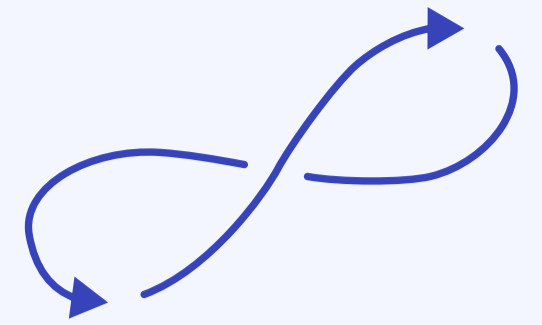# How does the Kubernetes API work?
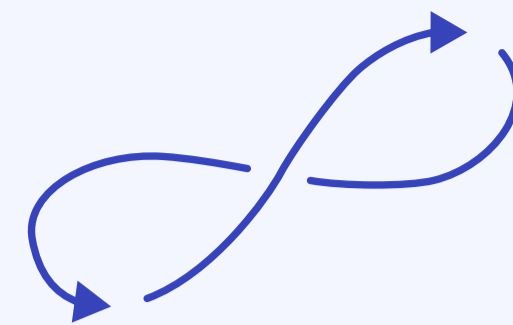
The Kubernetes API is made of several smaller components.

In this post you will learn about the component involved in storing resources into etcd.

gart.

When you type `kubectl apply -f my.yaml` your YAML is sent to the API and stored in etcd. But what is the API doing?
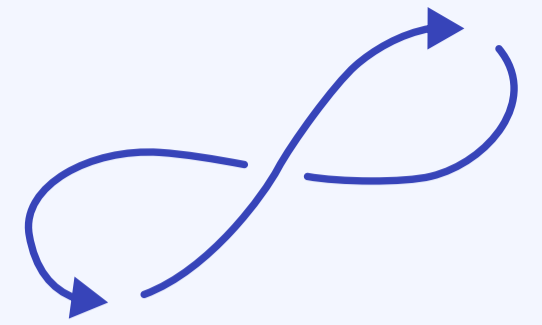
`kubectl apply -f my.yaml`

API server

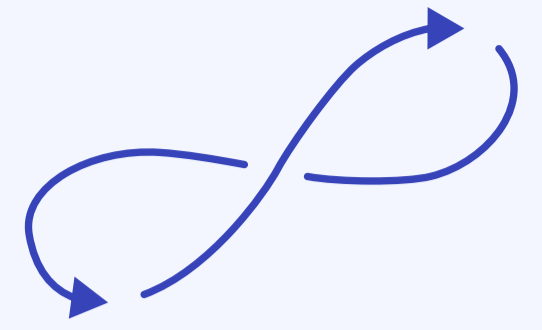API server → API server deployment #1

gart.

The API has a single block in the diagram, but the reality is that several components are involved in processing your request.

The first component in the API is the HTTP handler.

You can think about it a web server ready to receive HTTP requests:
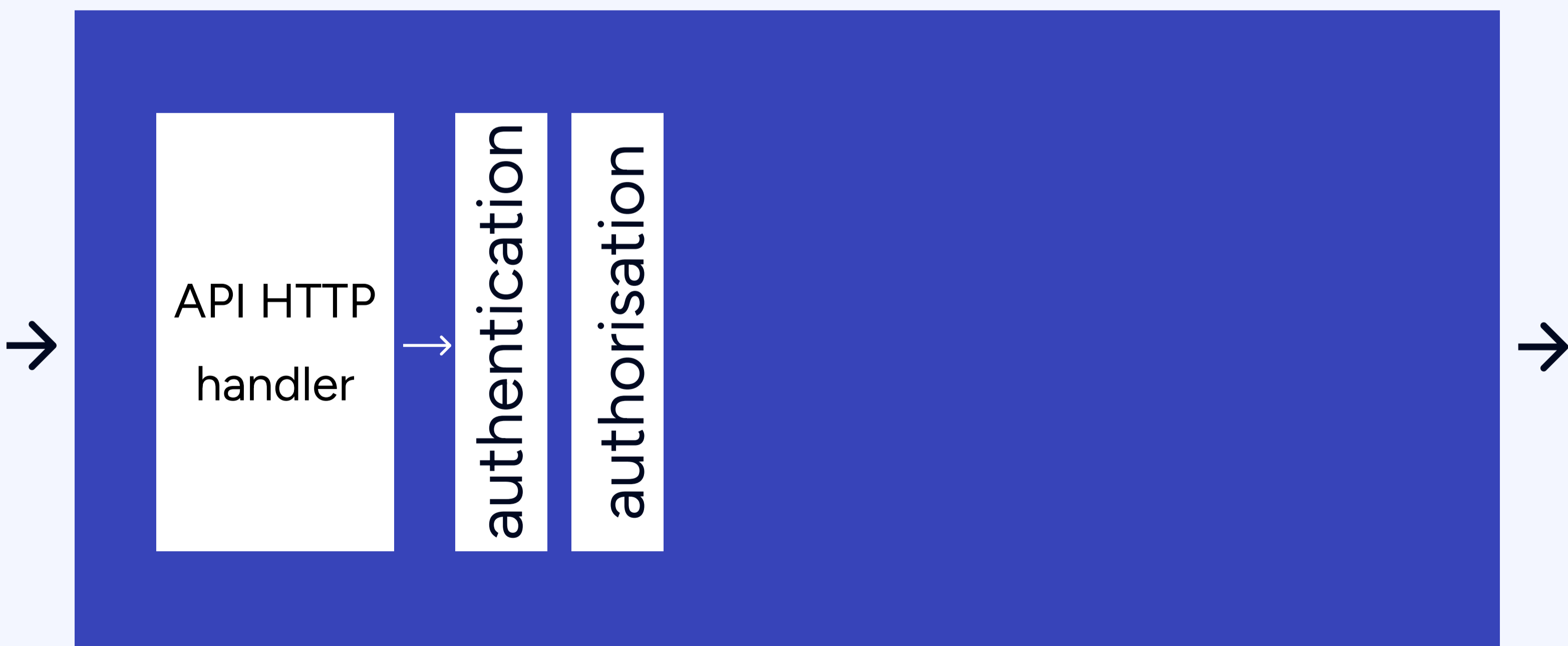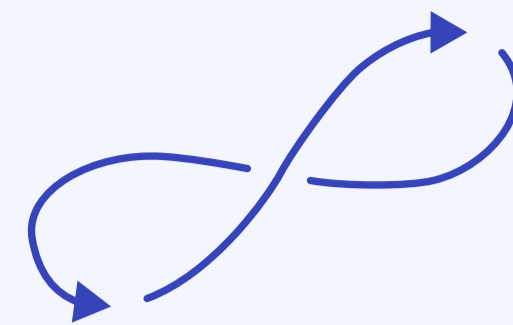
→

API HTTP

handler

→

gart.

In that part, the API has to make sure that:

- You have access to the cluster (authentication)
- You can create, delete, list, etc. resources (authorisation)

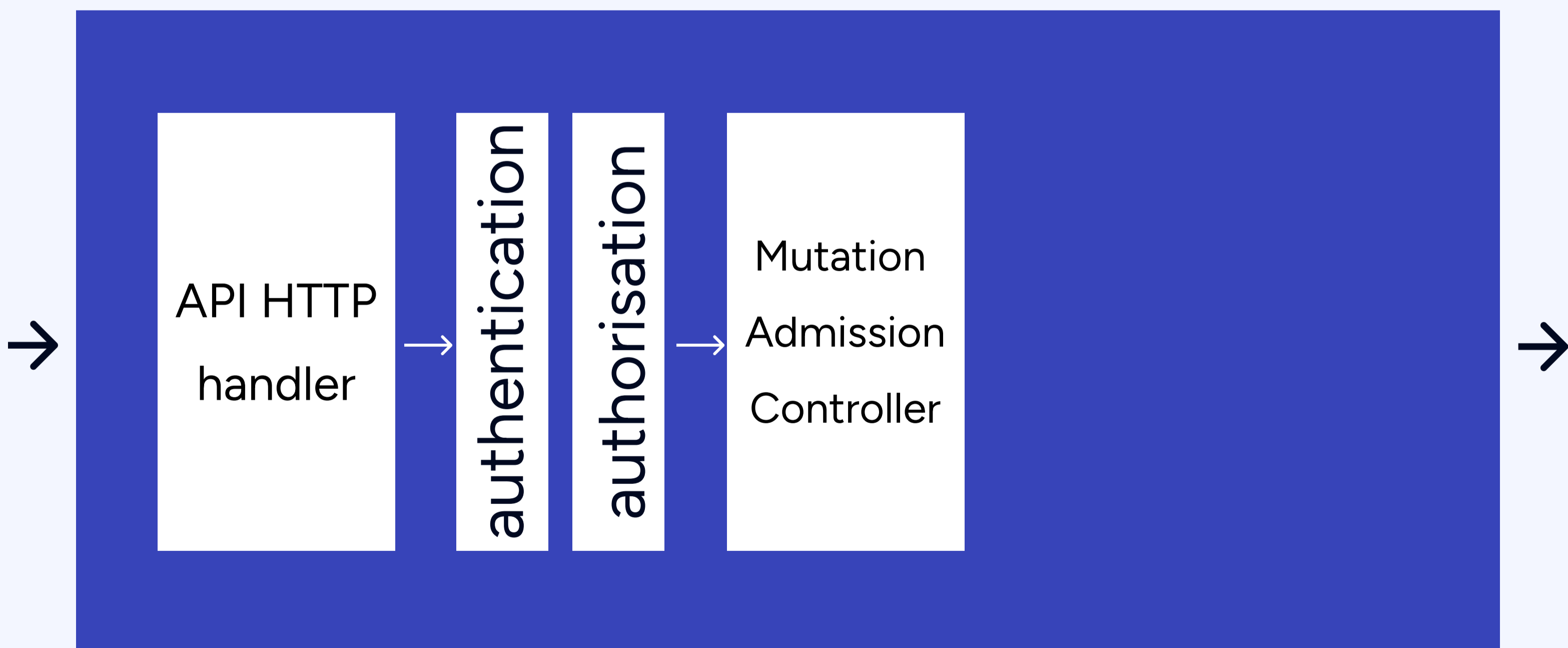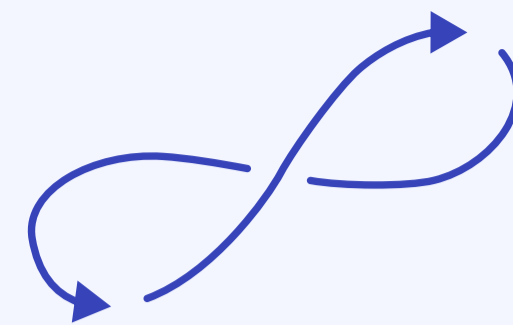This is the part where the RBAC rules are evaluated.

gart.

So you're authenticated and you can create Pods, what's next?

The API passes the request to the Mutation Admission Controller.

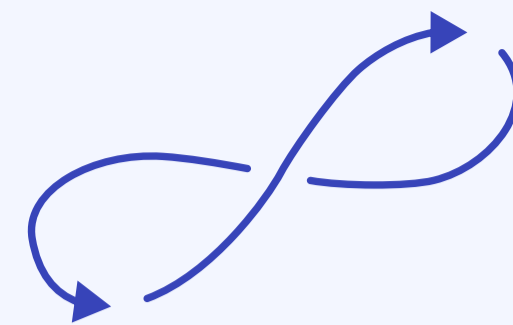This component is in charge of looking at your YAML and modifying it.

The Schema Validation component makes sure that the resource is valid against the internal schema.

You don't want malformed YAML to be stored in the cluster.

→ | API HTTP handler | → | authentication | authorisation | → | Mutation Admission Controller | → | Schema Validation | →
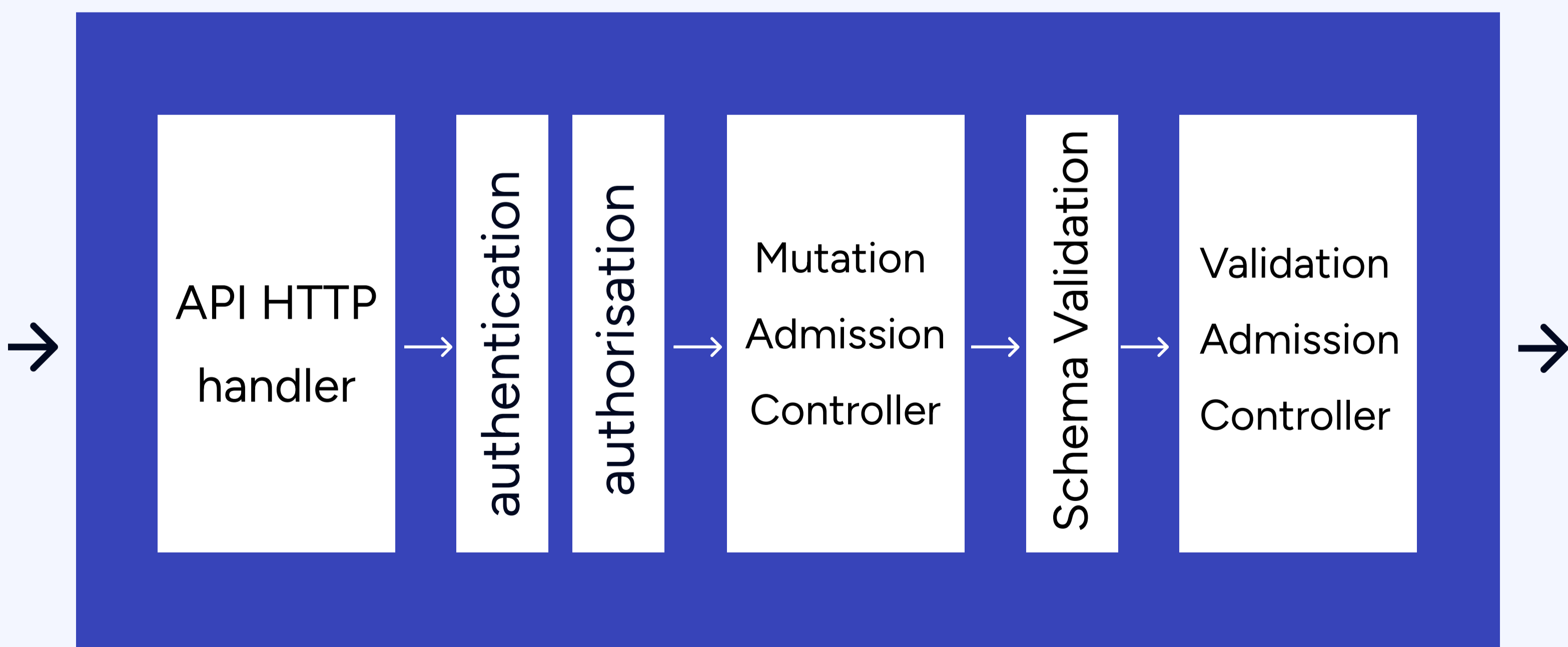
gart.
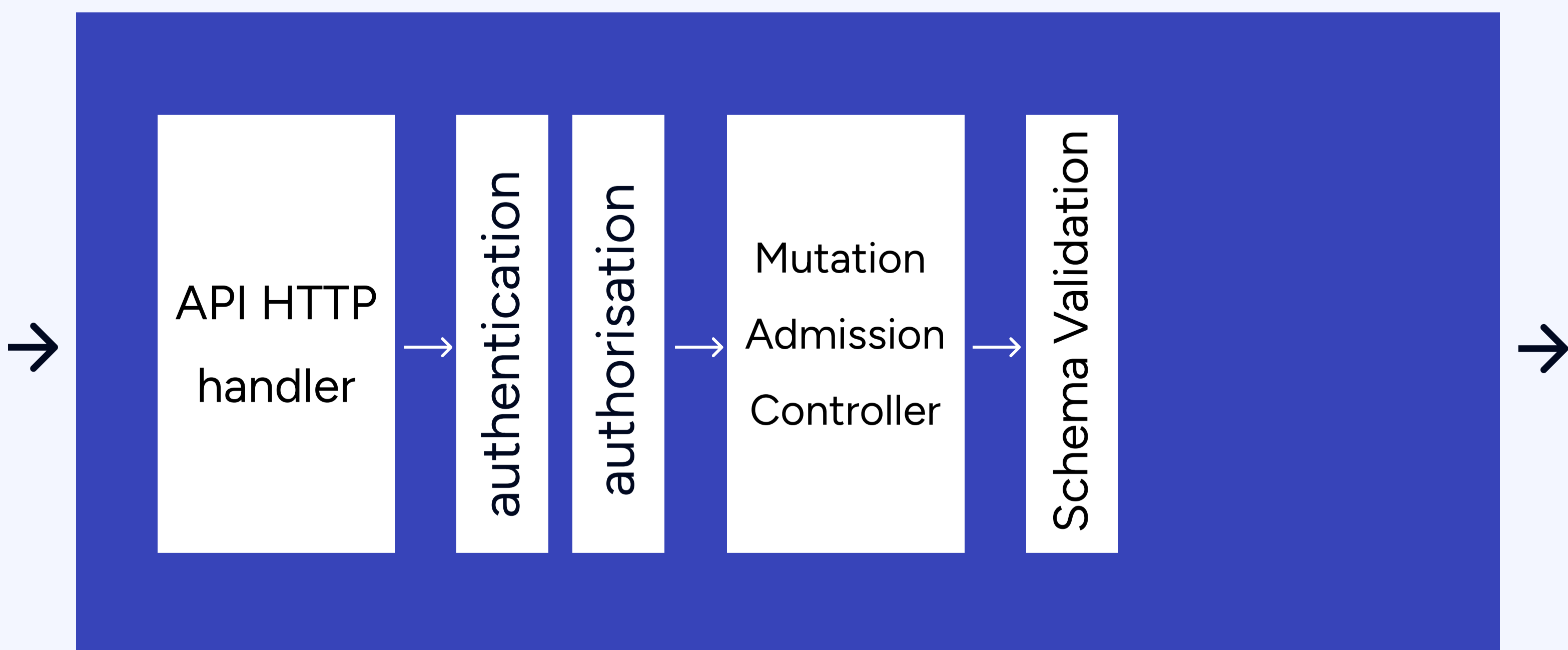
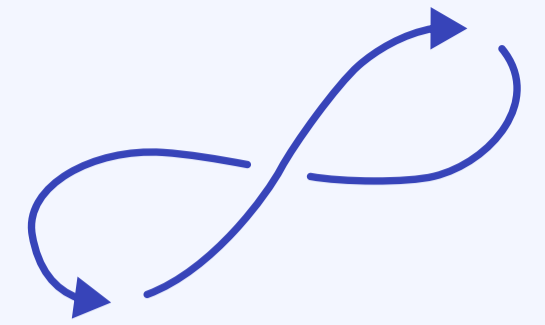If you tried to deploy a Pod in a namespace that doesn't exist, who stops you?

The Validation Admission Controller stops you.

Are you trying to deploy more resources than your Quota?

The controller stops you again.

→ API HTTP handler → authentication → authorisation → Mutation Admission Controller → Schema Validation → Validation Admission Controller →
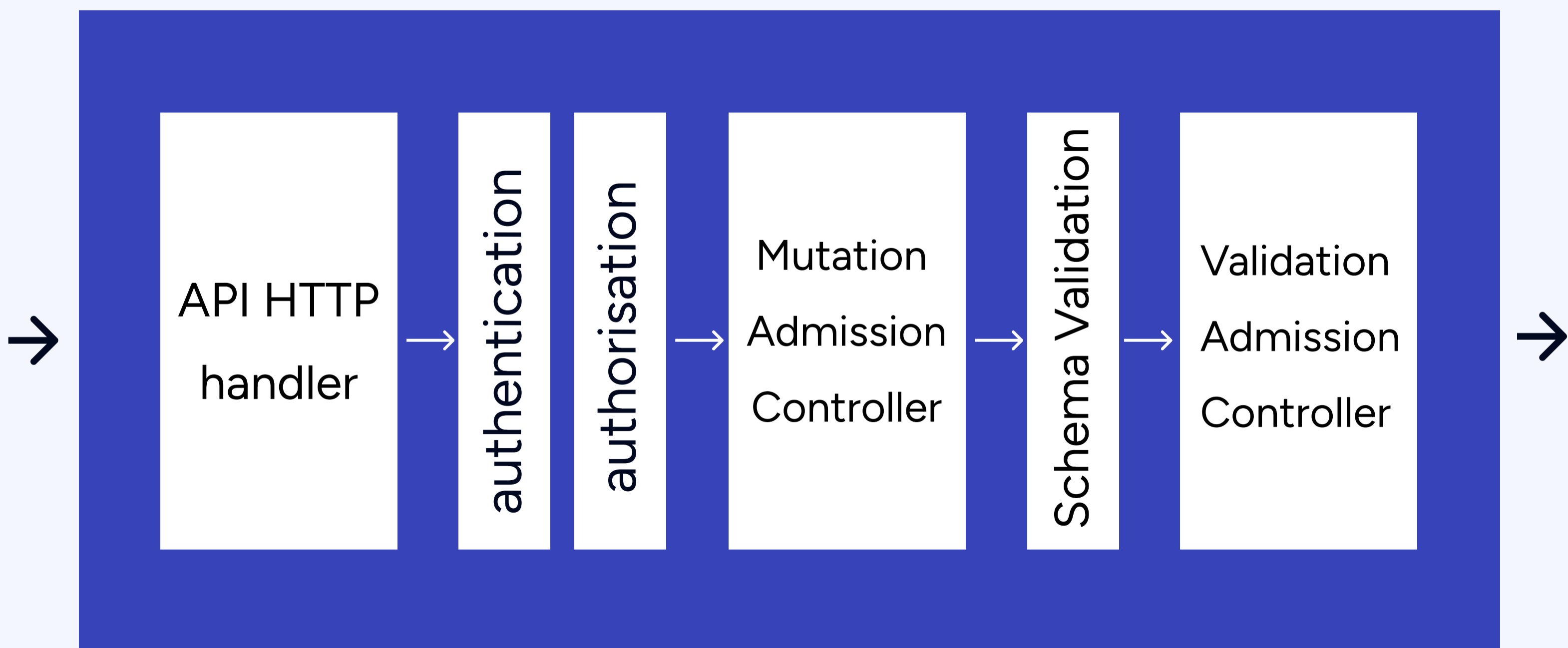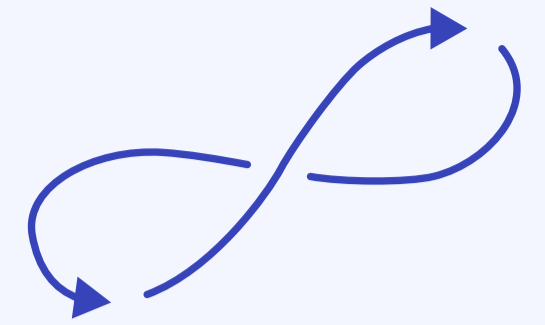
gart.

If you managed to pass the Validation Admission Controller, your resource is safely stored in etcd.
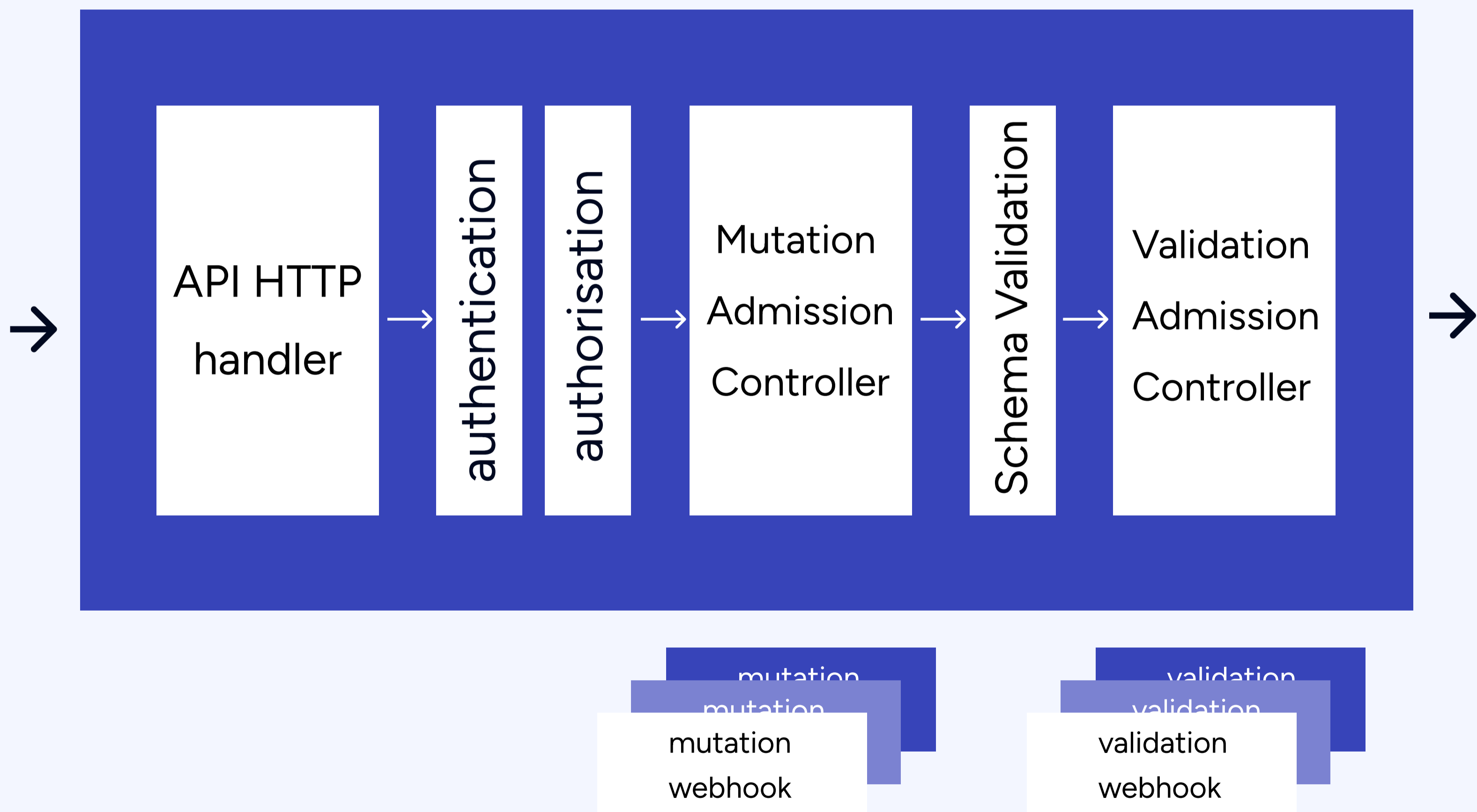
Now that you know about the Mutating Admission controller wouldn't be great if you could design your own?
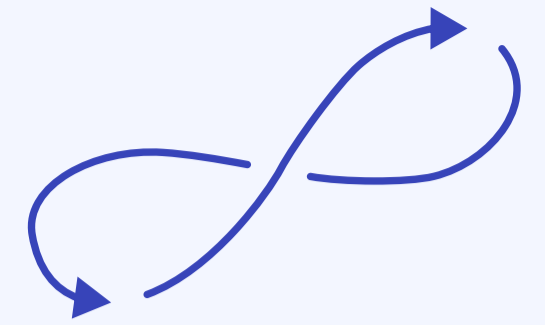
→ | API HTTP handler | → | authentication | authorisation | → | Mutation Admission Controller | → | Schema Validation | → | Validation Admission Controller | →

mutation
mutation
mutation
webhook

gart.

You can register your scripts with the Mutating Admission Controller.

You can also do the same with the Validating Admission Controller.

→ | API HTTP handler | → | authentication | authorisation | → | Mutation Admission Controller | → | Schema Validation | → | Validation Admission Controller | →

mutation
mutation
mutation webhook

validation
validation
validation webhook

gart.

Custom Admission controllers:

- Istio automatically injects an extra container to all Pods (mutation)
- Gatekeeper (Open Policy Agent) checks your resources against policies and reports violations (validation)
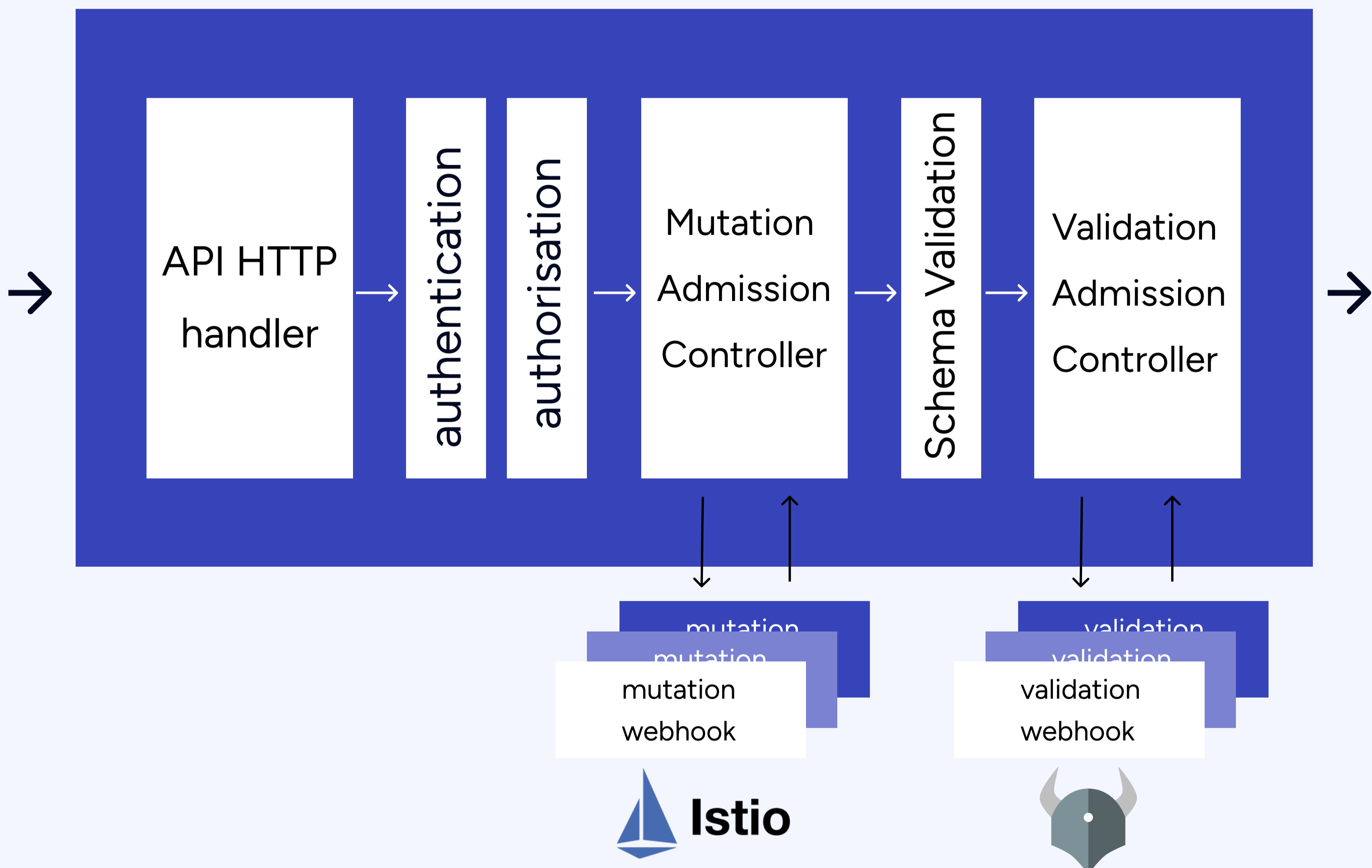
gart.

Like this type of content?

Hit that follow button for more DevOps goodies coming your way!

Roman Burdiuzha

Cloud Architect | Co-Founder & CTO at Gart

gart.