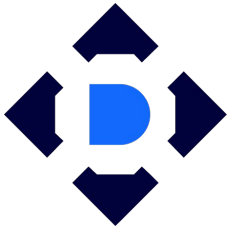
Istio

Request lifecycle

Istio enabled



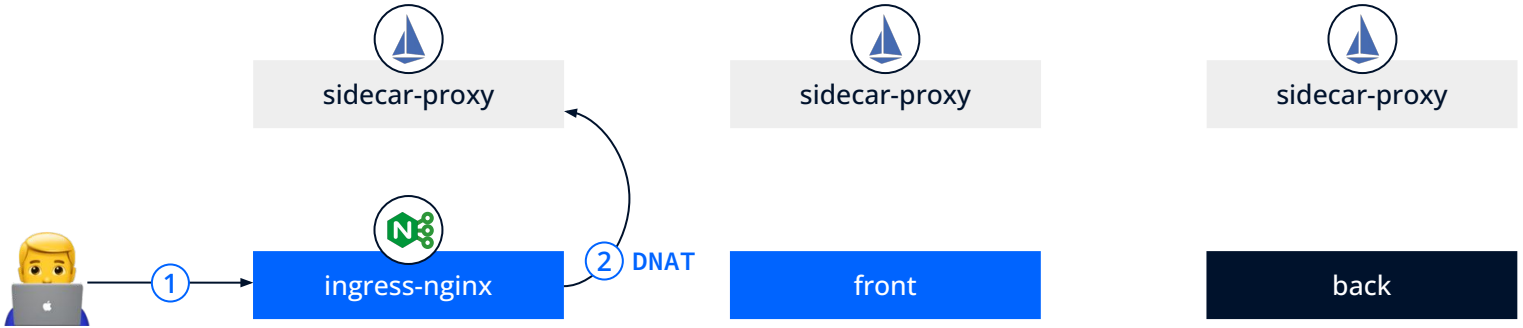
FLANT

Deckhouse

Kubernetes Platform

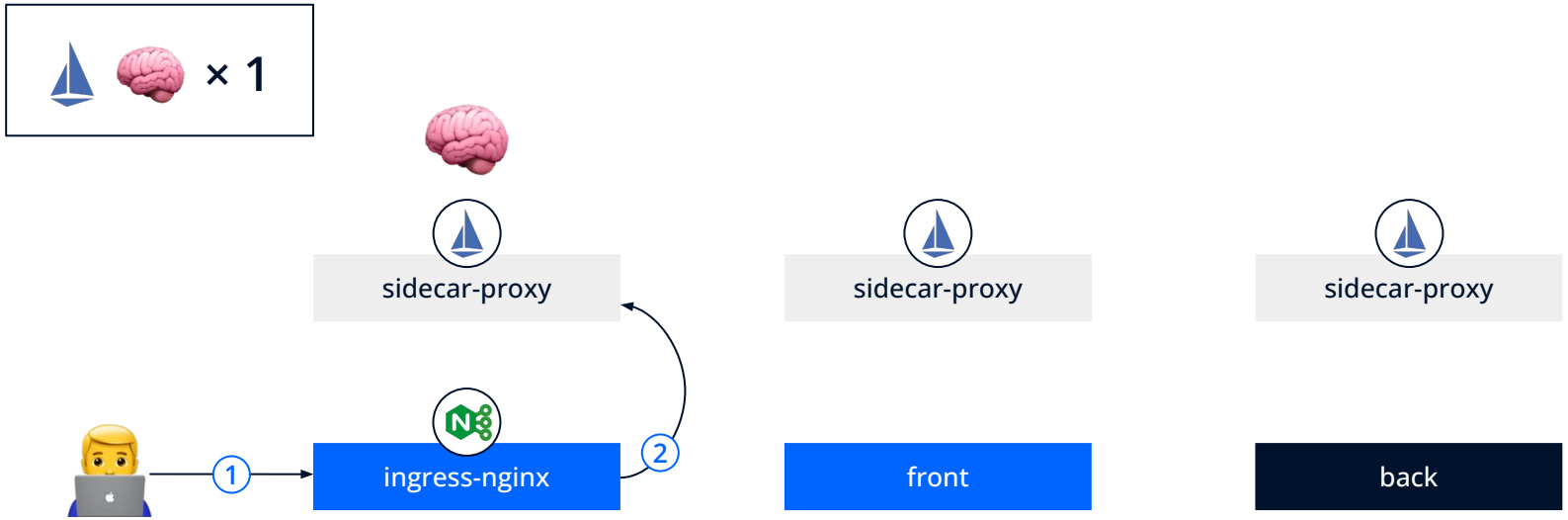
Suppose there is a standard app.example.com application with a frontend and a backend.
However, this time, it is managed by Istio.

1 The user sends a request to app.example.com that end up in the ingress controller.

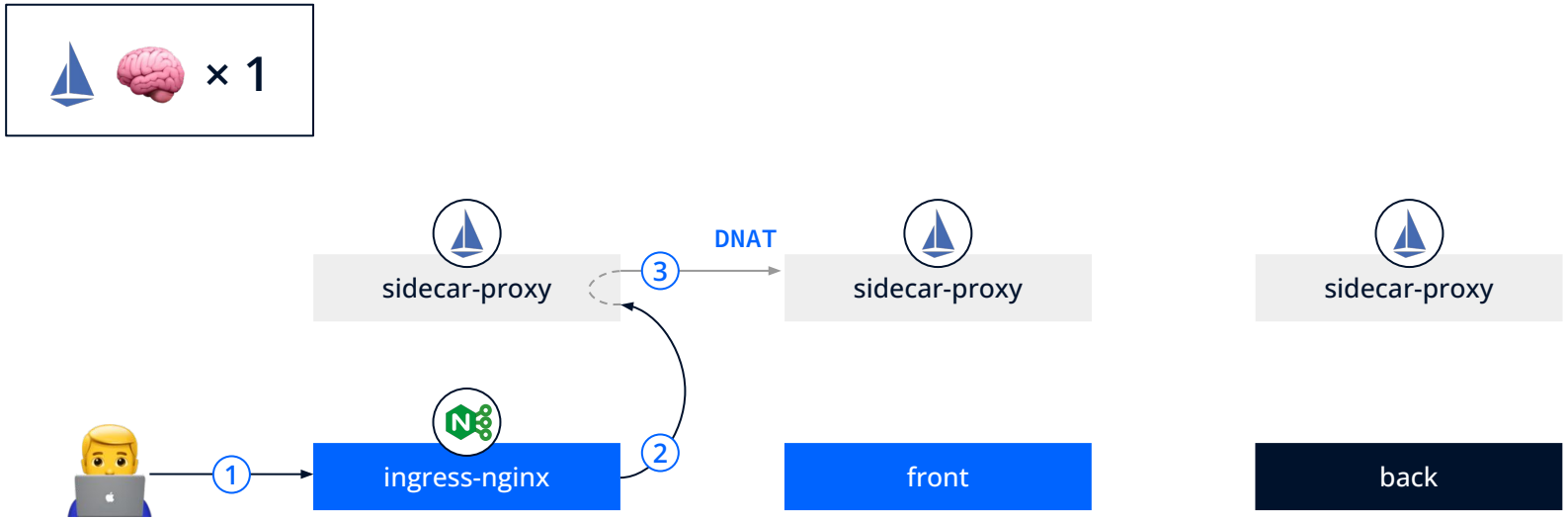


2

The ingress controller then routes the request to the *front* Pod,
but sidecar-proxy intercepts the request thanks to DNAT.

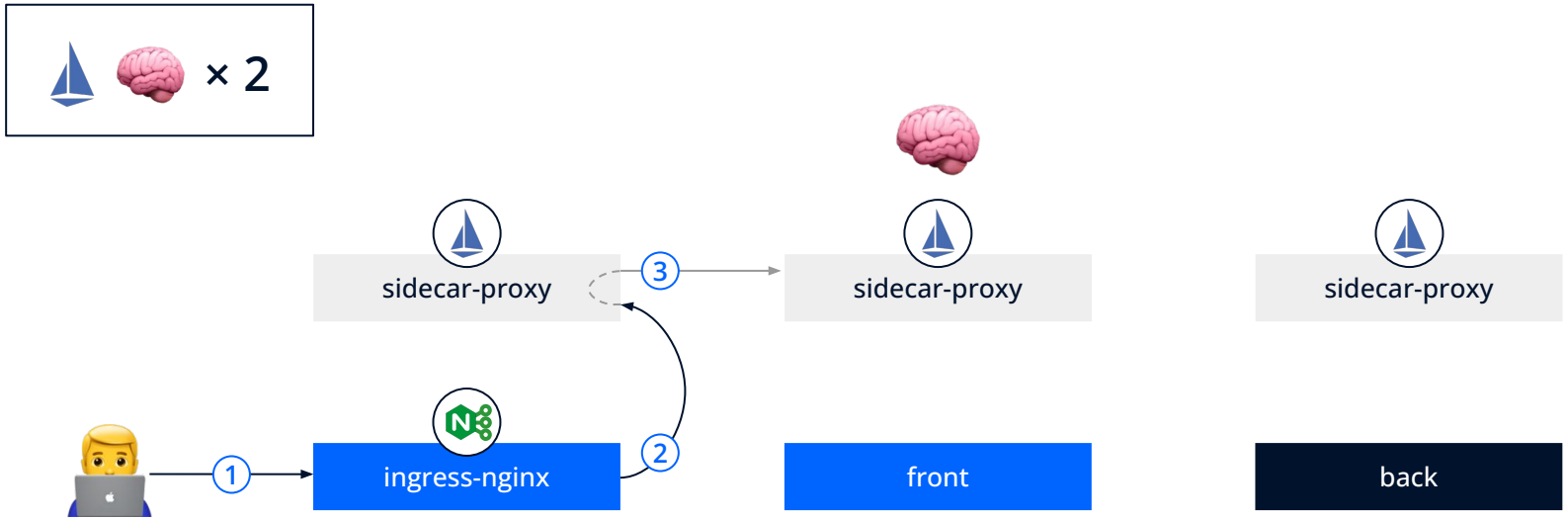


The sidecar-proxy identifies the request and applies the necessary routing, authorization, balancing rules, etc.

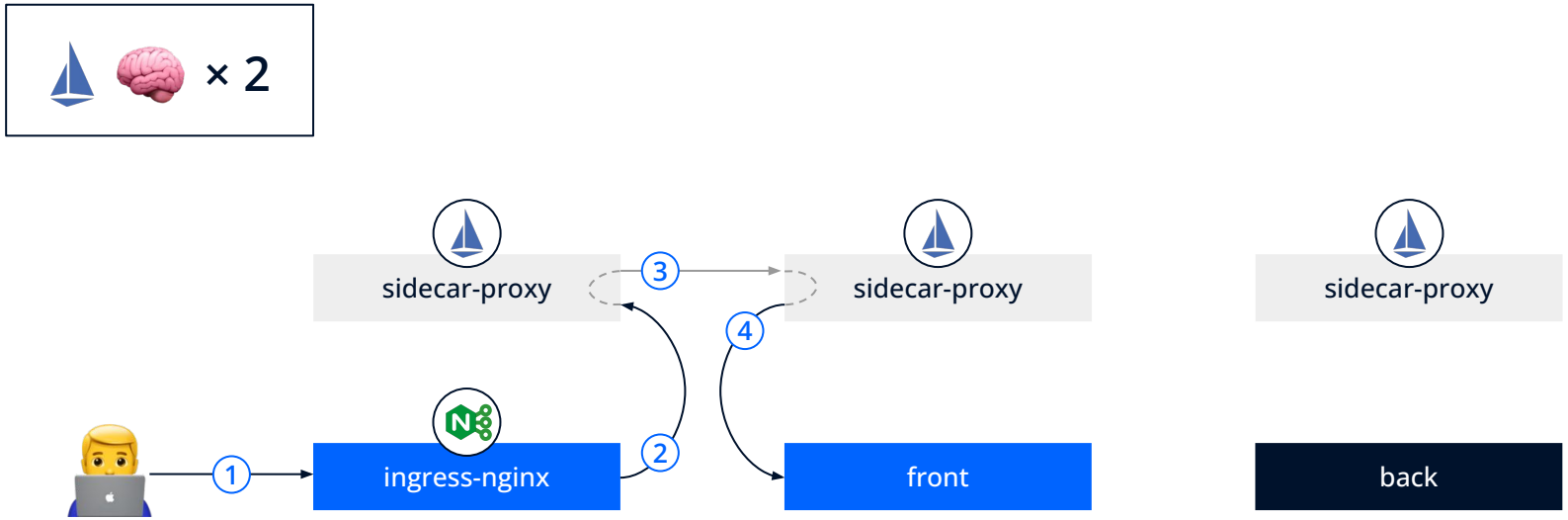


3

The sidecar-proxy proxies the request to the *front* Pod, but the request gets redirected to sidecar-proxy because of a special DNAT rule.

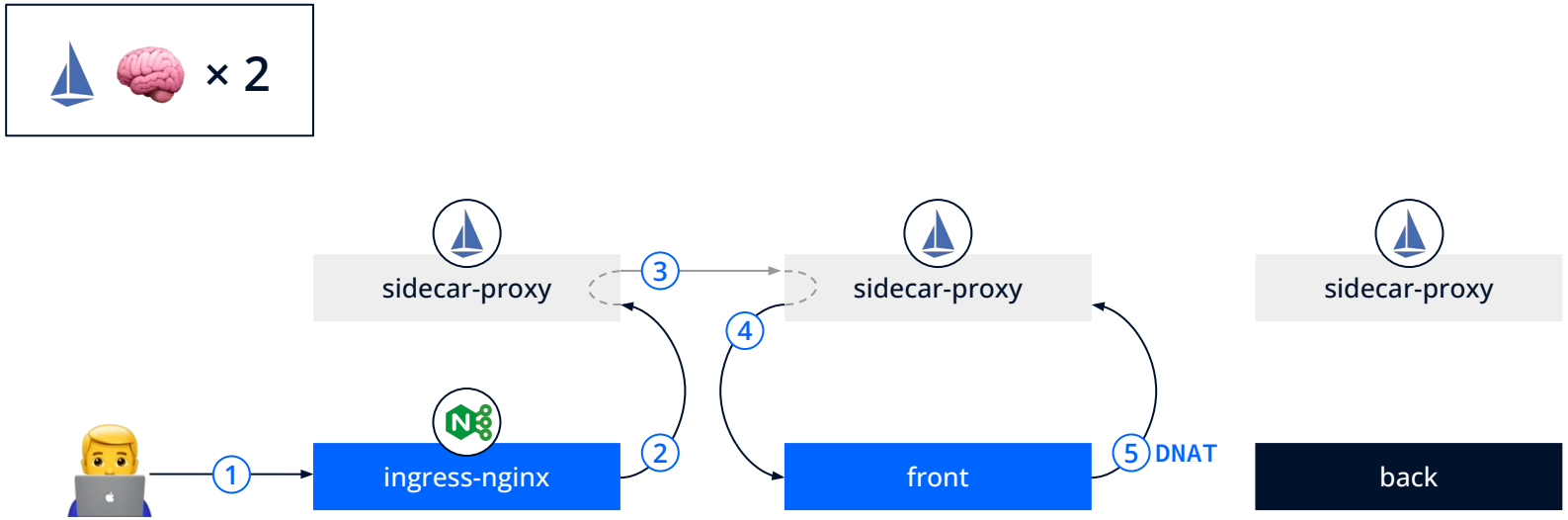


The sidecar-proxy then runs all the necessary validations...



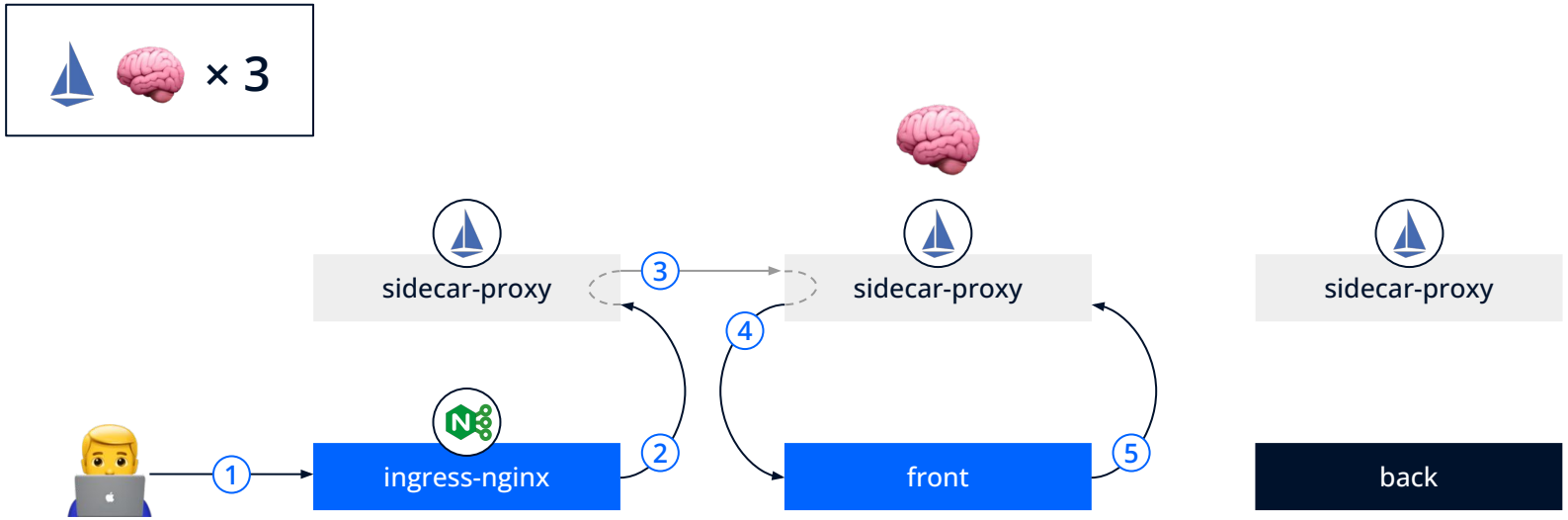
4

...and forwards the request to the *front* Pod.

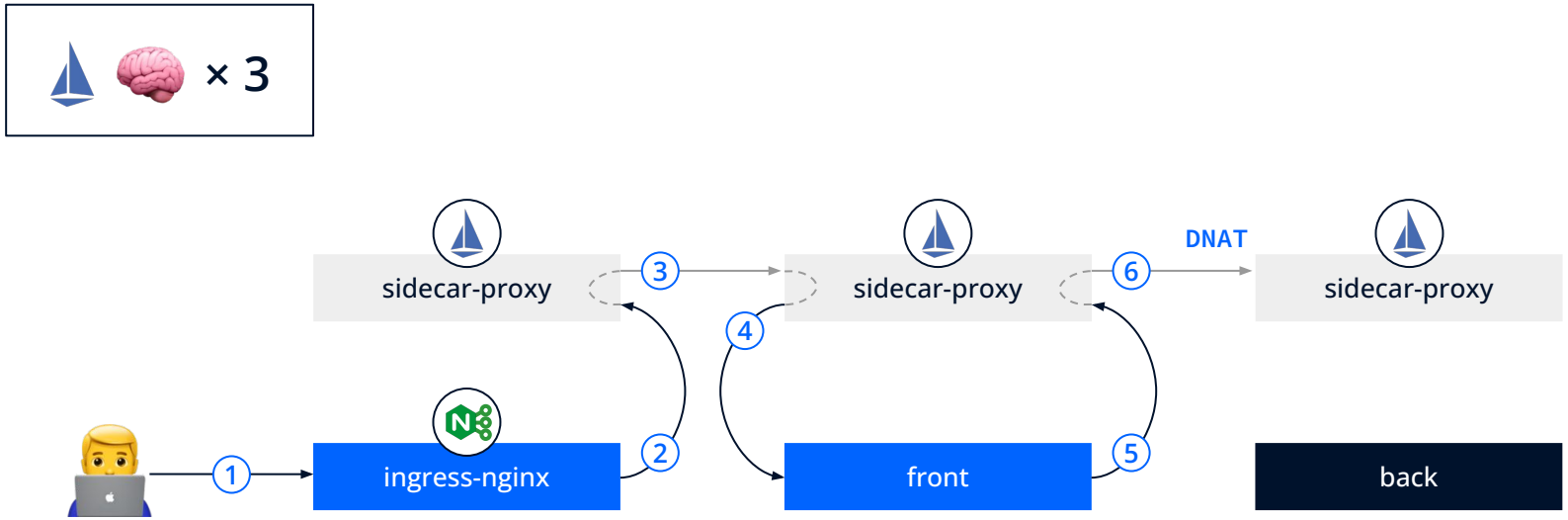


5

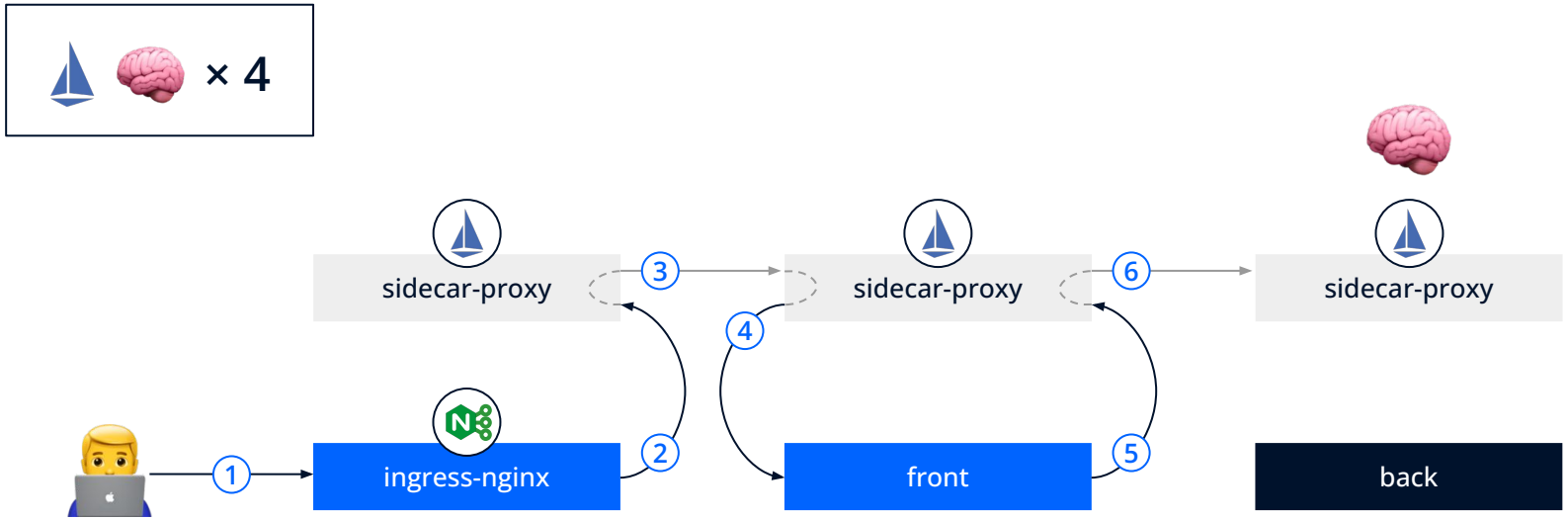
Next, the *front* Pod initiates a secondary request to the *back* Pod.



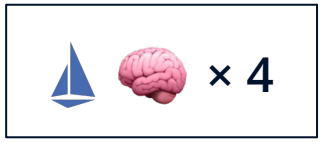
The sidecar-proxy decides what to do with the secondary request...

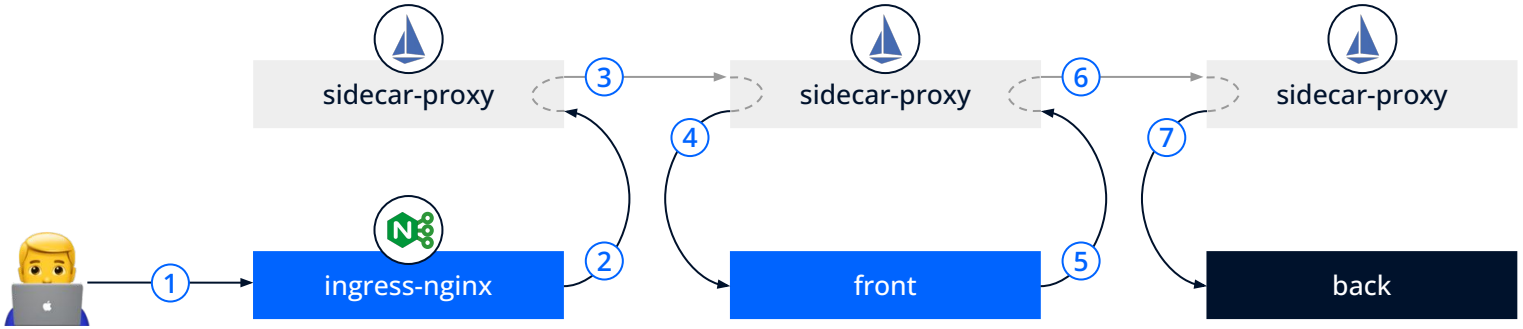


6 ...and sends it to the *back* Pod, where the sidecar-proxy also intercepts it.



The sidecar-proxy performs the necessary validations...





7

...and eventually forwards the request to the *back* Pod.