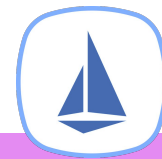


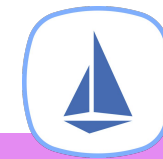
Istio

Request lifecycle

Istio enabled



istio-proxy



istio-proxy



istio-proxy



ingress-nginx

front

back

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

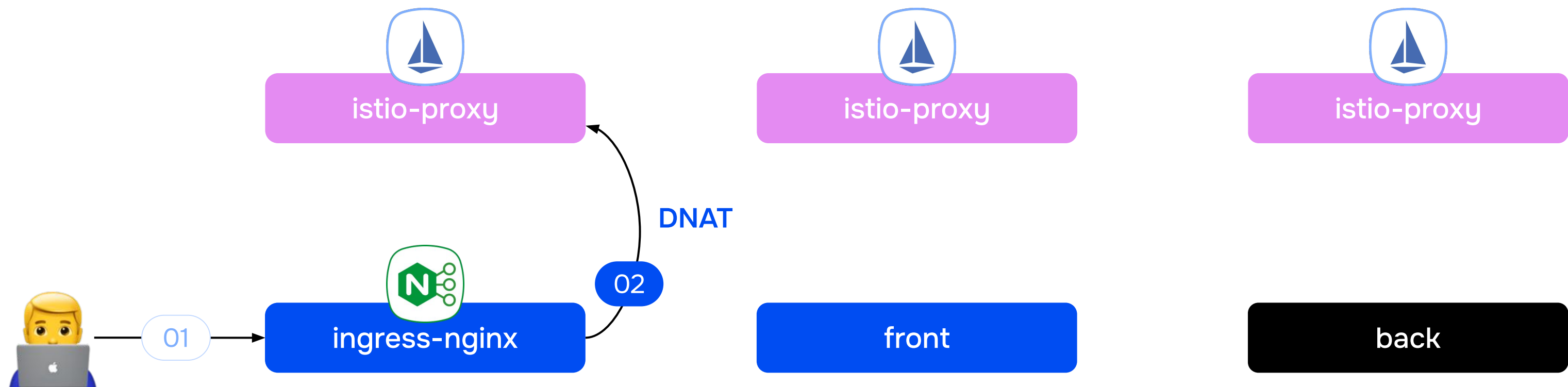




01

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

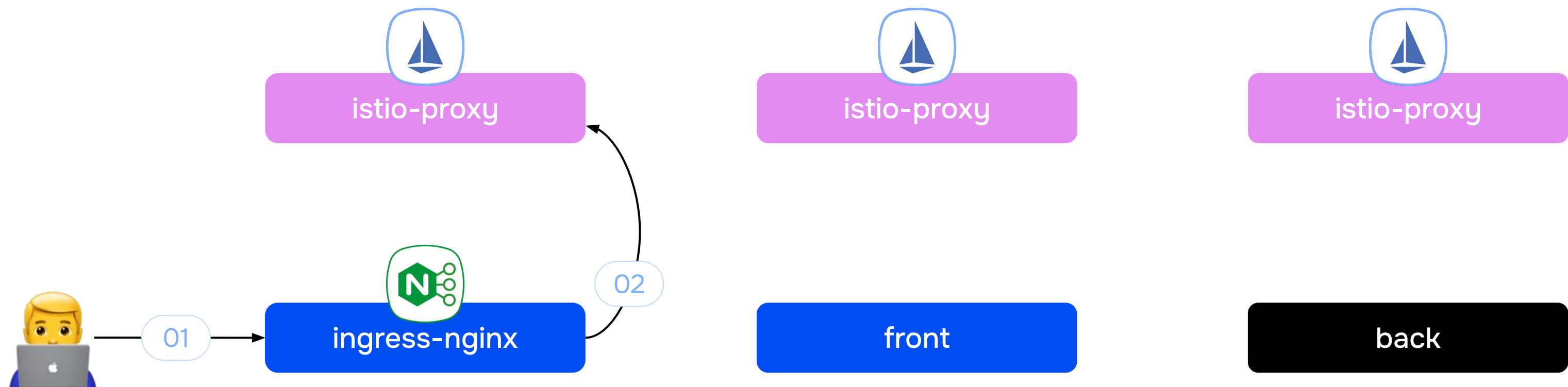




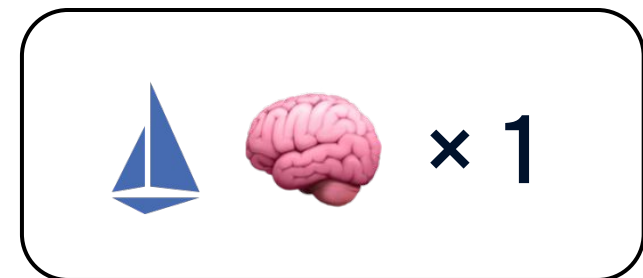
02

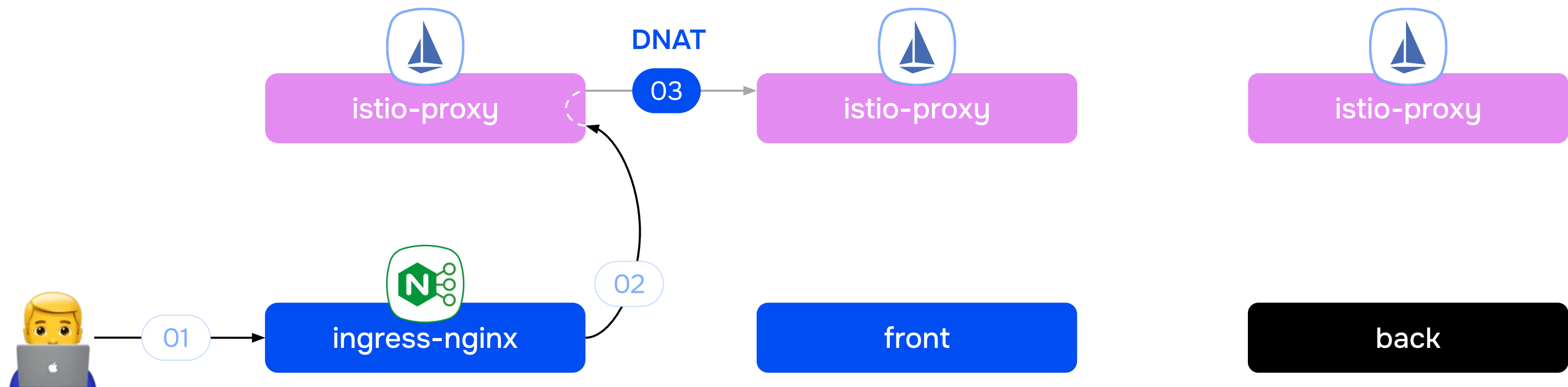
The ingress controller then routes the request to the **front** Pod, but sidecar-proxy intercepts the request thanks to DNAT.





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

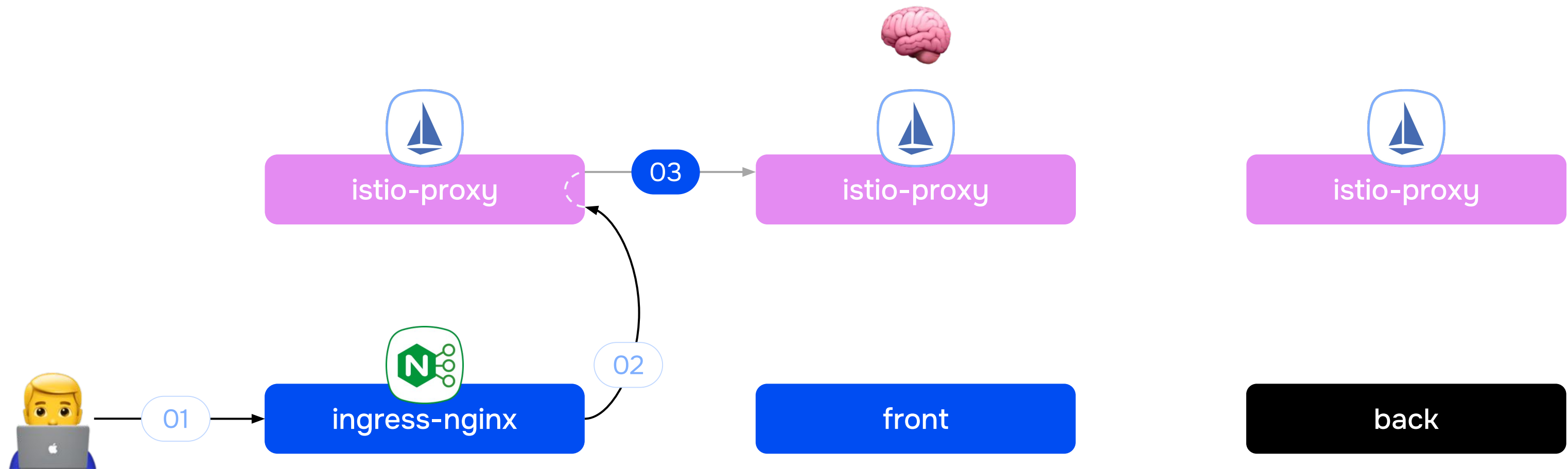




03

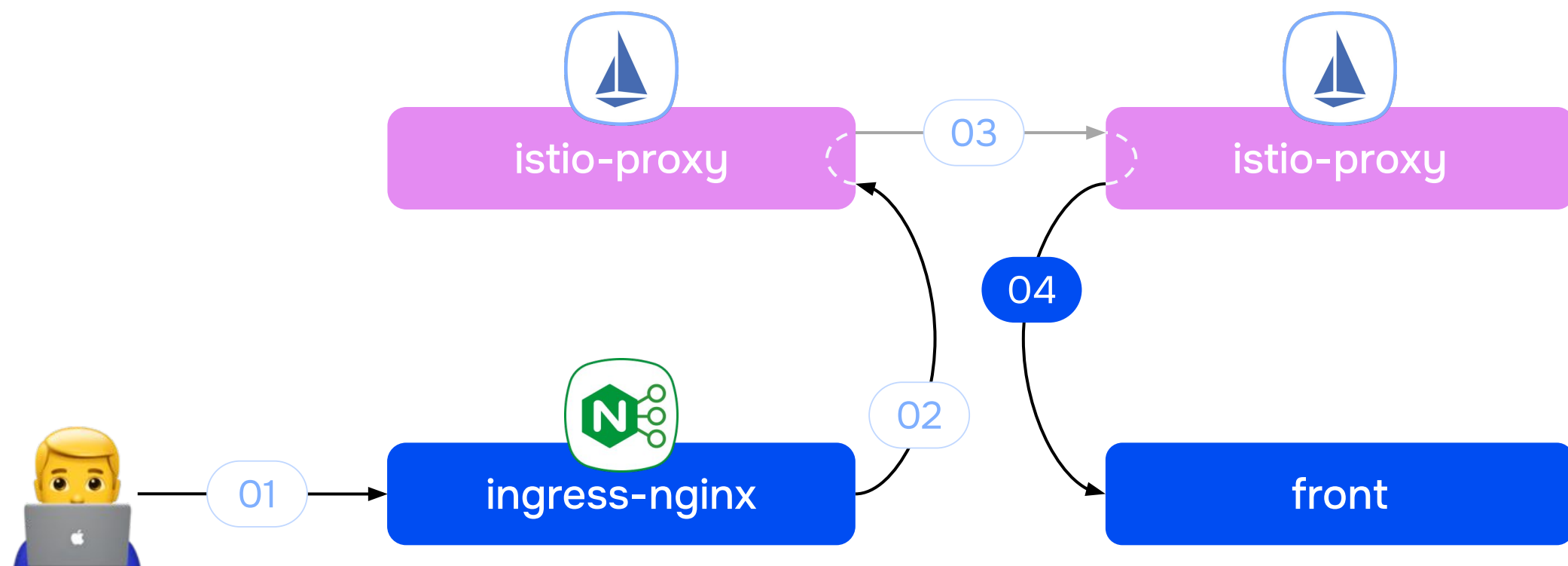
The sidecar-proxy proxies the request to the **front** Pod, but the request gets redirected to sidecar-proxy because of a special DNAT rule.



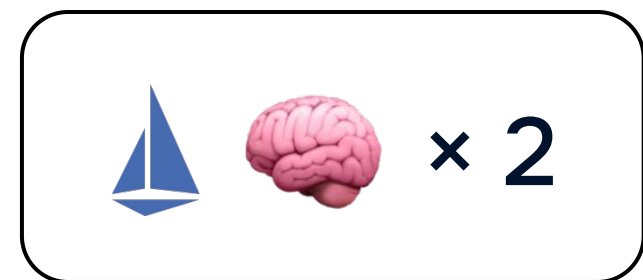


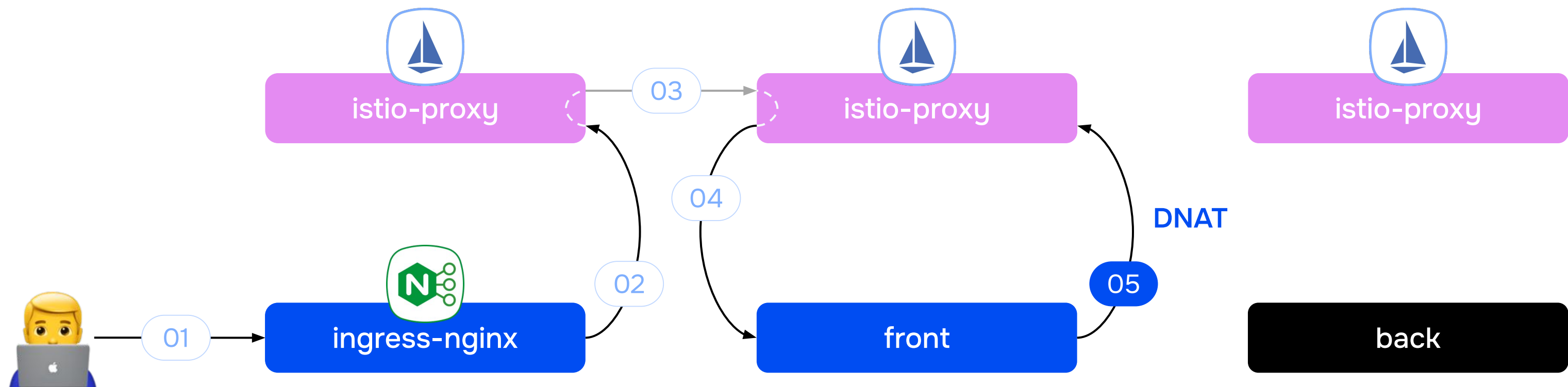
Next, the sidecar-proxy runs all the necessary validations...





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

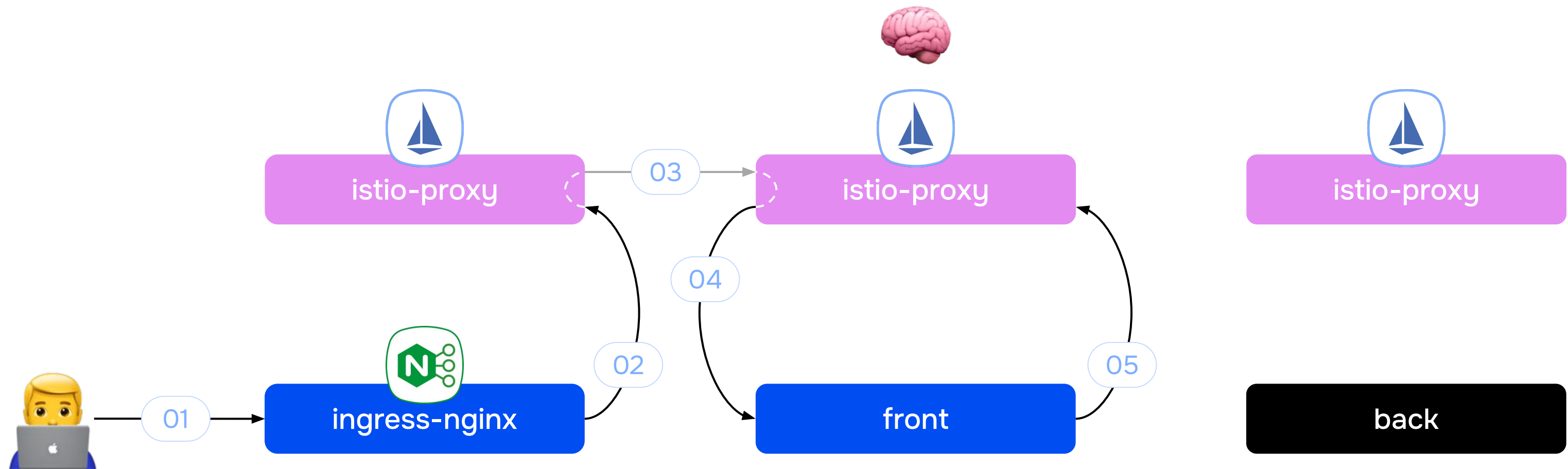




05

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

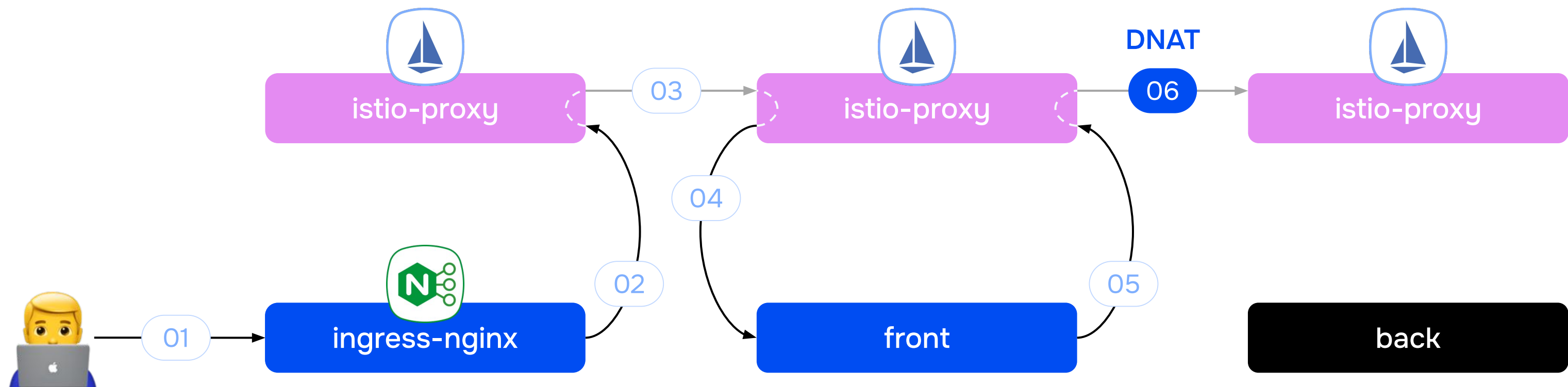




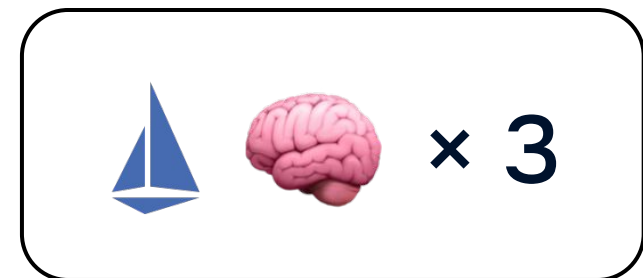
The sidecar-proxy decides what to do with the secondary request...

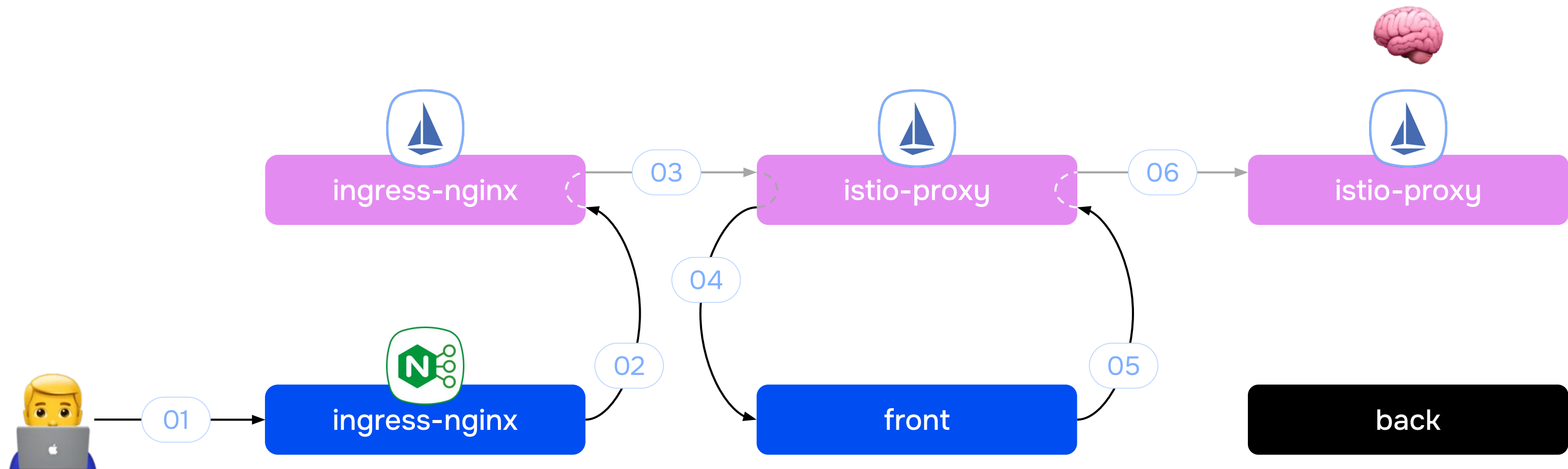


× 3



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

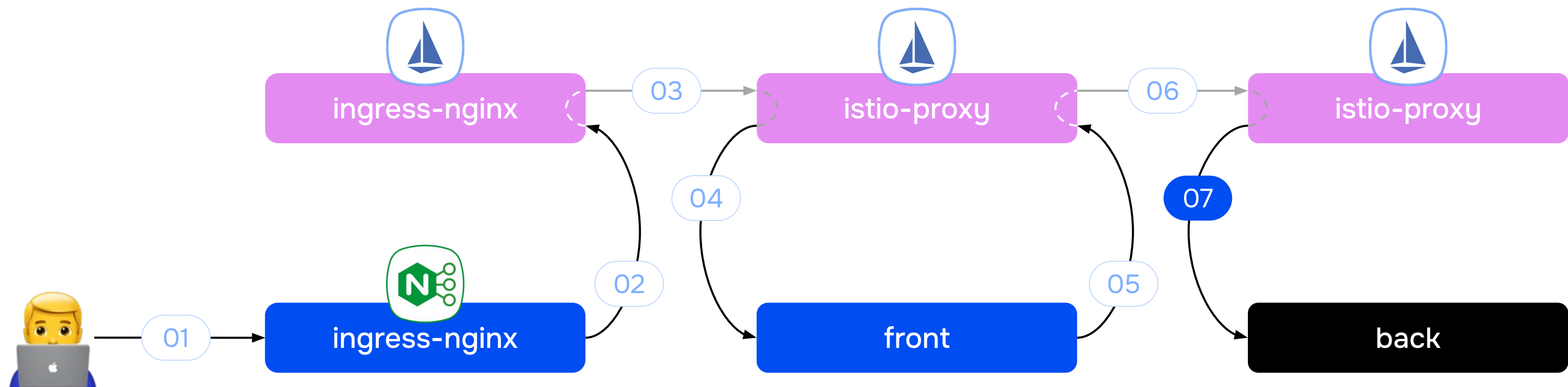




The sidecar-proxy performs the necessary validations...



× 4



07

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

