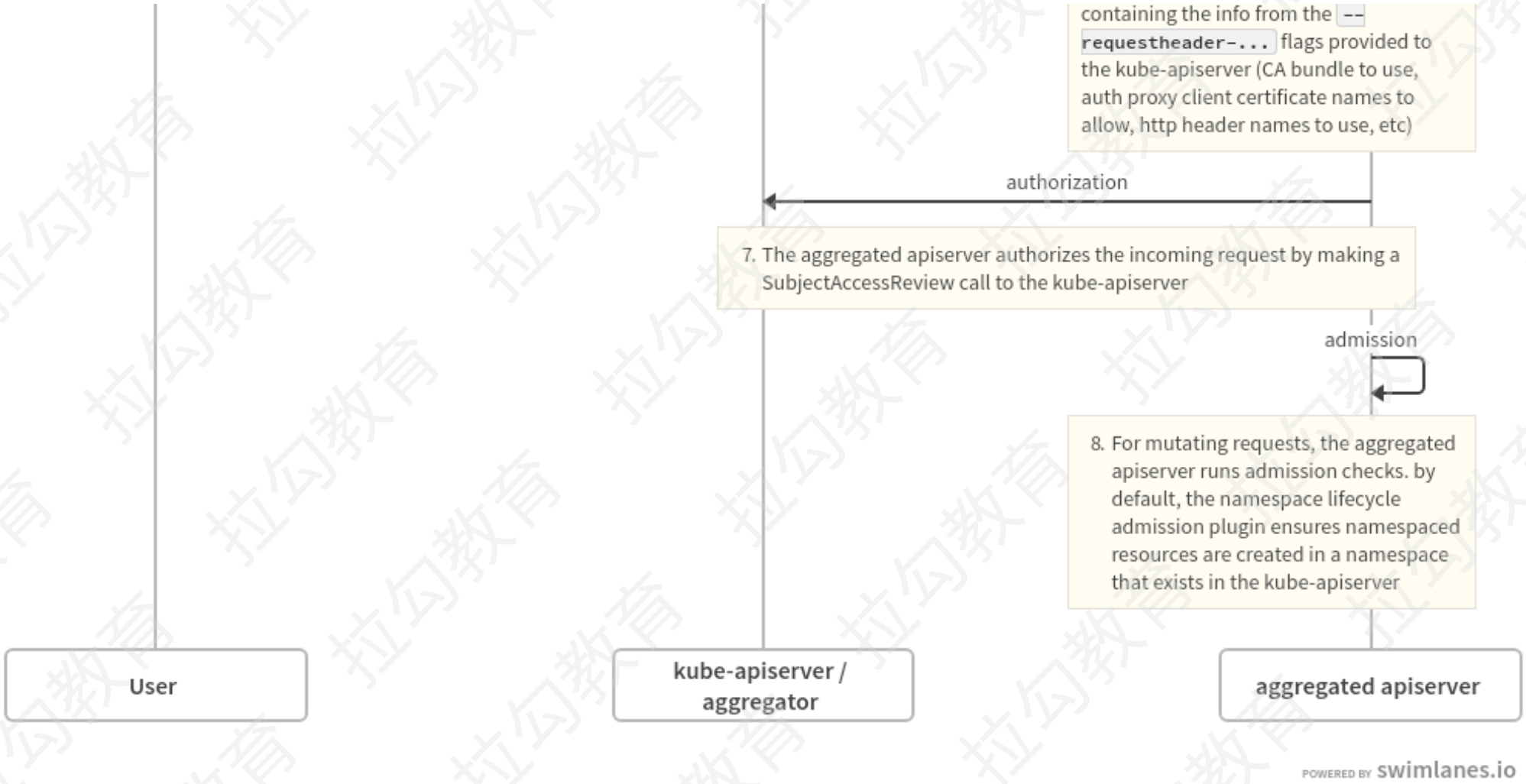
聚合 API


authorization
3. The Kube API server authorizes the requested URL using any configured authorization method (e.g. RBAC)
4. The aggregator opens a connection to the aggregated API server using --proxy-client-cert-file / --proxy-client-key-file client certificate/key to secure the channel
5. The aggregator sends the user info from step 1 to the aggregated API server as http headers, as defined by the following flags:
--requestheader-username-headers
--requestheader-group-headers
--requestheader-extra-headers-prefix
authentication
6. The aggregated apiserver authenticates the incoming request using the auth proxy authentication method:
verifies the request has a recognized auth proxy client certificate

聚合 API



聚合 API



聚合 API


```yaml
apiVersion: apiregistration.k8s.io/v1
kind: APIService
metadata:
  name: v1alpha1.wardle.example.com #对象名称
spec:
  insecureSkipTLSVerify: true
  group: wardle.example.com #扩展 Apiserver 的 API group 名称
  groupPriorityMinimum: 1000 # APIService 对对应 group 的优先级
  versionPriority: 15 # 优先考虑 version 在 group 中的排序
  service:
    name: myapi #扩展 Apiserver 服务的 name
    namespace: wardle #扩展 Apiserver 服务的 namespace
  version: v1alpha1 #扩展 Apiserver 的 API version
```

聚合 API


```yaml
apiVersion: apiregistration.k8s.io/v1
kind: APIService
metadata:
  name: v1alpha1.wardle.example.com #对象名称
spec:
  insecureSkipTLSVerify: true
  group: wardle.example.com #扩展 Apiserver 的 API group 名称
  groupPriorityMinimum: 1000 # APIService 对对应 group 的优先级
  versionPriority: 15 # 优先考虑 version 在 group 中的排序
  service:
    name: myapi #扩展 Apiserver 服务的 name
    namespace: wardle #扩展 Apiserver 服务的 namespace
  version: v1alpha1 #扩展 Apiserver 的 API version
```

CRD


```yaml
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
 #名字必须与下面的 spec 字段匹配，并且格式为'<名称的复数形式>.<组名>'
 name: crontabs.stable.example.com
spec:
 #组名称，用于 REST API: /apis/<组>/<版本>
 group: stable.example.com
 #列举此 CustomResourceDefinition 所支持的版本
 versions:
  - name: v1
   #每个版本都可以通过 served 标志来独立启用或禁止
   served: true
   #其中一个且只有一个版本必需被标记为存储版本
   storage: true
   schema:
    openAPIV3Schema:
     type: object
     properties:
```

CRD


```
    openAPIV3Schema:
      type: object
      properties:
        spec:
          type: object
          properties:
            cronSpec:
              type: string
            image:
              type: string
            replicas:
              type: integer
  #可以是 Namespaced 或 Cluster
  scope: Namespaced
  names:
    #名称的复数形式，用于 URL：/apis/<组>/<版本>/<名称的复数形式>
    plural: crontabs
    #名称的单数形式，作为命令行使用时和显示时的别名
    singular: crontab
```

CRD


```
        properties:
          cronSpec:
            type: string
          image:
            type: string
          replicas:
            type: integer
  #可以是 Namespaced 或 Cluster
  scope: Namespaced
  names:
    #名称的复数形式，用于 URL：/apis/<组>/<版本>/<名称的复数形式>
    plural: crontabs
    #名称的单数形式，作为命令行使用时和显示时的别名
    singular: crontab
    # kind 通常是单数形式的驼峰编码（CamelCased）形式。你的资源清单会使用这一形式。
    kind: CronTab
    # shortNames 允许你在命令行使用较短的字符串来匹配资源
    shortNames:
    - ct
```

CRD


```
apiVersion: "stable.example.com/v1"
kind: CronTab
metadata:
 name: my-new-cron-object
spec:
 cronSpec: "* * * * */5"
 image: my-awesome-cron-image
```

CRD


```
kubectl get crontab
NAME                 AGE
my-new-cron-object   6s
```

CRD


```yaml
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: crontabs.stable.example.com
spec:
  group: stable.example.com
  versions:
    - name: v1
      served: true
      storage: true
      schema:
        openAPIV3Schema:
          type: object
          properties:
            spec:
              type: object
```

CRD


```
      type: object
      properties:
        cronSpec:
          type: string
        image:
          type: string
        replicas:
          type: integer
    status:
      type: object
      properties:
        replicas:
          type: integer
        labelSelector:
          type: string
  # subresources 描述定制资源的子资源
```

CRD


```
    status: {}
    # scale 启用 scale 子资源
    scale:
      # specReplicasPath 定义定制资源中对应 scale.spec.replicas 的 JSON 路径
      specReplicasPath: .spec.replicas
      # statusReplicasPath 定义定制资源中对应 scale.status.replicas 的 JSON 路径
      statusReplicasPath: .status.replicas
      # labelSelectorPath 定义定制资源中对应 scale.status.selector 的 JSON 路径
      labelSelectorPath: .status.labelSelector
  scope: Namespaced
  names:
    plural: crontabs
    singular: crontab
    kind: CronTab
    shortNames:
    - ct
```

写在最后

参考**文档（见文章末尾）**

可以通过**网址（见文章末尾）**中的内容来比较 CRD 和 聚合 API 的功能差异

CRD 更为易用，而聚合 API 更为灵活

Next：《28 | 面向 K8s 编程：如何通过 Operator 扩展 Kubernetes API？》

拉勾教育

—互联网人实战大学—

关注拉勾「教育公众号」
获取更多课程信息

L / A / G / O / U