

headscales

How we are using integration testing to reimplement Tailscale

About us

Juan Font Alonso (@juanfont)

- Headscale creator
- European Space Agency
- First boss of Kristoffer
- Attention span of a goldfish on caffeine

Kristoffer Dalby

- Headscale high scorer
- Member of Technical Staff at Tailscale
- Juan Fonts manager at ESA
- Wants to rewrite Headscale in new langs

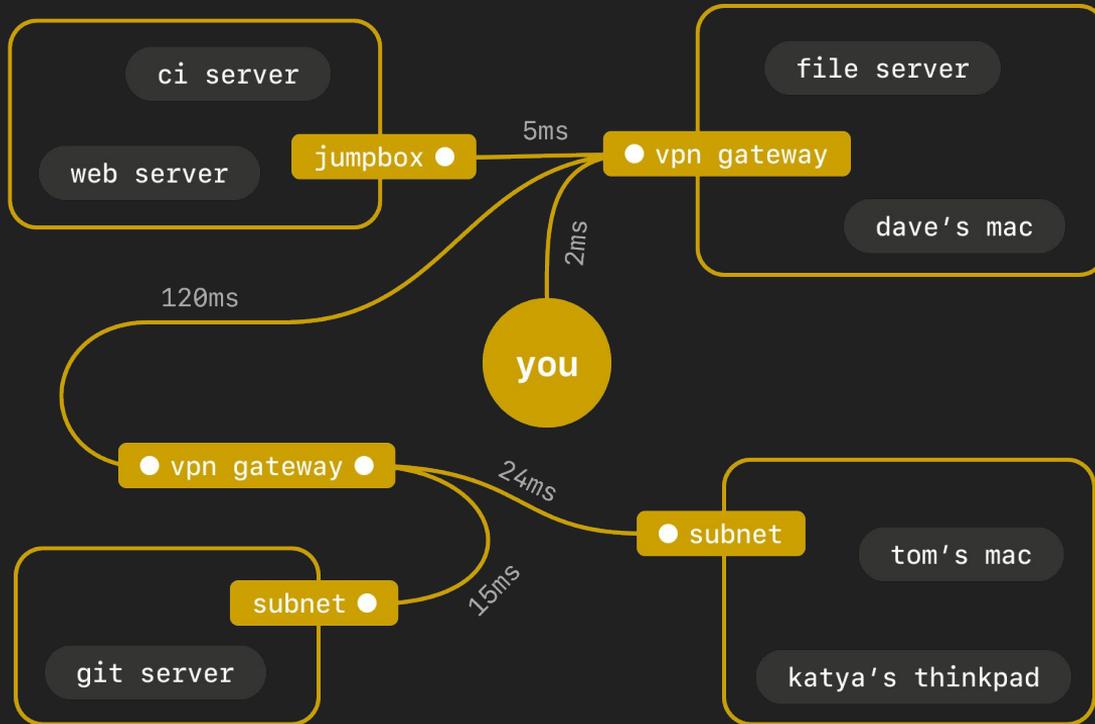


Who has heard of

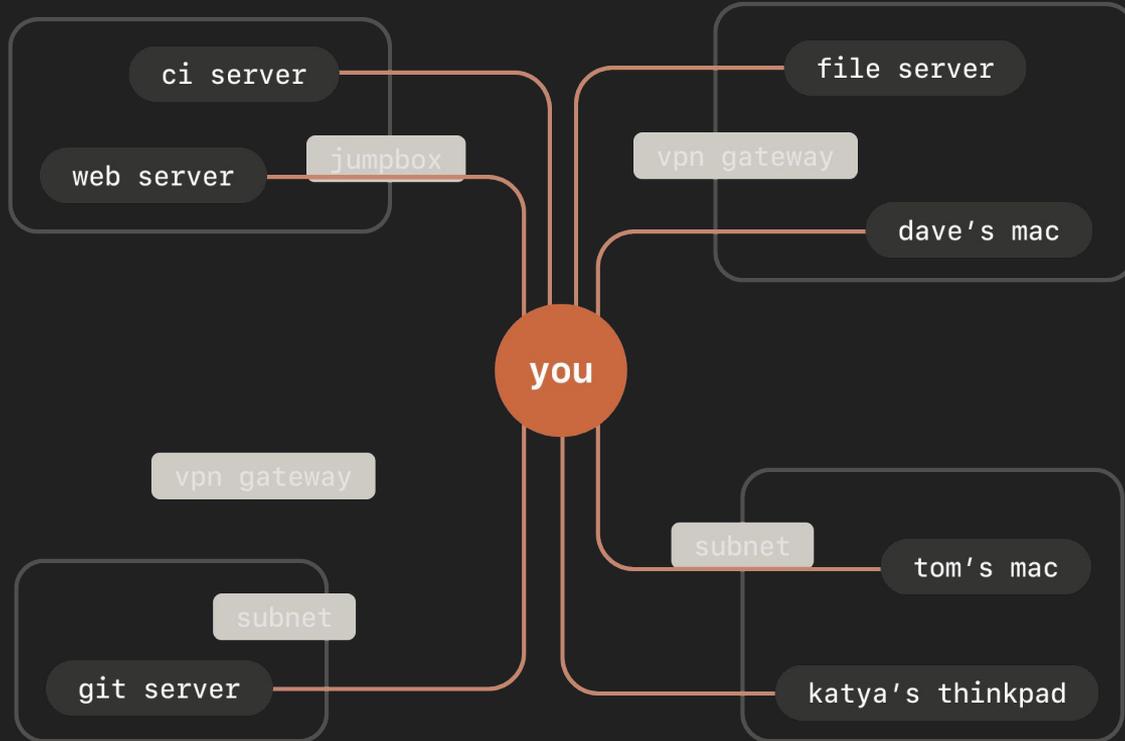
Tailscale

Headscale

What is Tailscale?



What is Tailscale?



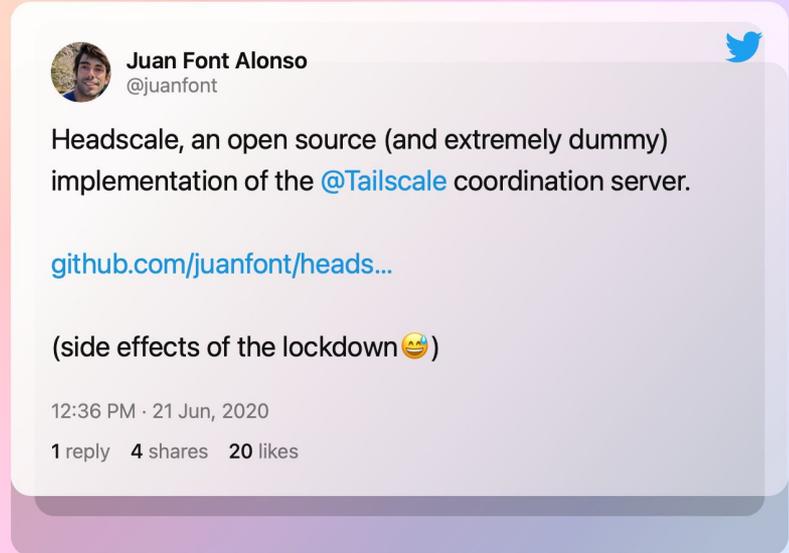
“

the open source Tailscale node software (...) talks to what we call a “coordination server” - essentially, a shared dropbox for public keys.

<https://tailscale.com/blog/how-tailscale-works/>

Quick Headscale history

- Juan makes Headscale to learn about how Tailscale works
- Kristoffer doesn't get it, happy with his Ansible WireGuard setup
- Juan does a bunch of work
- Headscale gets traction
- Kristoffer gets curious
- Kristoffer is afraid of breaking things
- Kristoffer did the logo!



What do we need to implement?

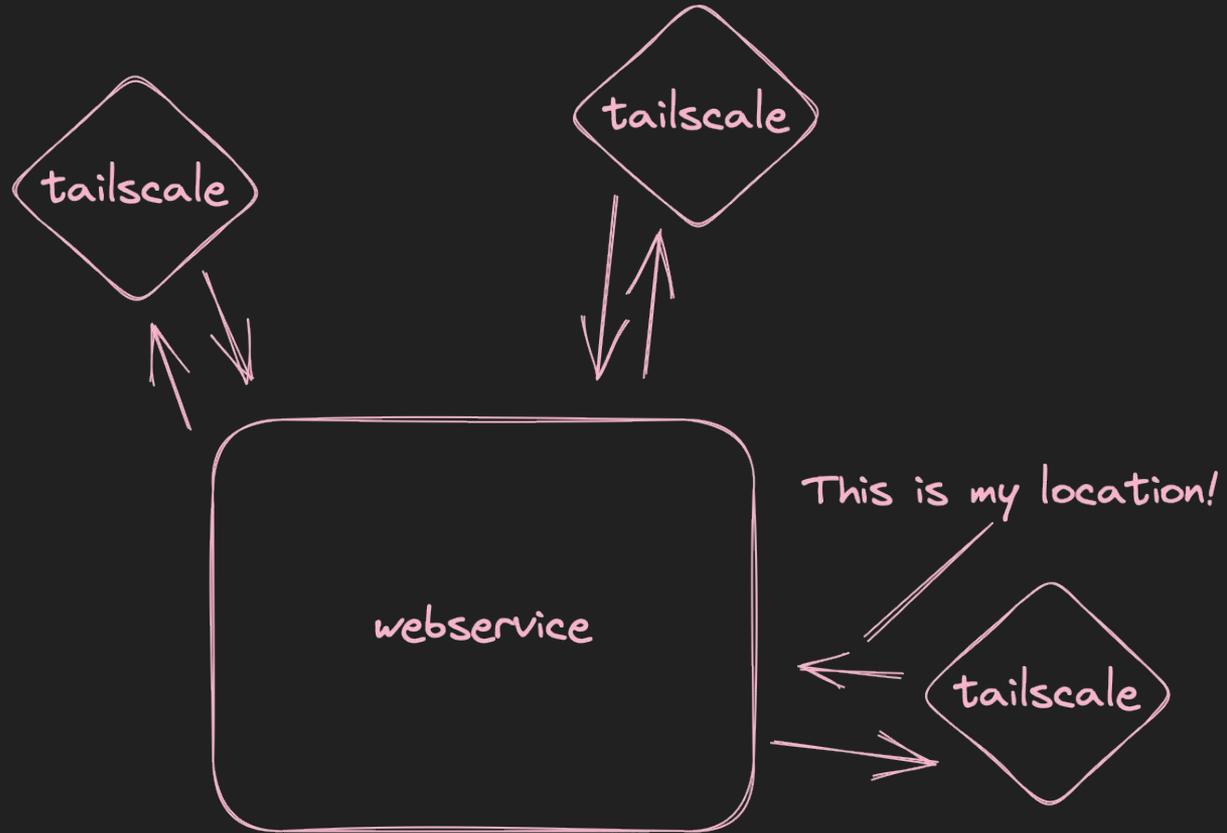
tailscale

tailscale

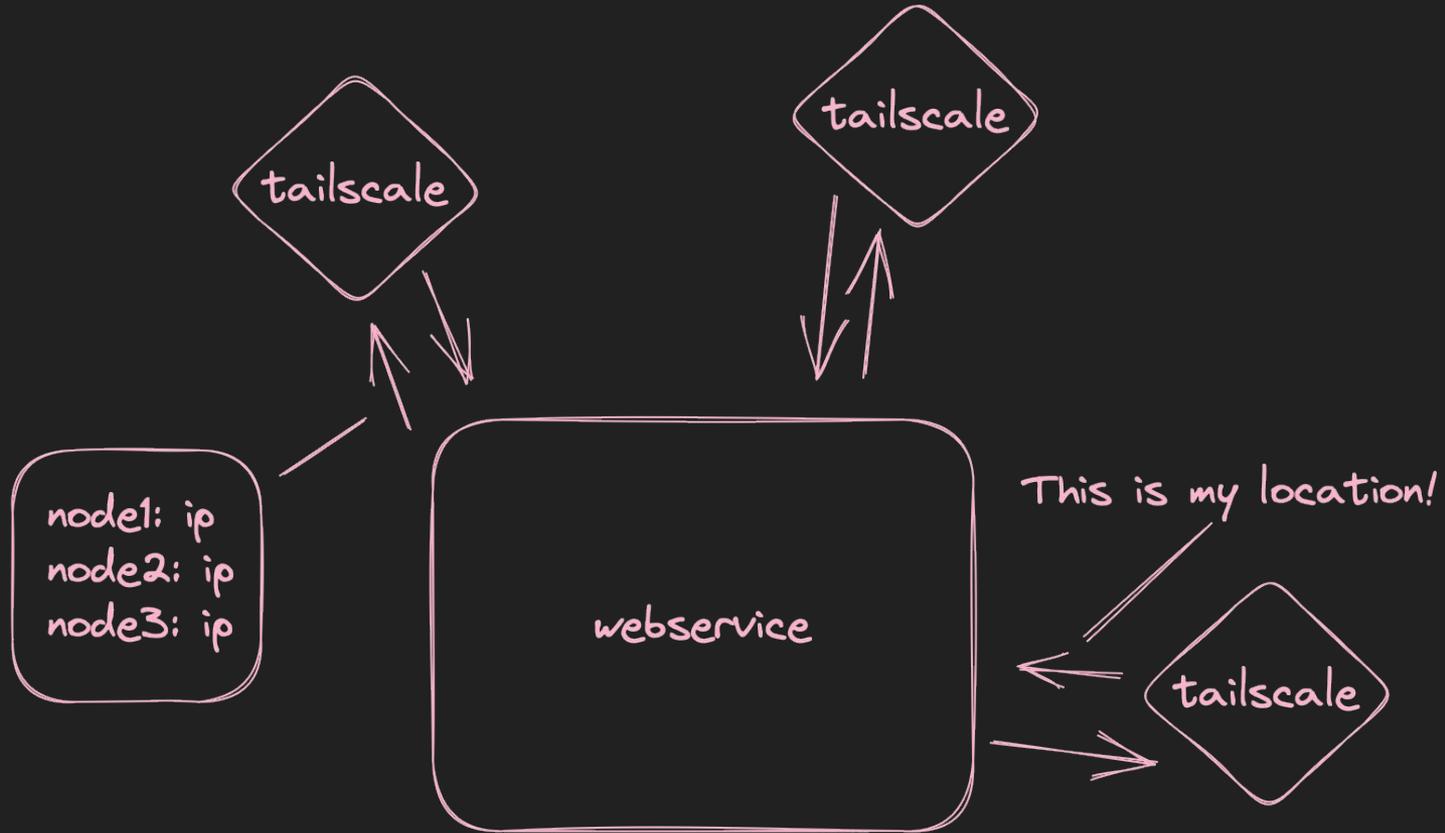
webservice

tailscale

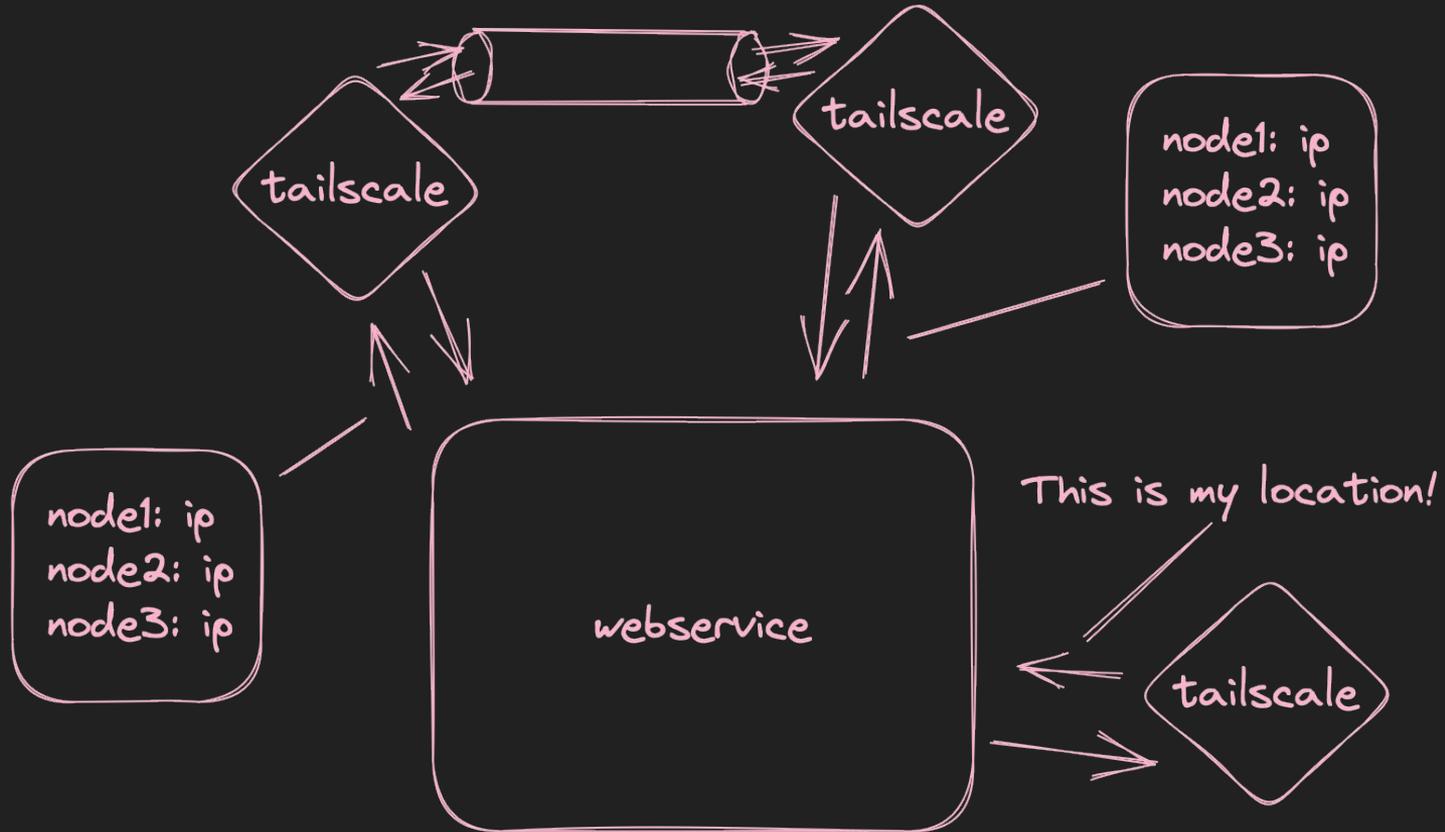
What do we need to implement?



What do we need to implement?

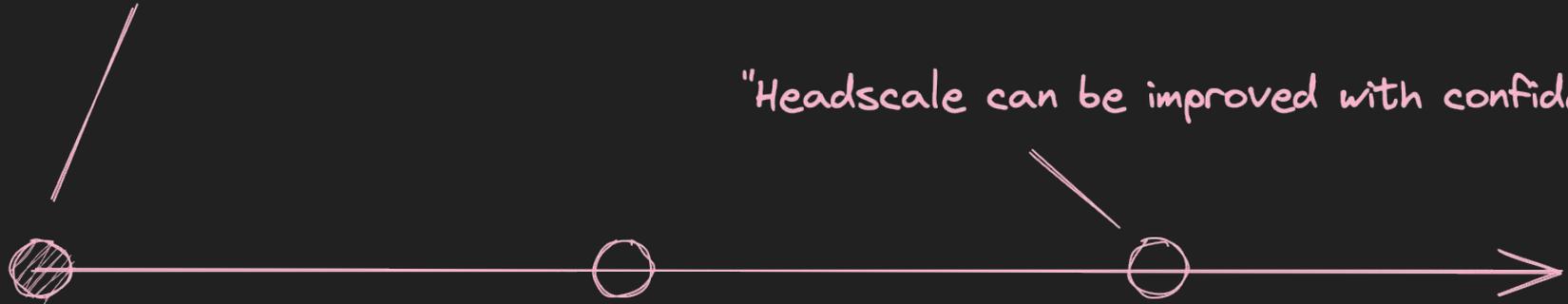


What do we need to implement?



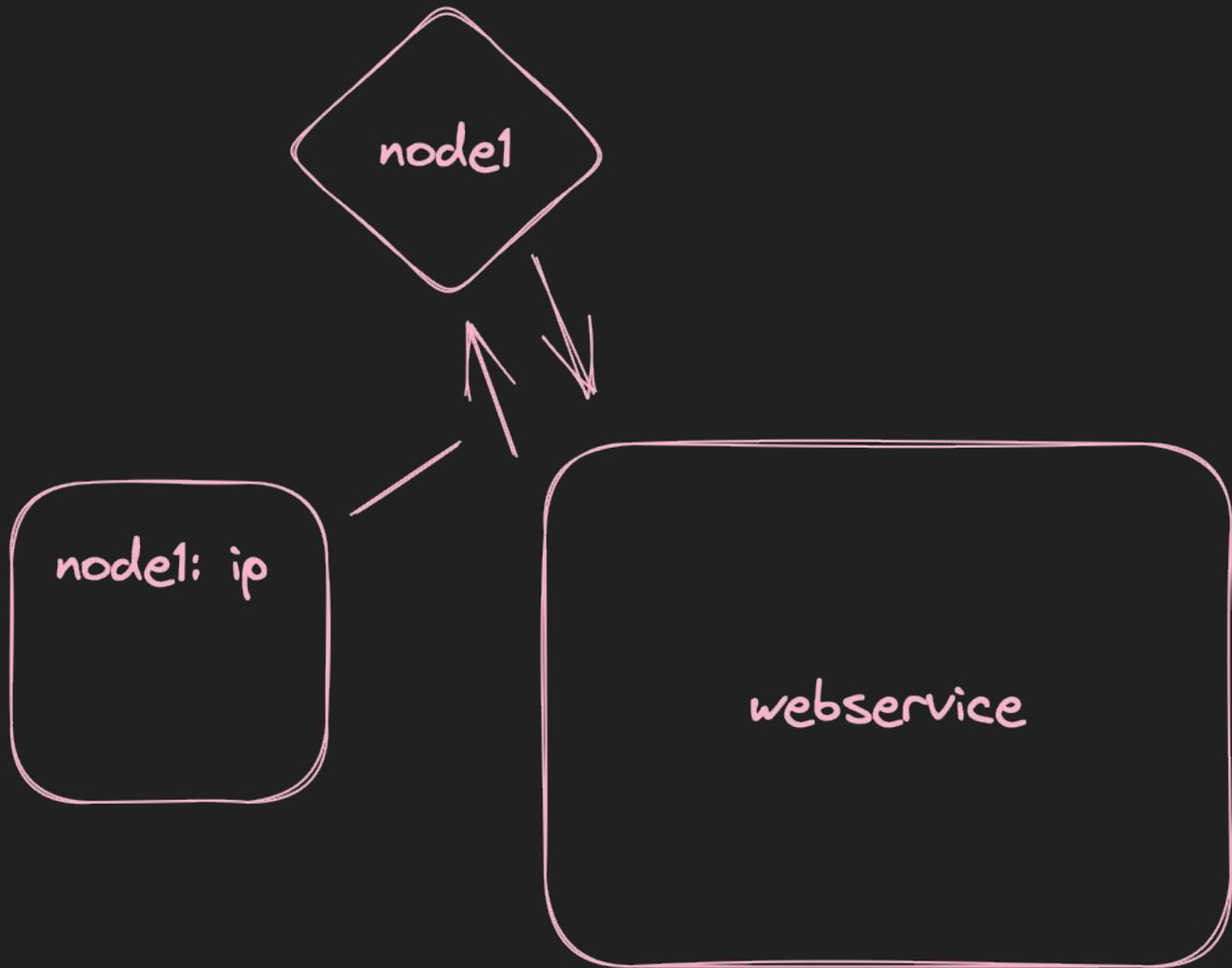
Headscale's three stages

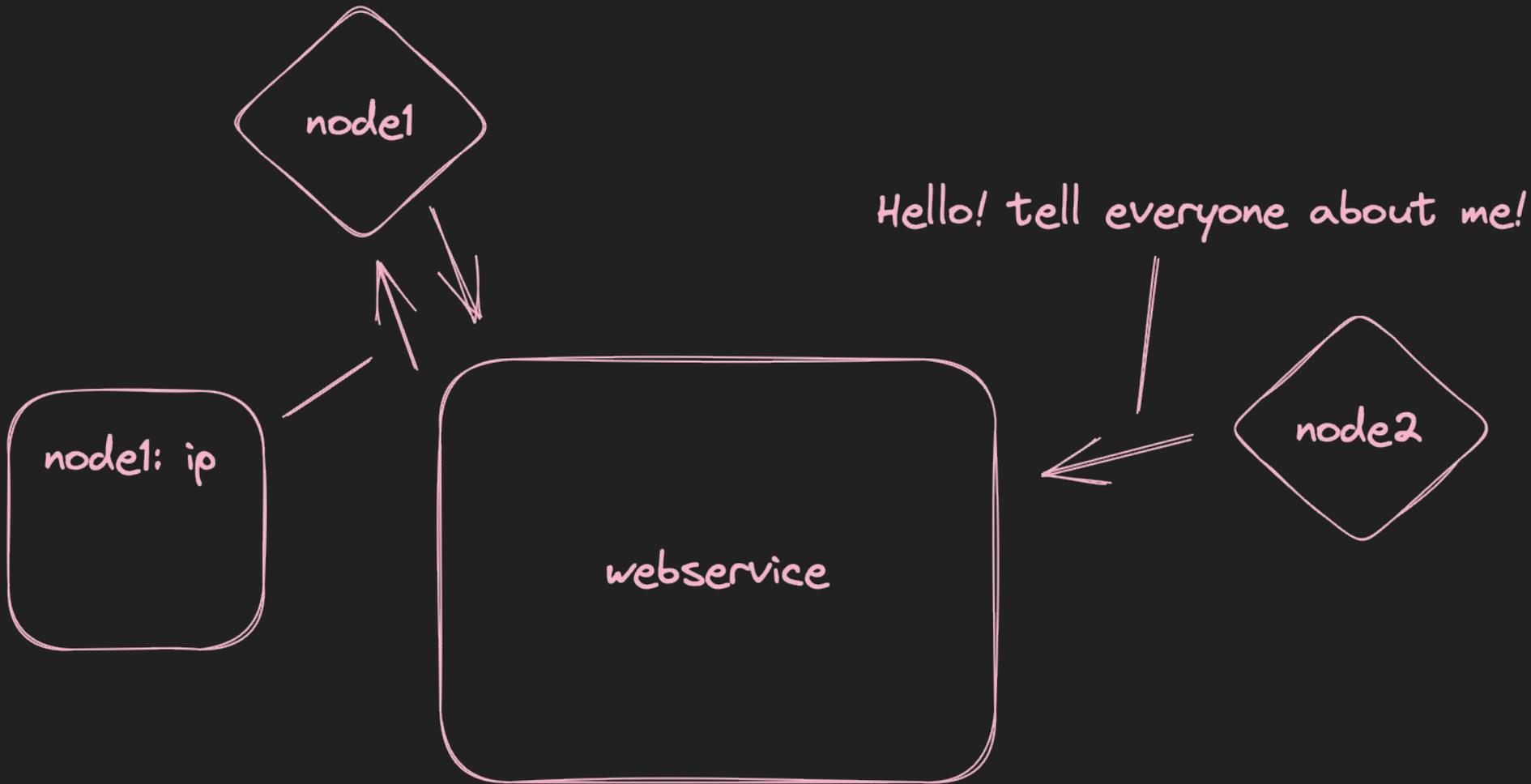
"the illusion of working, except when it does not"

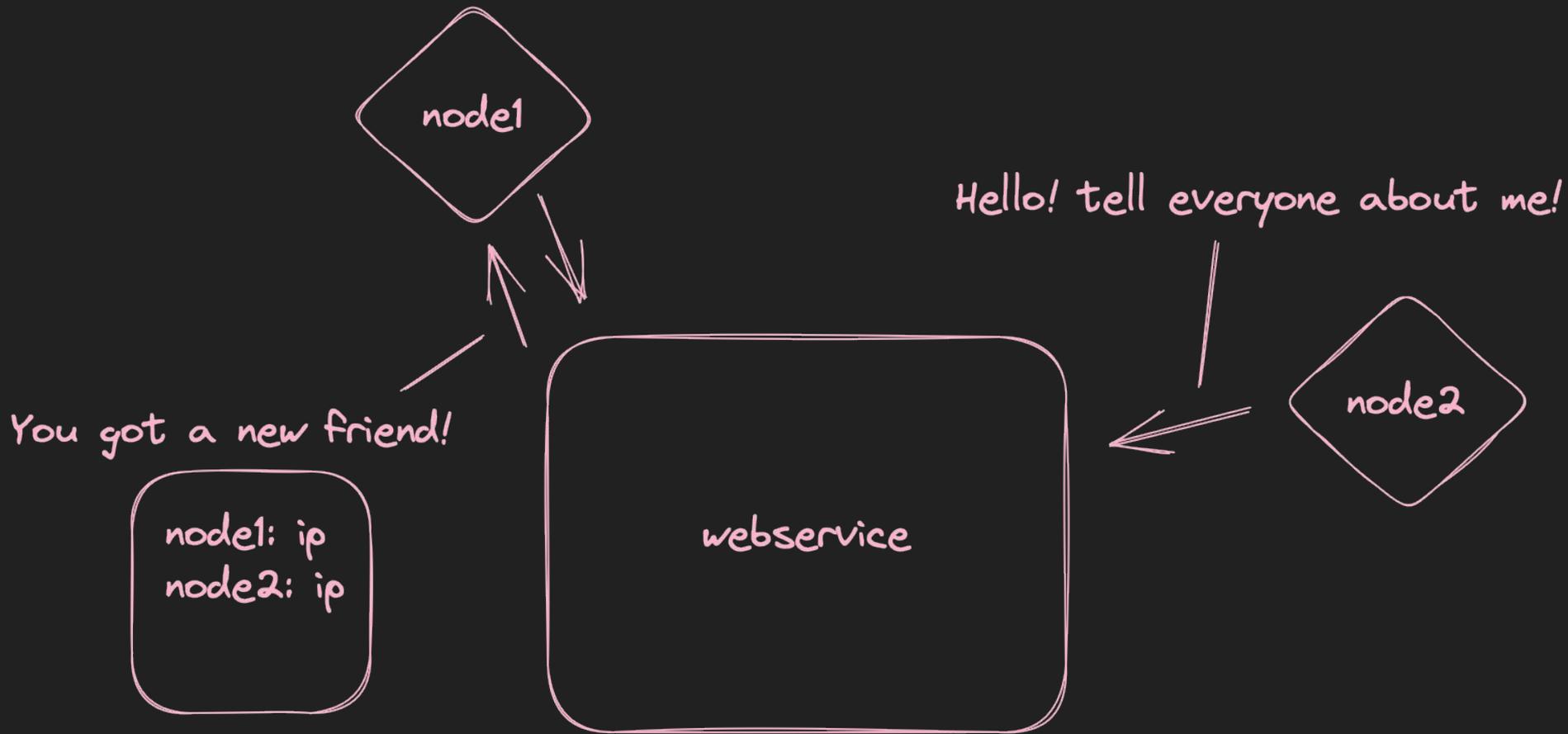


"Headscale can be improved with confidence"

"works most of the time"





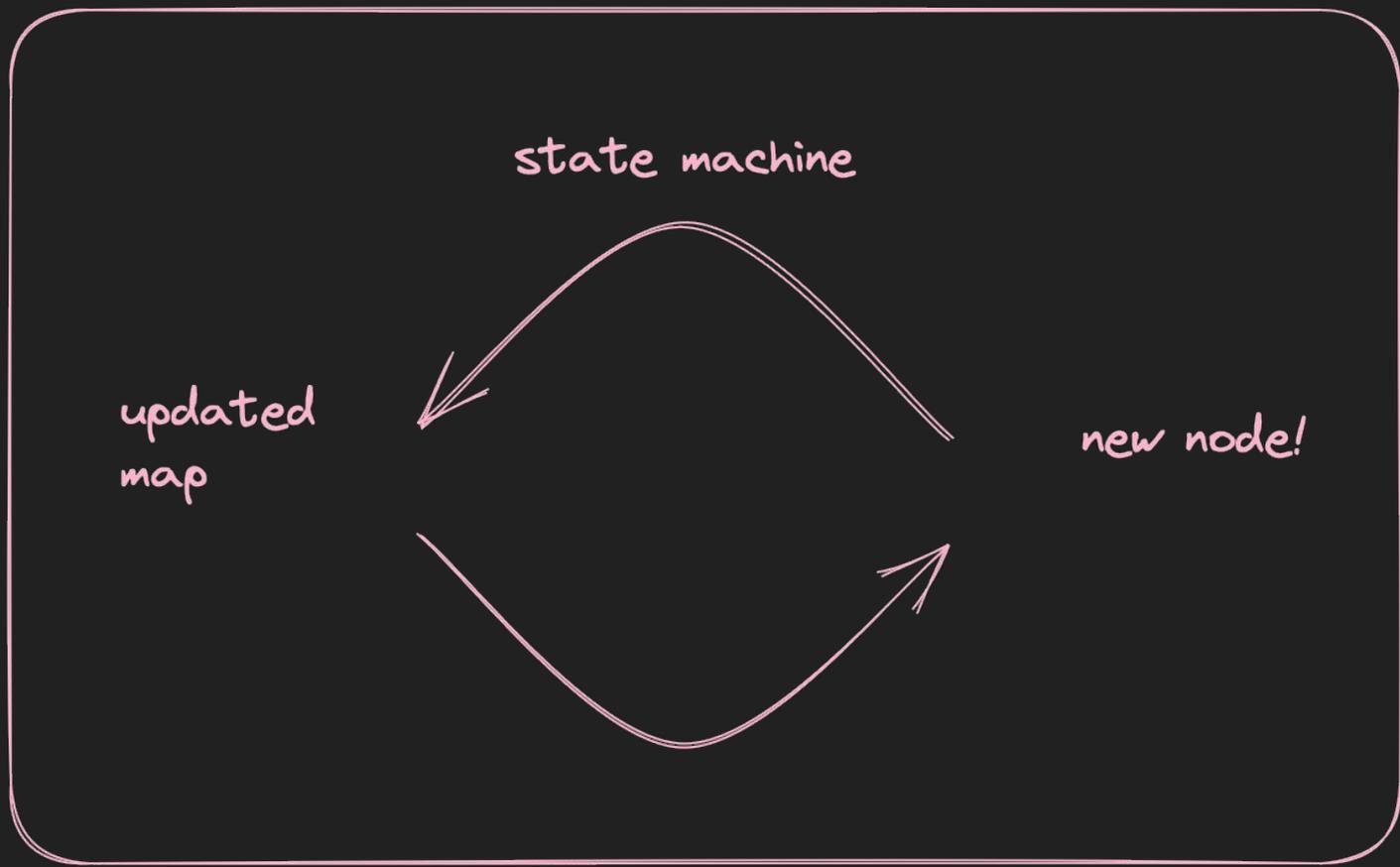


headscale

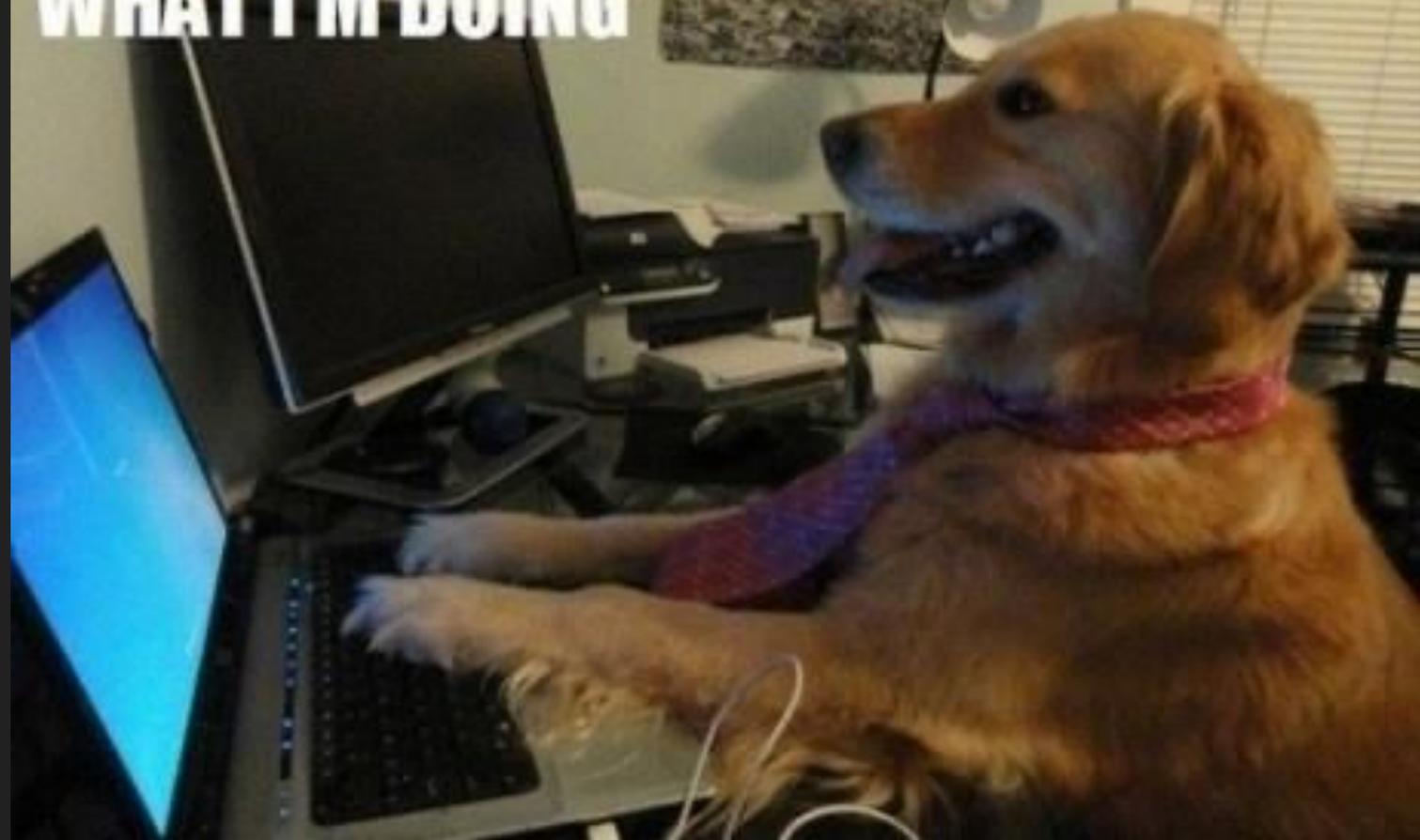
state machine

updated
map

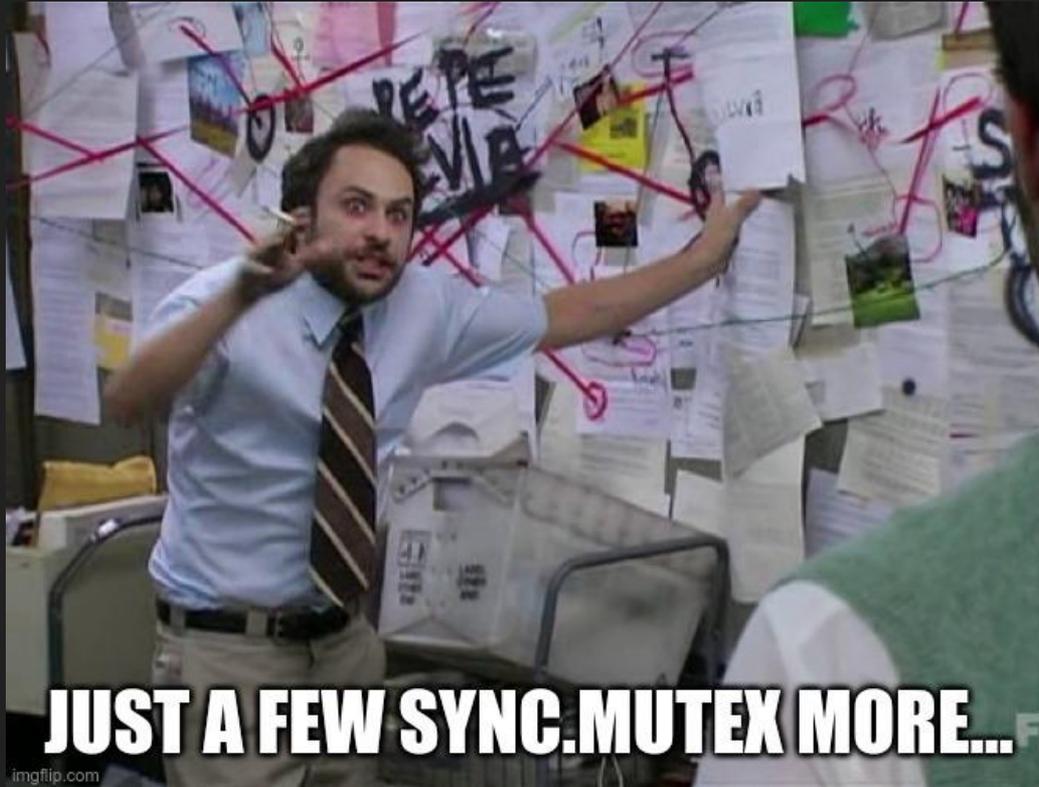
new node!



**I HAVE NO IDEA
WHAT I'M DOING**



And an unorthodox approach to handle concurrency

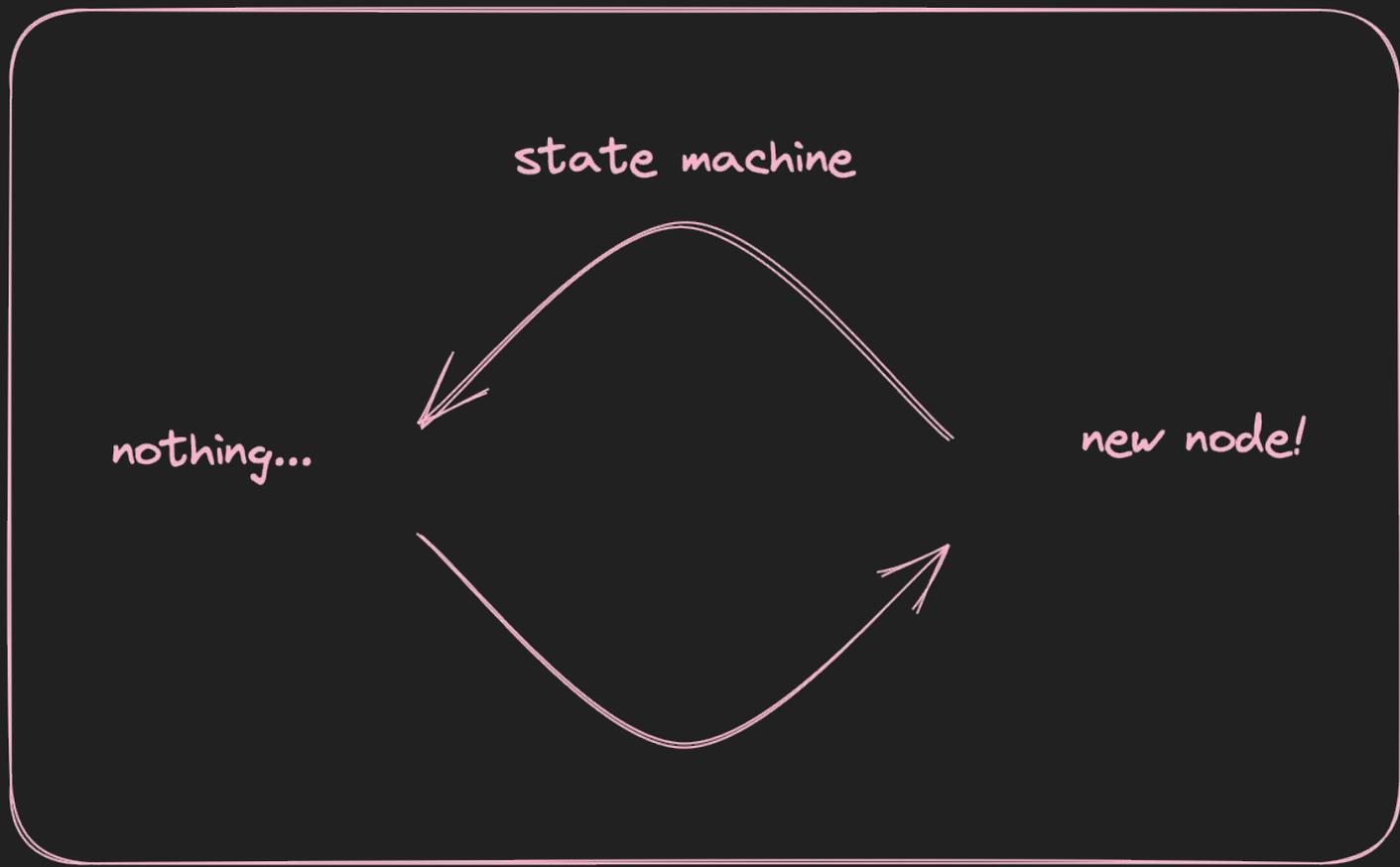


headscale

state machine

nothing...

new node!

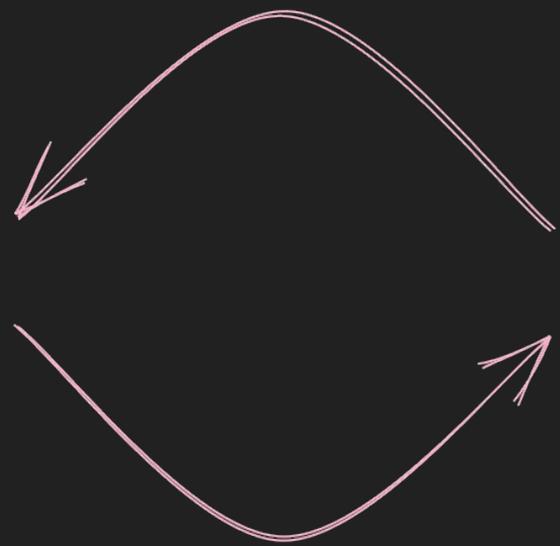


headscale

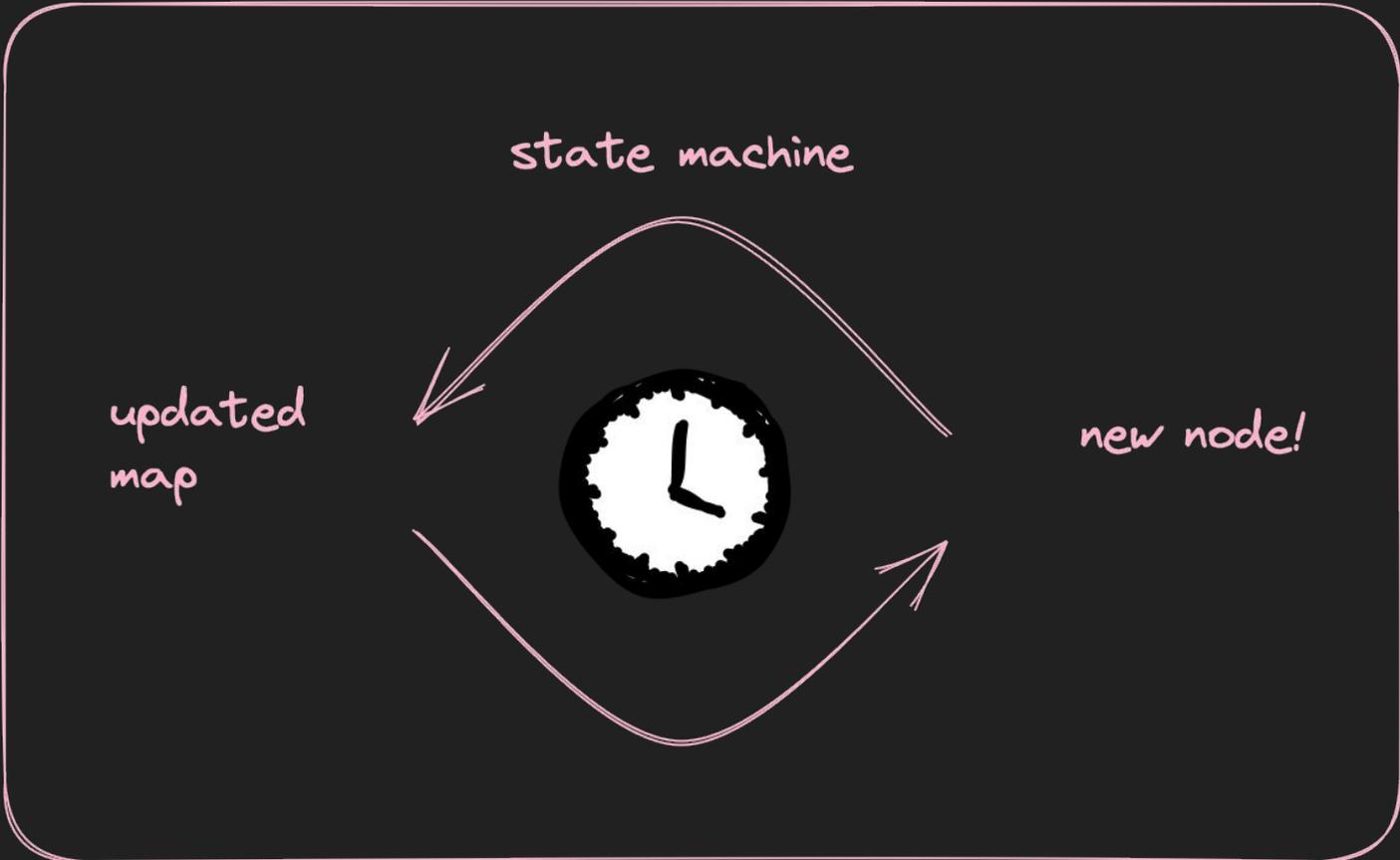
state machine

node1: ip
<missing>

new node!



headscale



headscale

state machine

updated
map



new node!

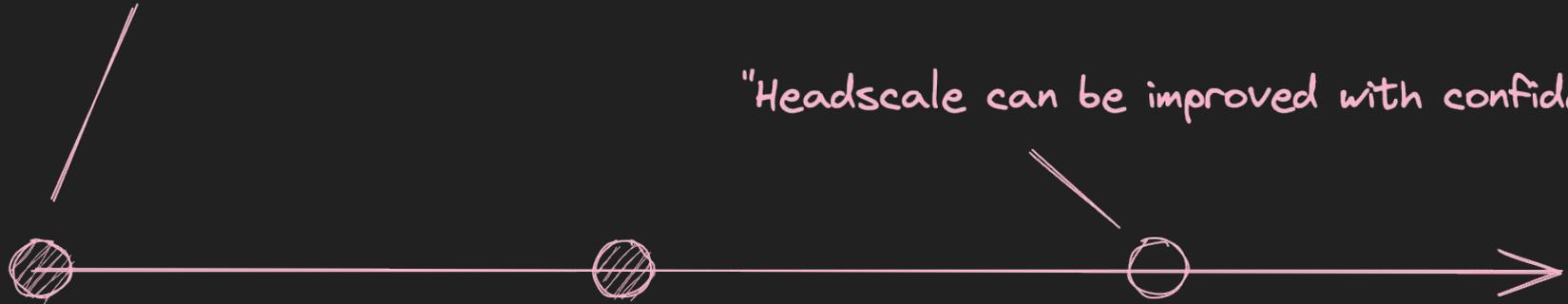


A man with a mustache and long hair, wearing a pinstriped suit and tie, is shown from the chest up. He is holding a camera with both hands and looking towards the left. The background features a bar with various bottles on shelves and a bookshelf with books. The text "60% of the time, it works every time!" is overlaid on the bottom half of the image.

**60% of the time, it works
every time!**

Stage two - works most of the time

"the illusion of working, except when it does not"



"Headscale can be improved with confidence"

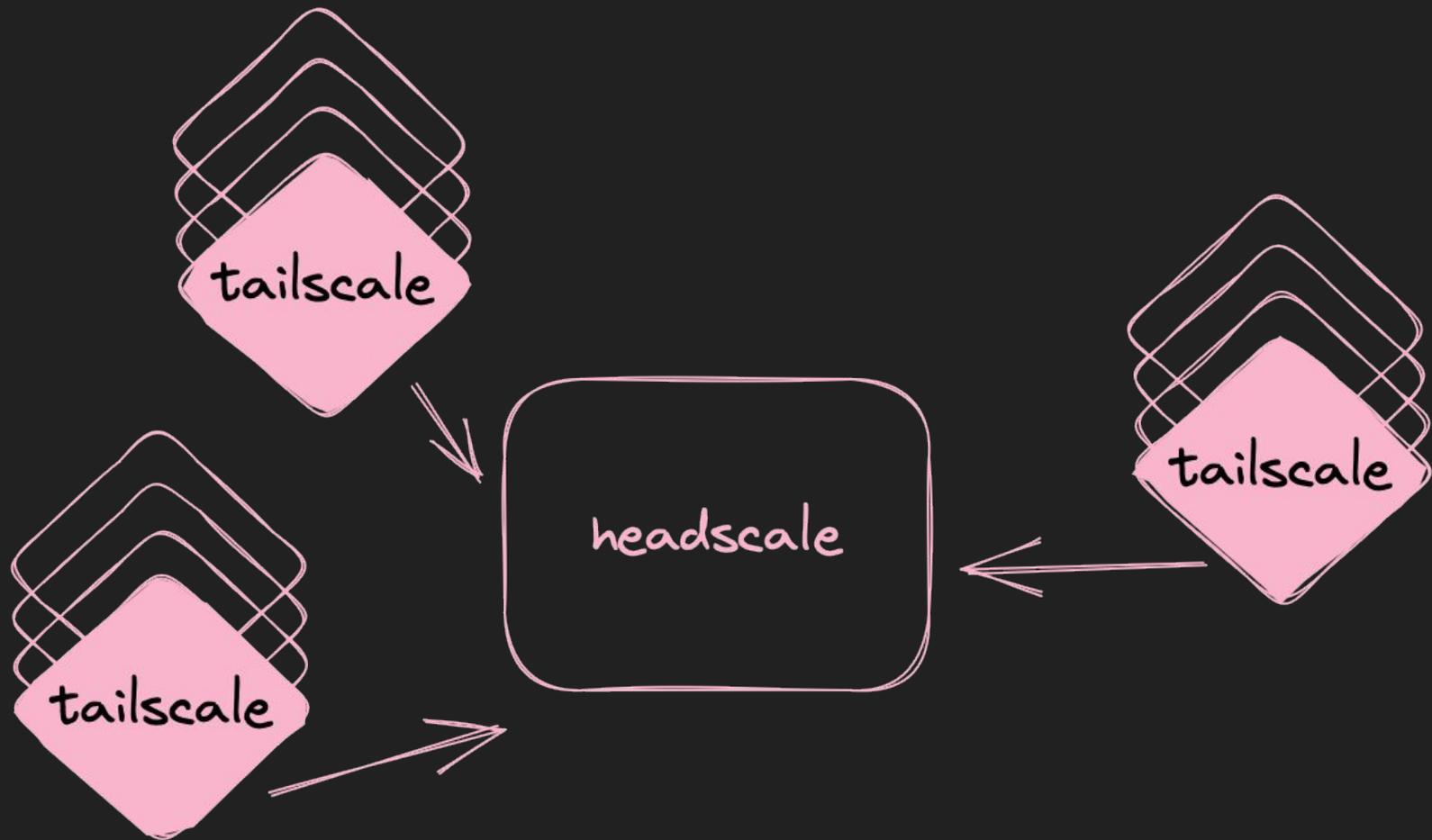
"works most of the time"

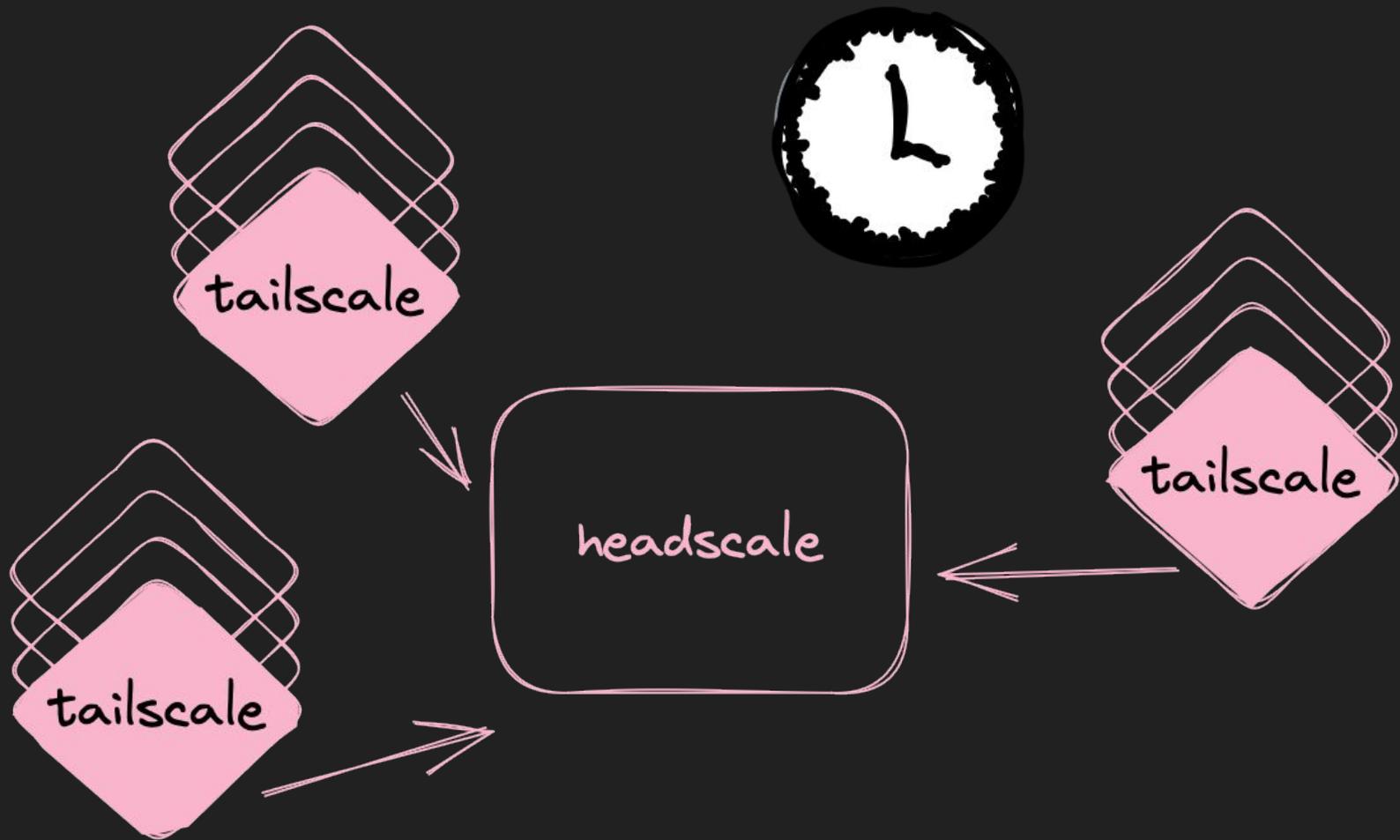
headscale

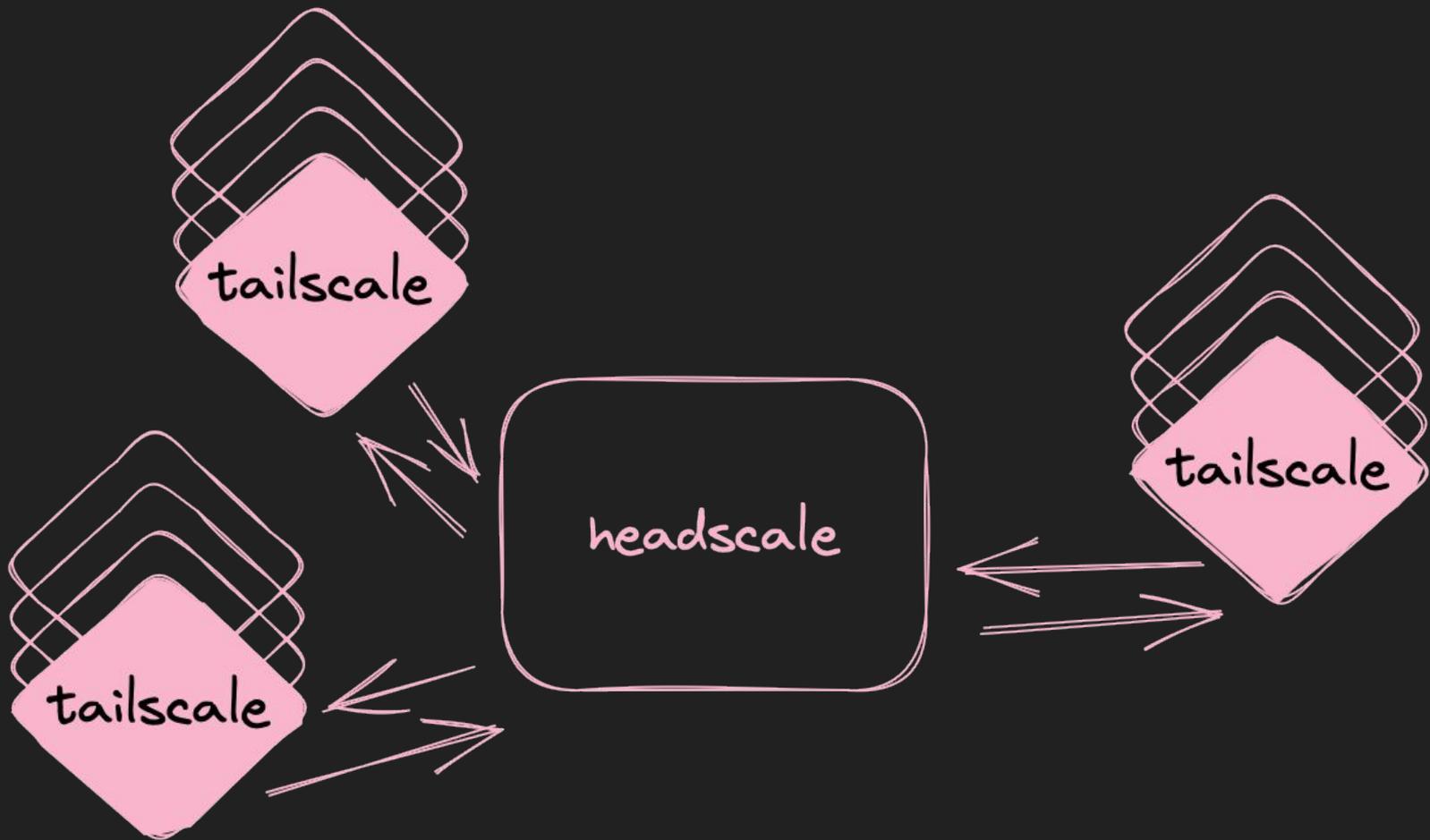


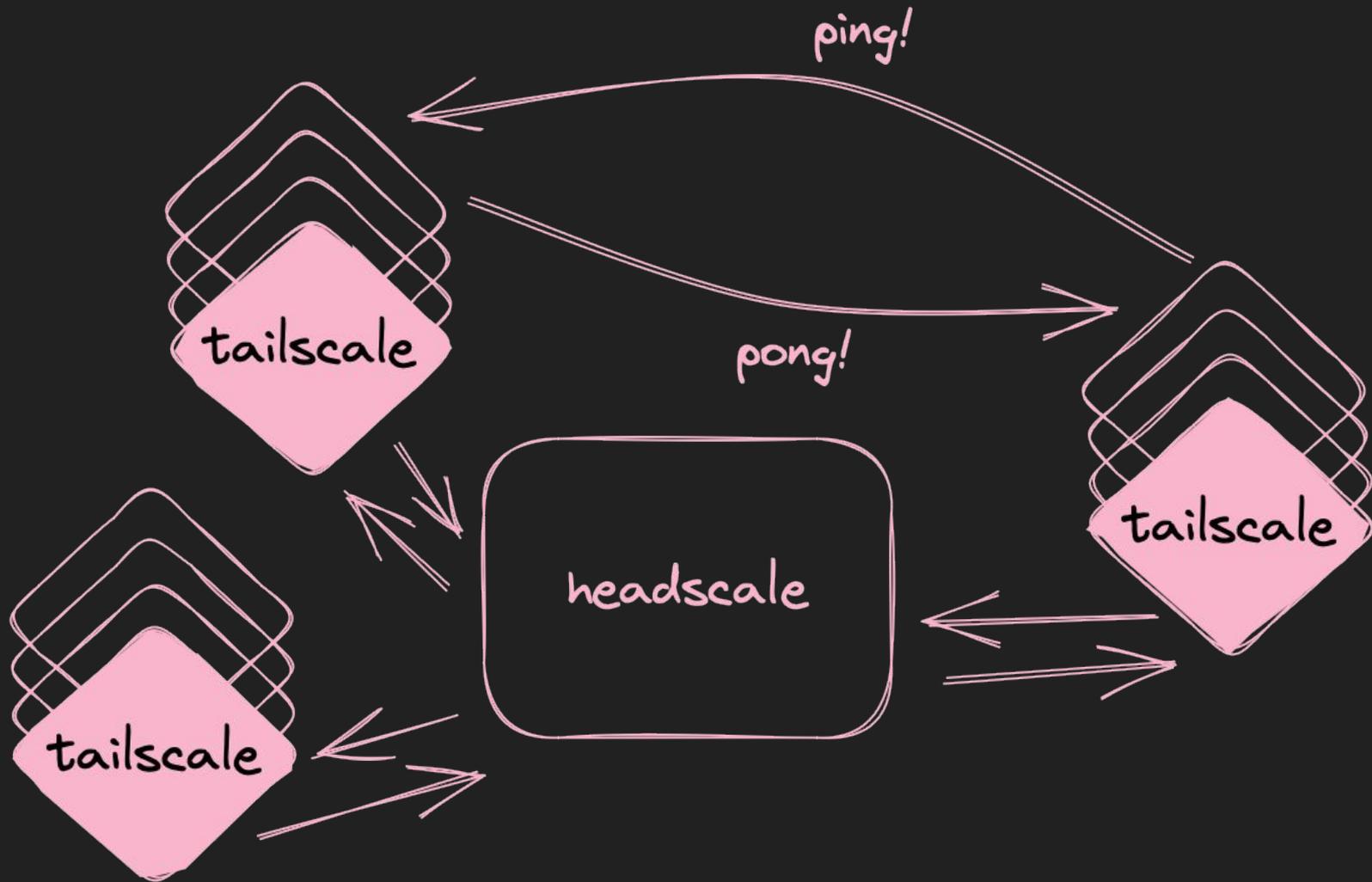












haha!

70/100 runs failed

but!

now, we have a goal we can measure

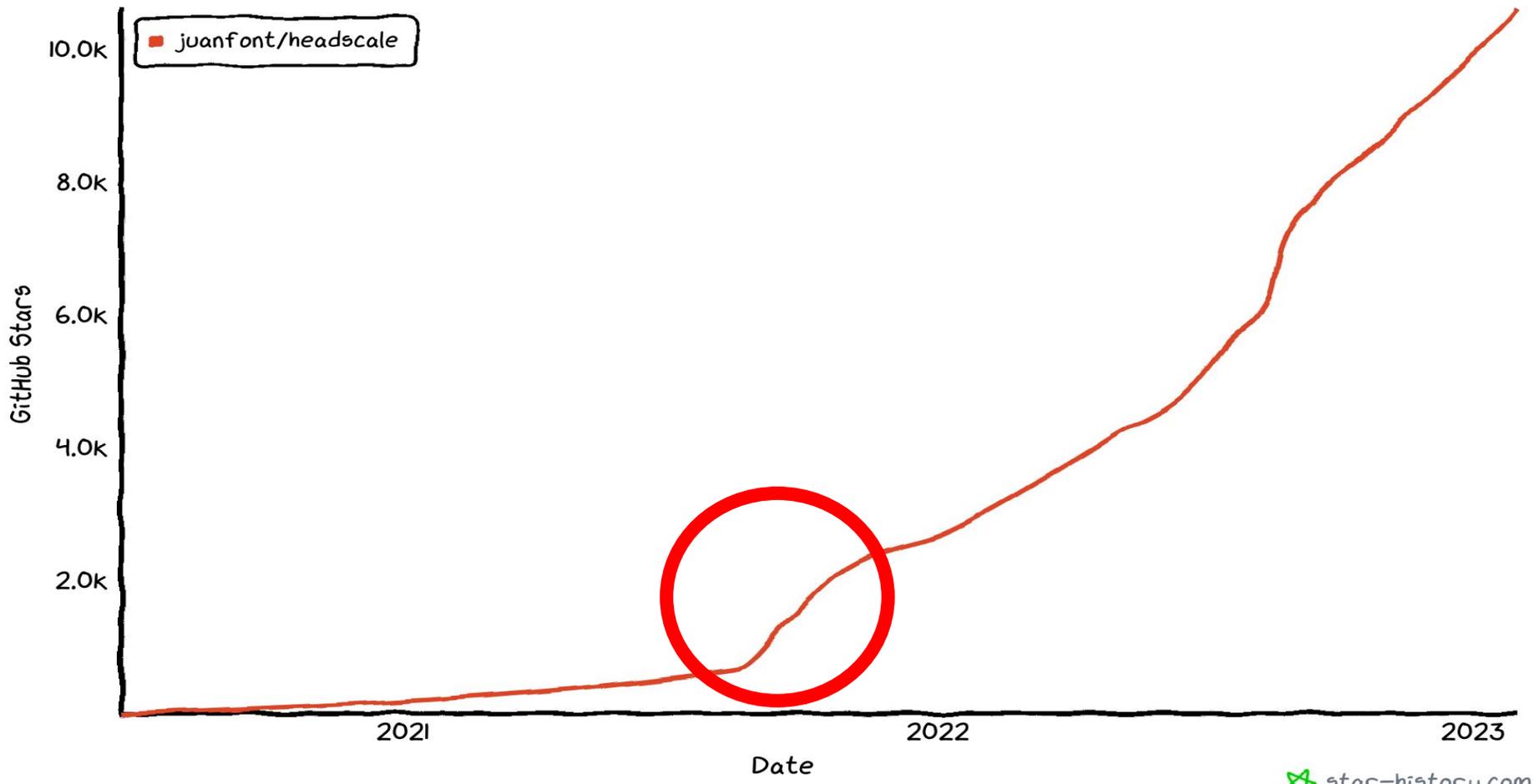
We can build on top of that

- We had two problems
 - Reliably sending updates to clients (PR#83)
 - 30% of the tests failed!
 - Determining if we need to update a client (PR#84)
 - 1% of the tests failed!

Changing the Rambo culture takes time

- We kept merging untested stuff for a while
- In the Real World™ things kind of worked, so didn't feel the pressure
- Popularity kept growing, and so did external contributions

Star History



Stage three - actually works

"the illusion of working, except when it does not"

"Headscale can be improved with confidence"



"works most of the time"

LJDD (Leeroy Jenkins Driven Development)

- Simplify and improve register/reauth flow #227

+1,305 -537 

- Drop Gin as web framework for TS2019 API #656

+965 -409 

- TS2021 (Noise-based Tailscale v2 protocol) #738

+1,600 -951 

- Rename Namespace -> User #1144

+3,227 -3,297 

Integration testing helps with that little minor detail

- We have a big dependency - **client software that we do not control**
 - Although we get heads-up when they will change something big!
- Our tests target multiple Tailscale versions
 - HEAD
 - unstable
 - Nine official minors releases (v1.20 - v1.36)
- We will know if Tailscale changes something



Integration testing supporting FOSS development

- Tests do not only help with software quality... but also the community around
- Maintainers can be more “confident” in PRs from contributors
- Contributors can be more confident when submitting a PR

Challenges

- Requiring tests makes some contributors disappear
- Raises the learning curve (go test, our test framework vs no tests)
- It is hard to convince people how awesome tests are, not only a chore

Integration test “v2”

- Before, a lot of repeated / copied code
- High bar to add new tests
- Hard to update or change
- Do not rely on time.Sleep...
- Can be run in parallel
- No documentation or good examples

Abstracting things

- ControlServer
 - Implemented as hsic
(HeadscaleInContainer)
 - Exposes convenience functions:
 - WaitForReady
 - CreateNamespace
 - CreateAuthKey
 - ListMachinesInNamespace
- TailscaleClient
 - Implemented as tsic
(TailscaleInContainer)
 - Exposes convenience functions:
 - Up
 - IPs
 - FQDN
 - Status
 - WaitForPeers
 - Ping

What do we test now?

- Connectivity between nodes by IP and by MagicDNS within the Tailnet
 - Taildrop files all to all
 - All registration flows (authenticated keys, web+cli flow, OpenID Connect)
 - Isolated nodes and Ping via embedded DERP server
 - SSH (all to all, negative)
-
- headscale CLI

Future

- Proper ACL tests
 - Unit tests
 - Suffer from “how we think it should be” (need to reverse-engineer what comes from SaaS)
- Use Tailscale SaaS to verify our tests correctness
- Run “Tailscale in VM” to test with non-userspace

TL;DR: I am here waiting for the next talk

- Integration tests are the key to our success.
- The name “Headscale” also helps. Much better than Ponytail-scale.
- How we have maintained compatibility with Tailscale.
- The reason we are able to take contributions.
- The tests are not perfect, but they save us a lot.

Tailscale Happy Hour



Today 18:30 to 20:00 at Brewdog (Putterie 20)

Questions and how to reach us

github.com/juanfont/headscale

Discord

and feel free to chat with us here at Fosdem!