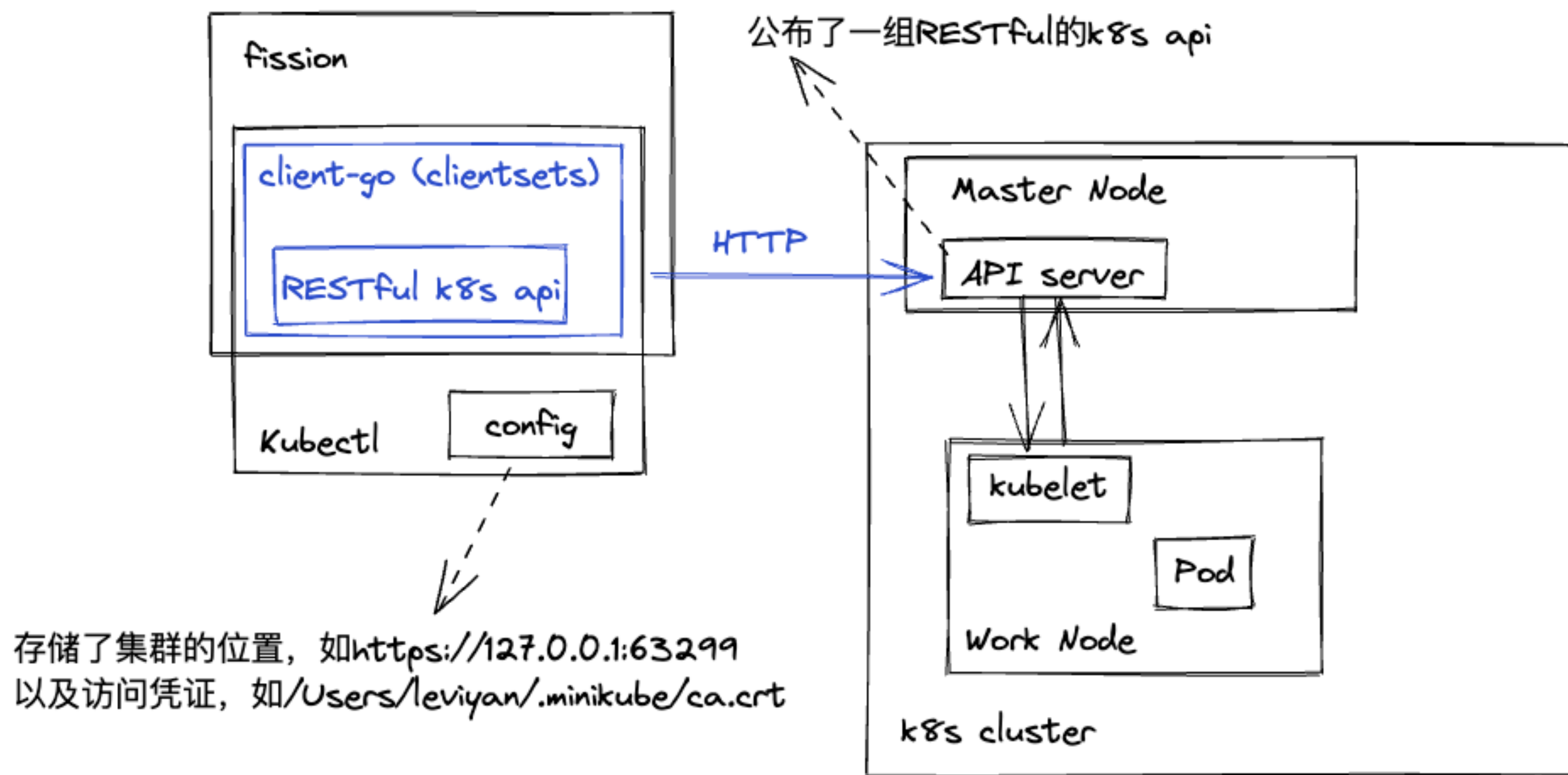
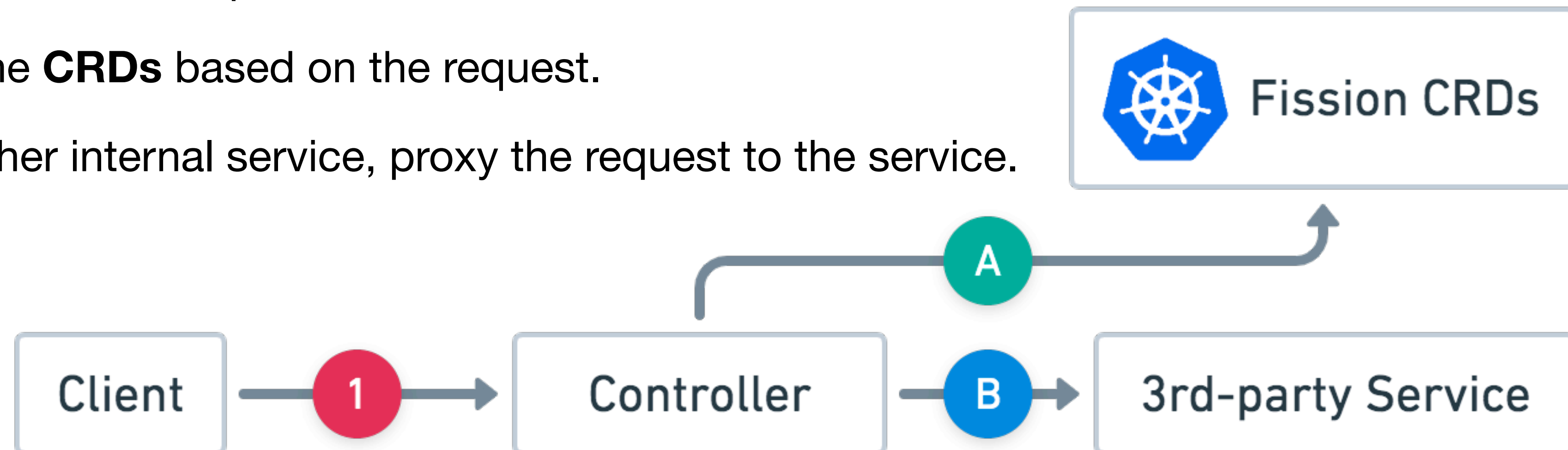


上次的内容的直观理解



fission

- controller
 - Accept **REST API requests** and **create Fission resources**
 - All fission resources are stored in [Kubernetes CRDs](#).
- 创建过程
 - The clients send requests to the endpoints on Controller.
 - (A) Controller operates the **CRDs** based on the request.
 - (B) If a request is to another internal service, proxy the request to the service.



运行fission

使用newdeploy运行

- fission env create --name python --image xxx

```
~/Doc/1/202205/fission fission fn create --name hello --env python --code hello.py --executortype newdeploy --minscale 1 --maxscale 3 --targetcpu 50  
Package 'hello-d1b97f2f-03f9-44d4-b72b-8b9e25f8731a' created  
● function 'hello' created
```

k8s

data structure

- ObjectMeta: metadata that **all persisted resources** must have, which includes all objects users must create.
 - Name String
 - Namespace string
 - UID types.UID: UID is the unique in time and space value for this object.
 - It is typically generated by the server on **successful creation of a resource** and is not allowed to change on PUT operations
 - ClusterName string: The name of the cluster which the object belongs to

fission

data structure

- **Package:** Think of these as **function-level images**
 - **PackageSpec:** includes **source/deploy archives** and the reference of **environment to build the package**
 - **Environment:** a reference to the environment for **building source archive**
 - **Source Archive:** Source is the archive contains **source code and dependencies file**.
 - **Literal** byte[]: **Literal contents** of the package, Can be used for **encoding packages**
 - **URL** String: URL references a package
 - **Deployment Archive:** the **deployable archive** that environment runtime **used to run user function**
 - Since it is a deployment archive, there is **no need to build it BuildCommand**(a custom build command that builder used to build the source archive)
 - --buildCommand xxx
 - 使用 --code xxx.py命令, 会创建deployment archive
 - **PackageStatus:** indicates **the build status of package**.
 - BuildStatus String: the package build status

fission

data structure

- Function
 - TypeMeta
 - Kind
 - APIVersion
 - ObjectMeta
 - FunctionSpec: describes the contents of the function
 - Environment EnvironmentReference: is the **build and runtime environment** that this function is associated with
 - namespace + name
 - **Package** FunctionPackageRef: Reference to a **package** containing deployment and optionally the source
 - PackageRef(创建好资源后的metadata) + functionName
 - **Resources** apiv1.ResourceRequirement: cpu and memory resources as per K8S standards
 - 用来限制资源使用率
 - **InvokeStrategy** InvokeStrategy
 - 指定executionStrategy
 - PodSpec *apiv1.PodSpec: specifies podspec to use for executor type container based functions
 - **Volumes** []Volume: List of volumes that can be mounted by containers belonging to the pod

fission

data structure

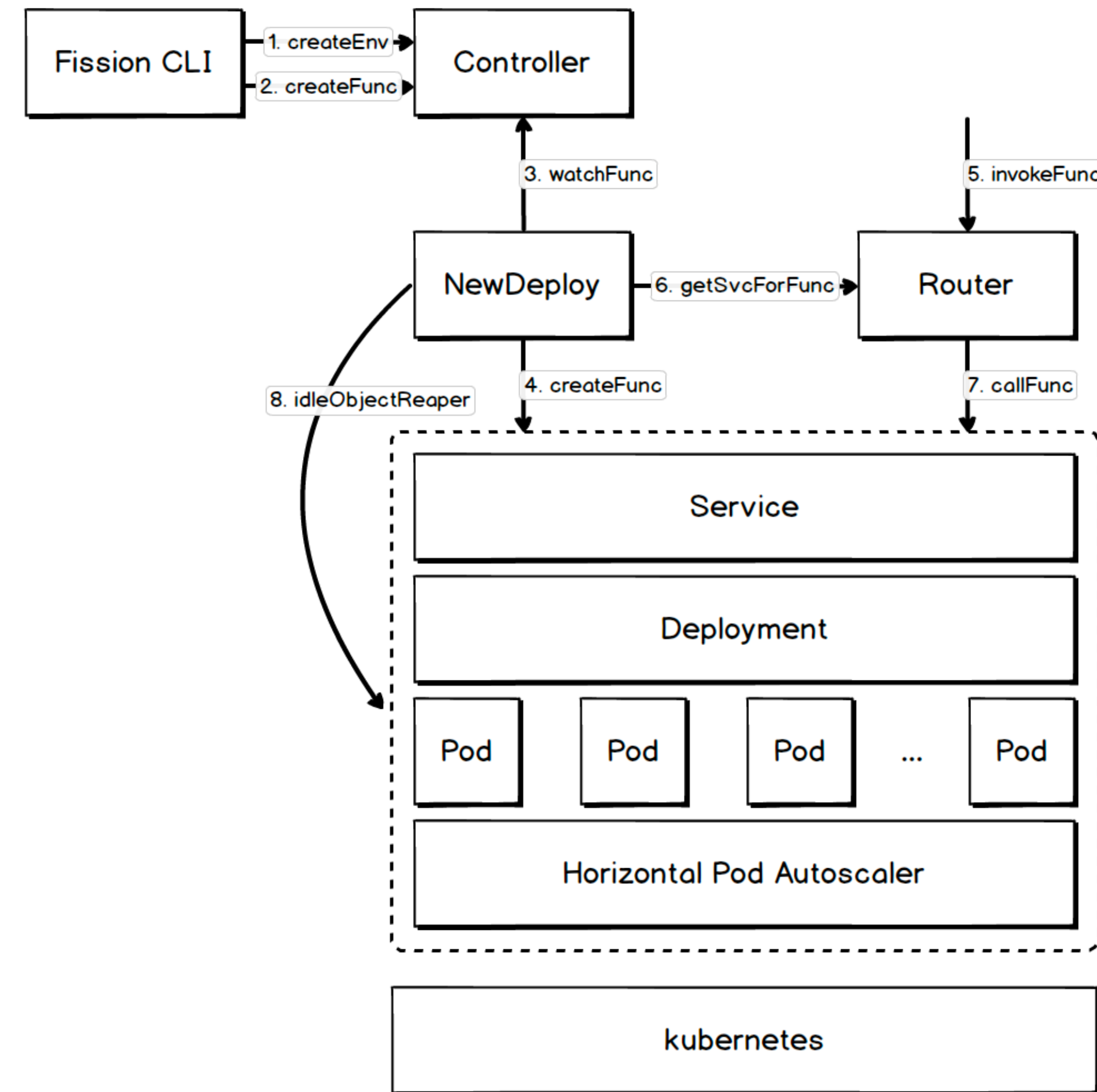
- **CreateSubCommand**
 - cmd.CommandActioner
 - function *fv1.Function
 - specFile string

fission运行流程

创建package、function

```
~/Doc/1/202205/fission fission fn create --name hello --env python --code hello.py --executortype newdeploy --minscale 1 --maxscale 3 --targetcpu 50
Package 'hello-d1b97f2f-03f9-44d4-b72b-8b9e25f8731a' created
function 'hello' created
```

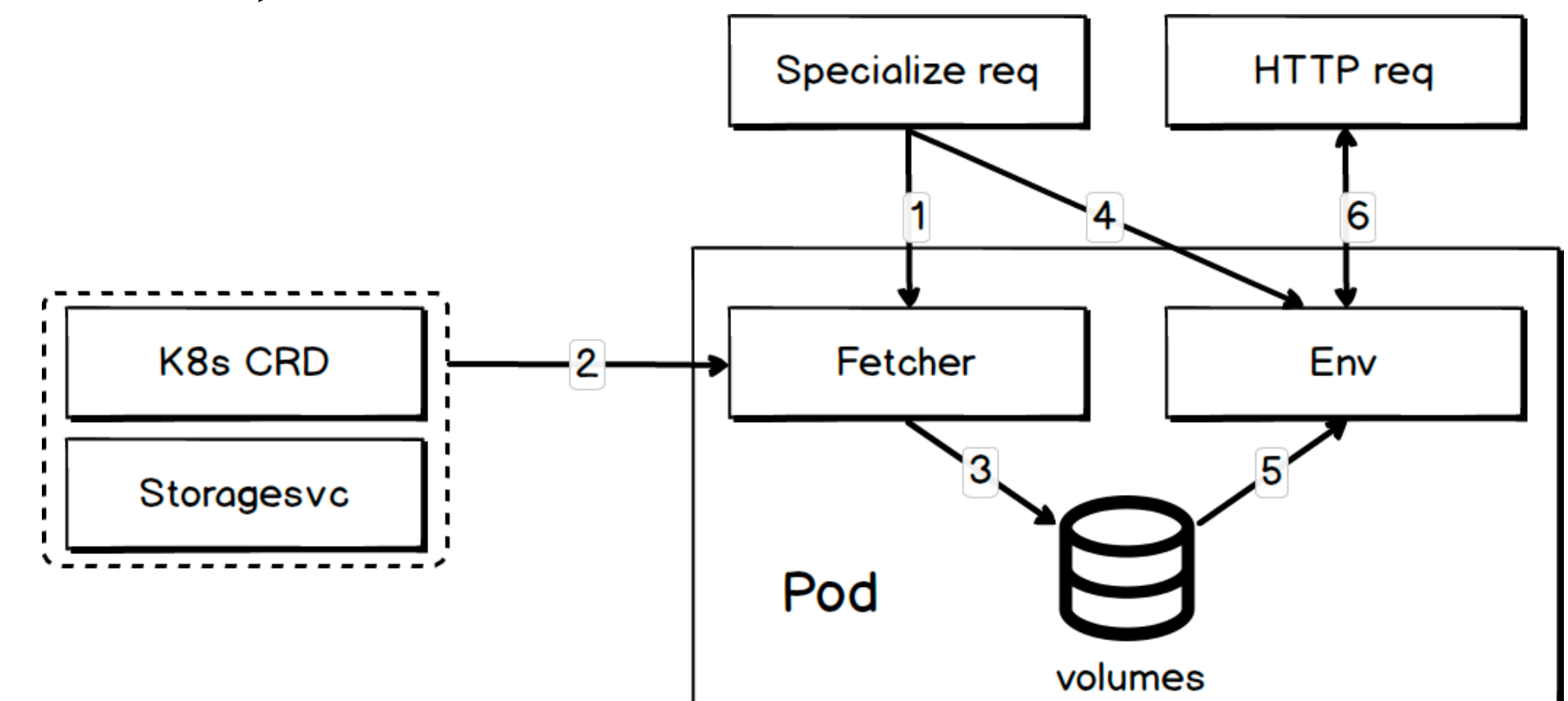
- <https://processon.com/diagraming/62f0c5096376896d38772e90>
- 问题
 - HTTP请求发给谁?
 - 如果发送给executor?
 - 发送给executor, 然后executor发送给k8s创建? 可能就和上次的AddFunc连接起来;
 - ...
 - 编译在哪里进行?
 - 可能并不用编译成image(如python), package就是一种image;
 - 可能编译过程在environment进行;



fission

environment

- env: 用户函数运行的载体；当它成功加载共享volume 中的函数后，便可以接受用户请求
- Since Fission invokes Functions through HTTP, this means **the runtime of an environment is a container with an HTTP server**, and usually **a dynamic loader** that can load a function.
- Some environments also contain **builder containers**, which take care of **compilation and gathering dependencies**.



参考

- Controller
- fission concept