

Pruning MTC down — and growing it back up

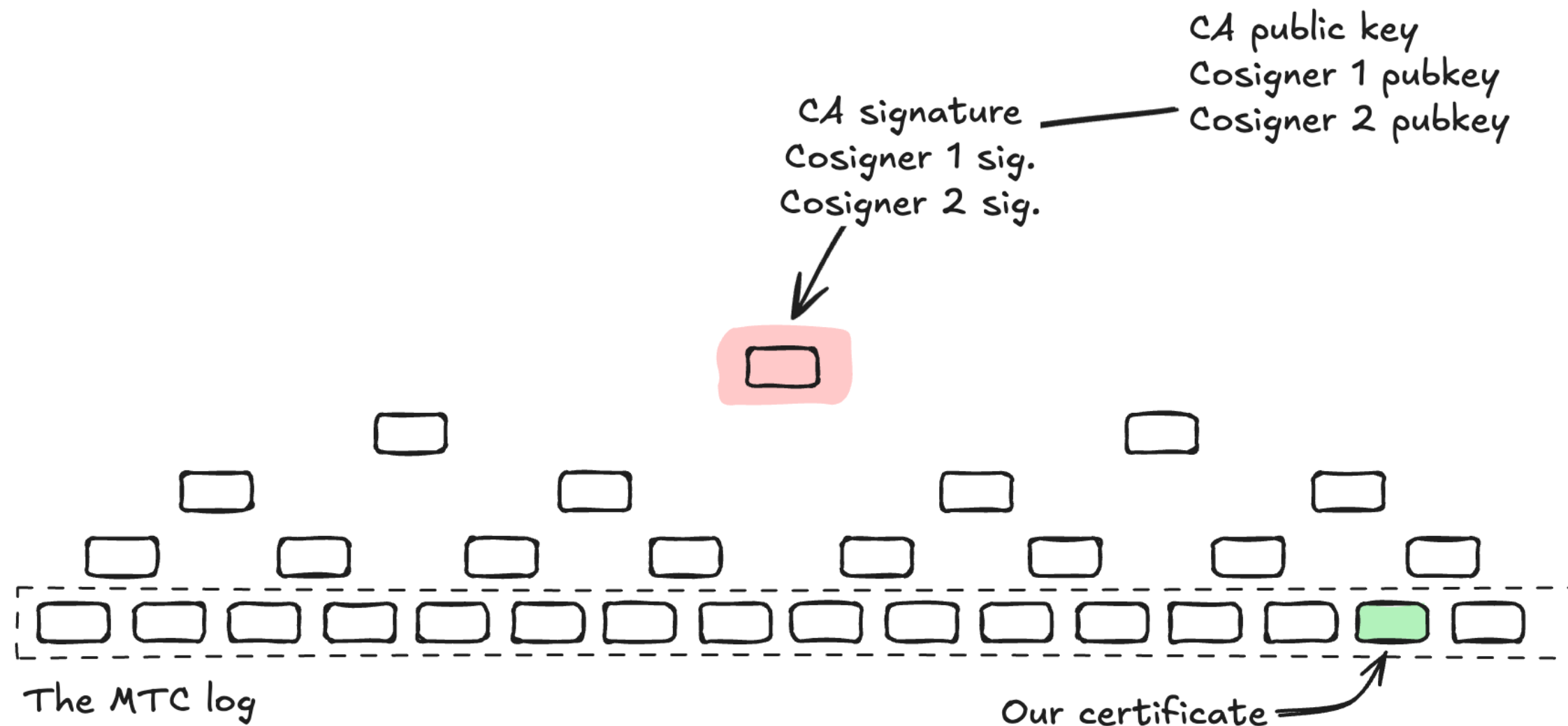
Filippo Valsorda

IETF 125 / PLANTS WG 

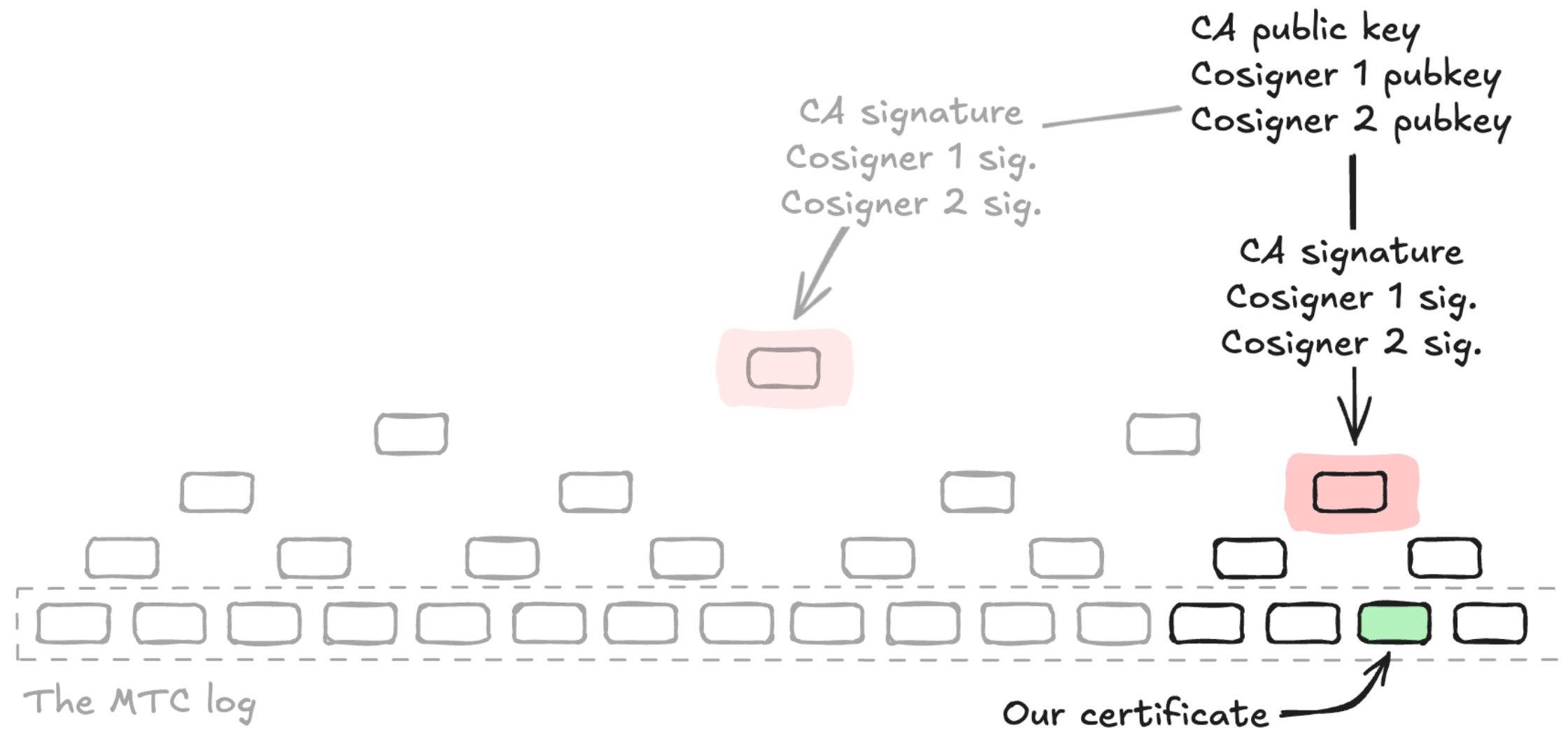
You can use MTC wherever

it's the **size benefits that need client
sync and negotiation*





Merkle Tree Certificates put certificates in a Merkle log, then sign the tree head

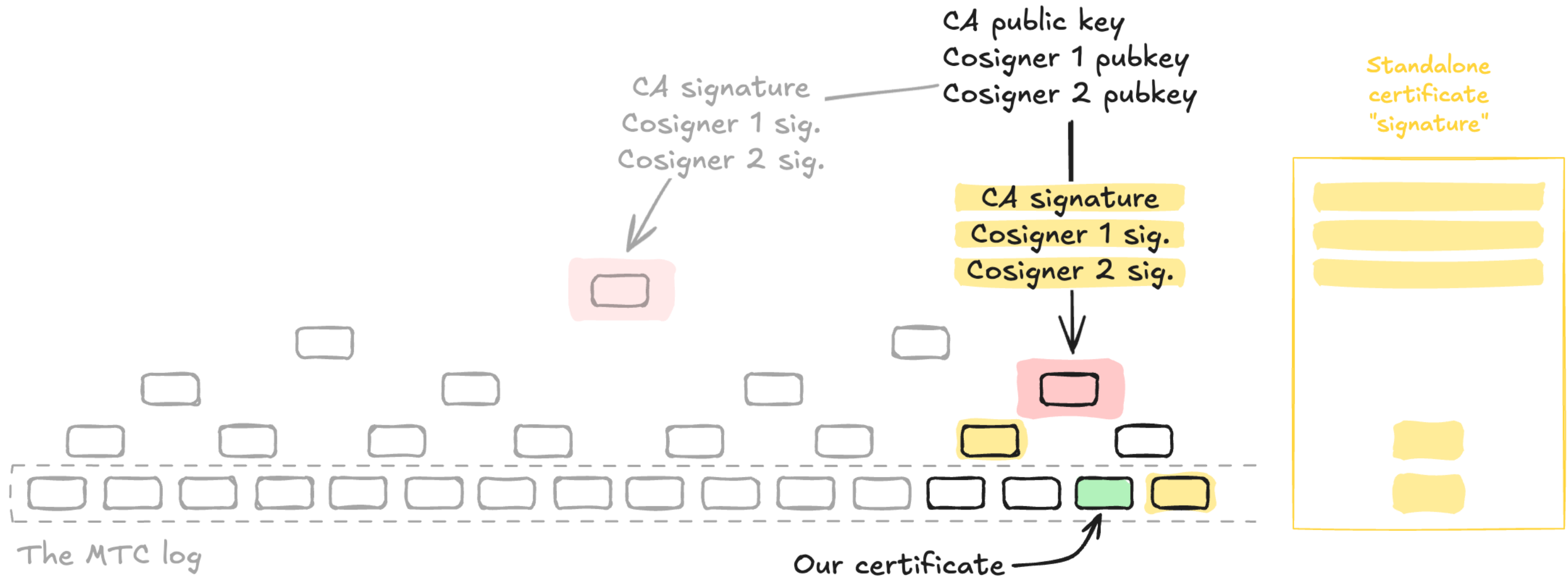


More precisely, they sign both the tree and a subtree

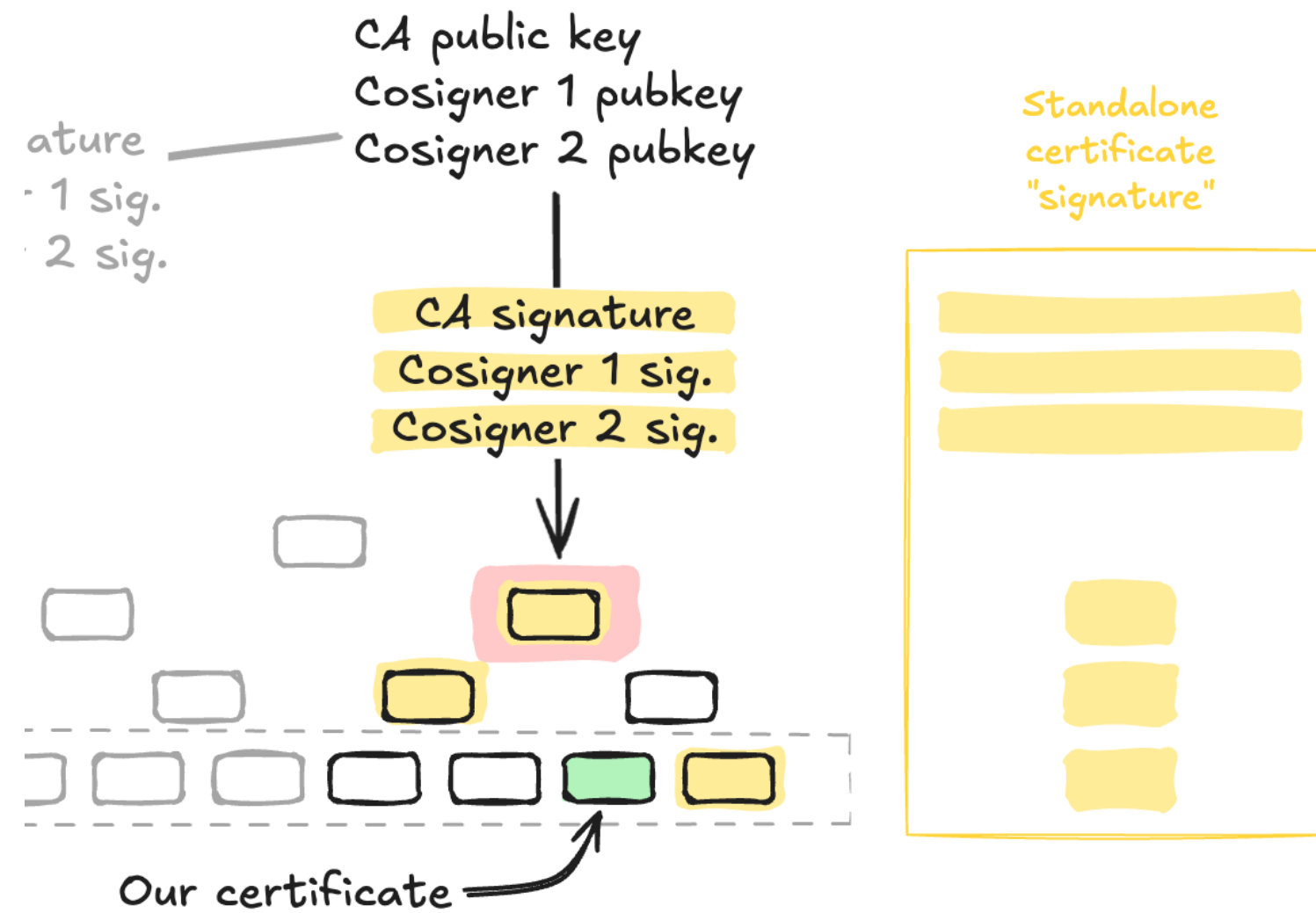
Ok, well, that sounds complicated.

**How could I possibly use it for my
microcontroller's binary signing?**

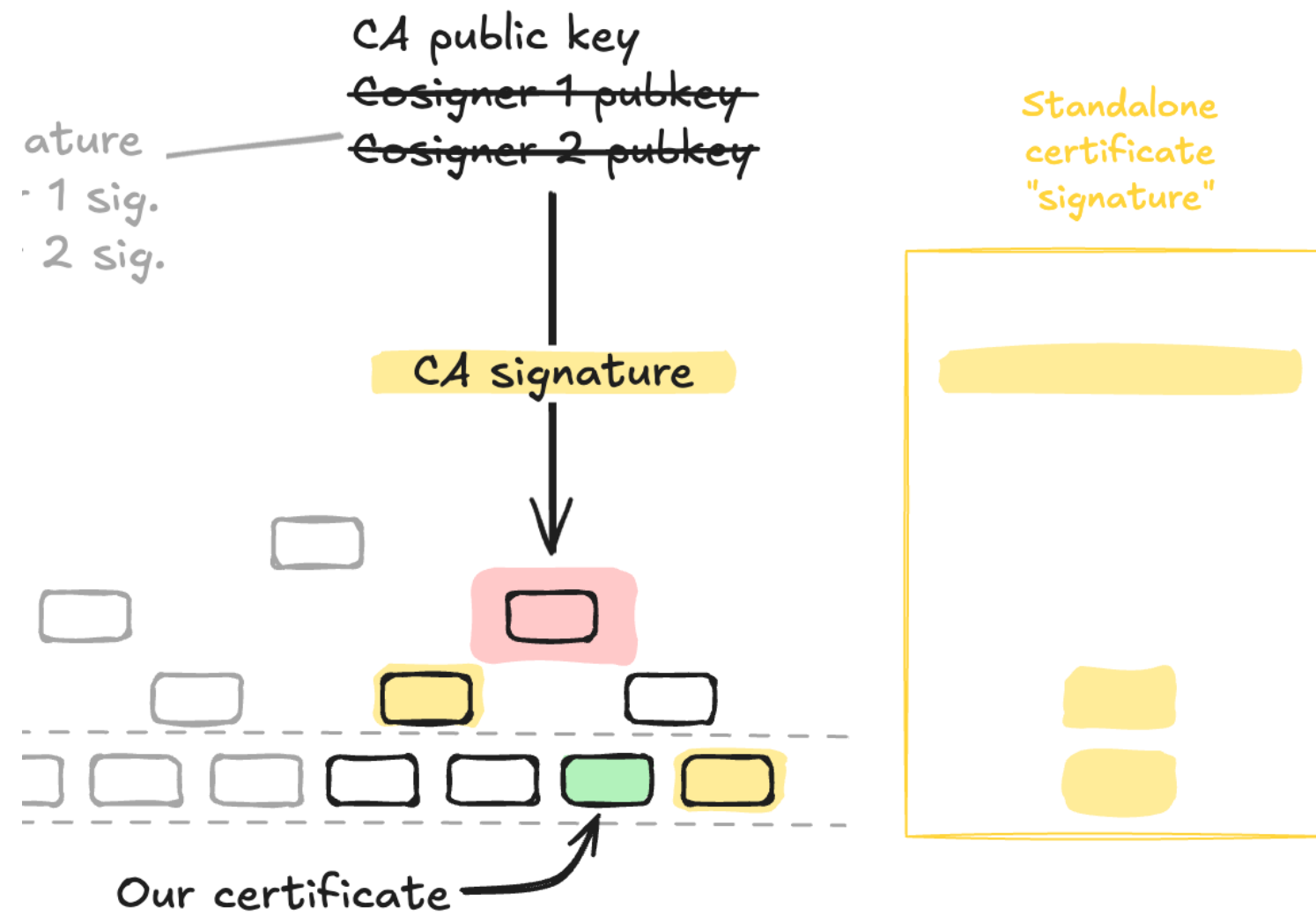
By the way my CA software can do one thing at a time, and is stateless.



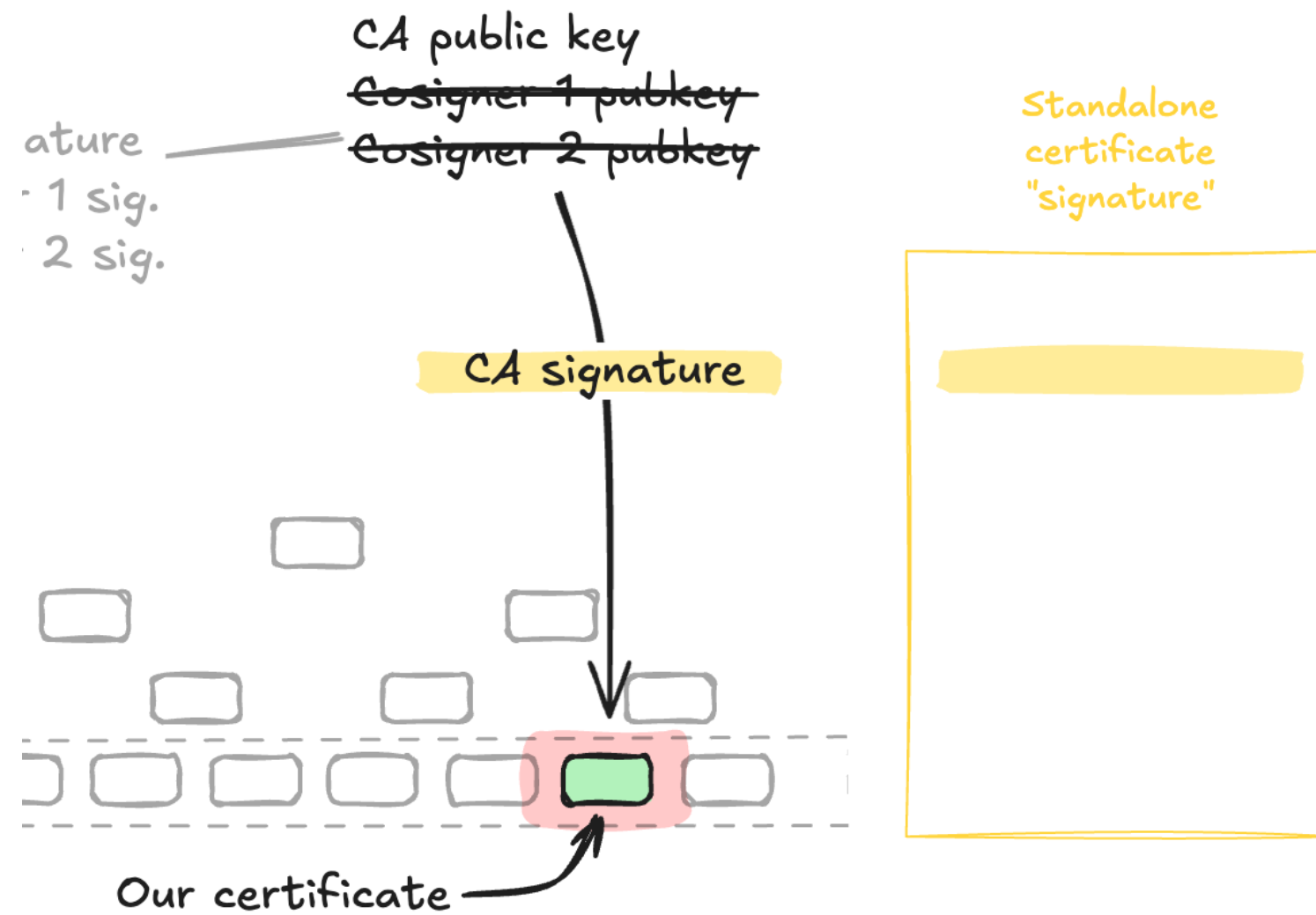
Step 1: take a standalone MTC, which includes signatures and inclusion path as a special signature field



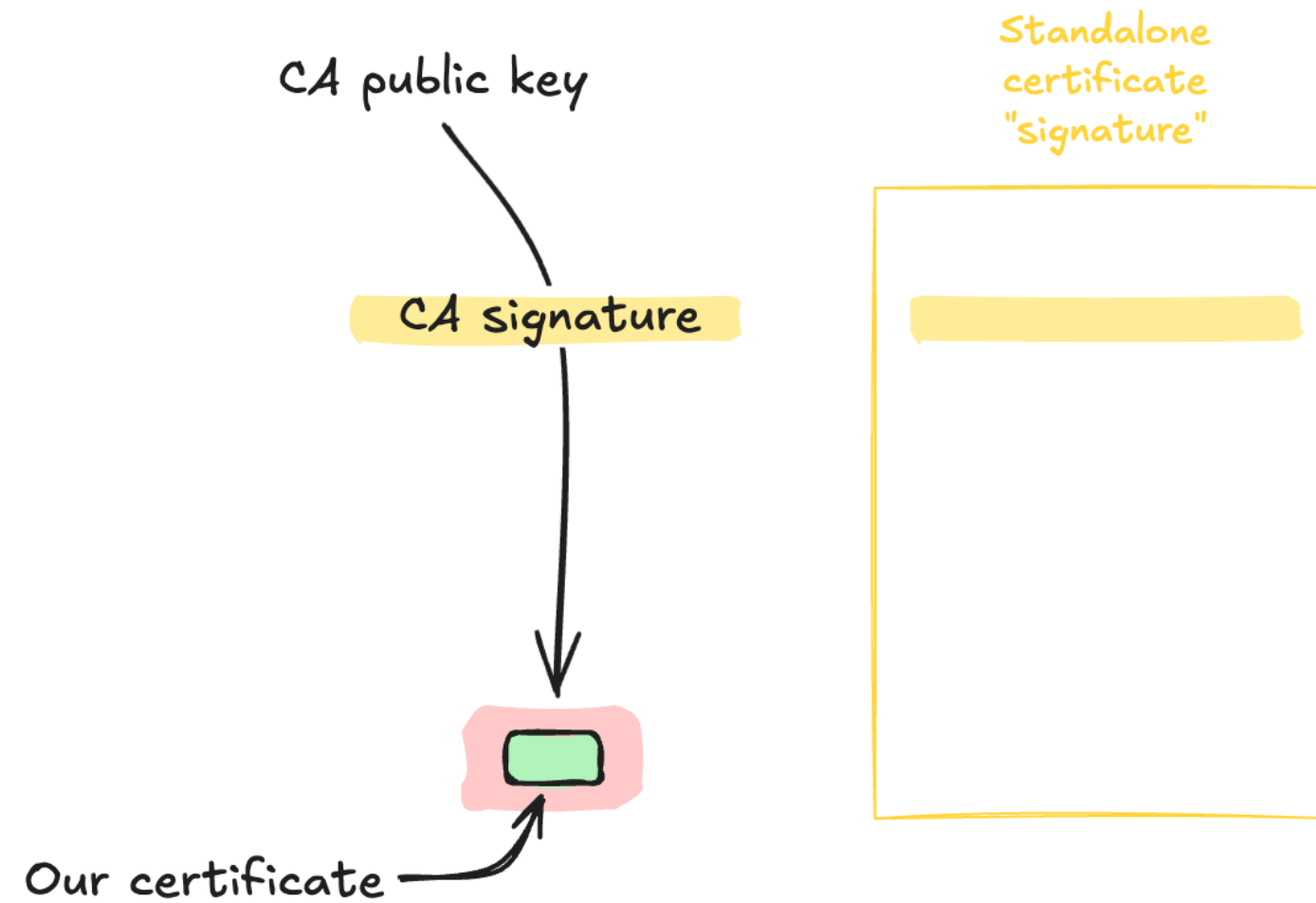
Let's ignore the rest of the log for now



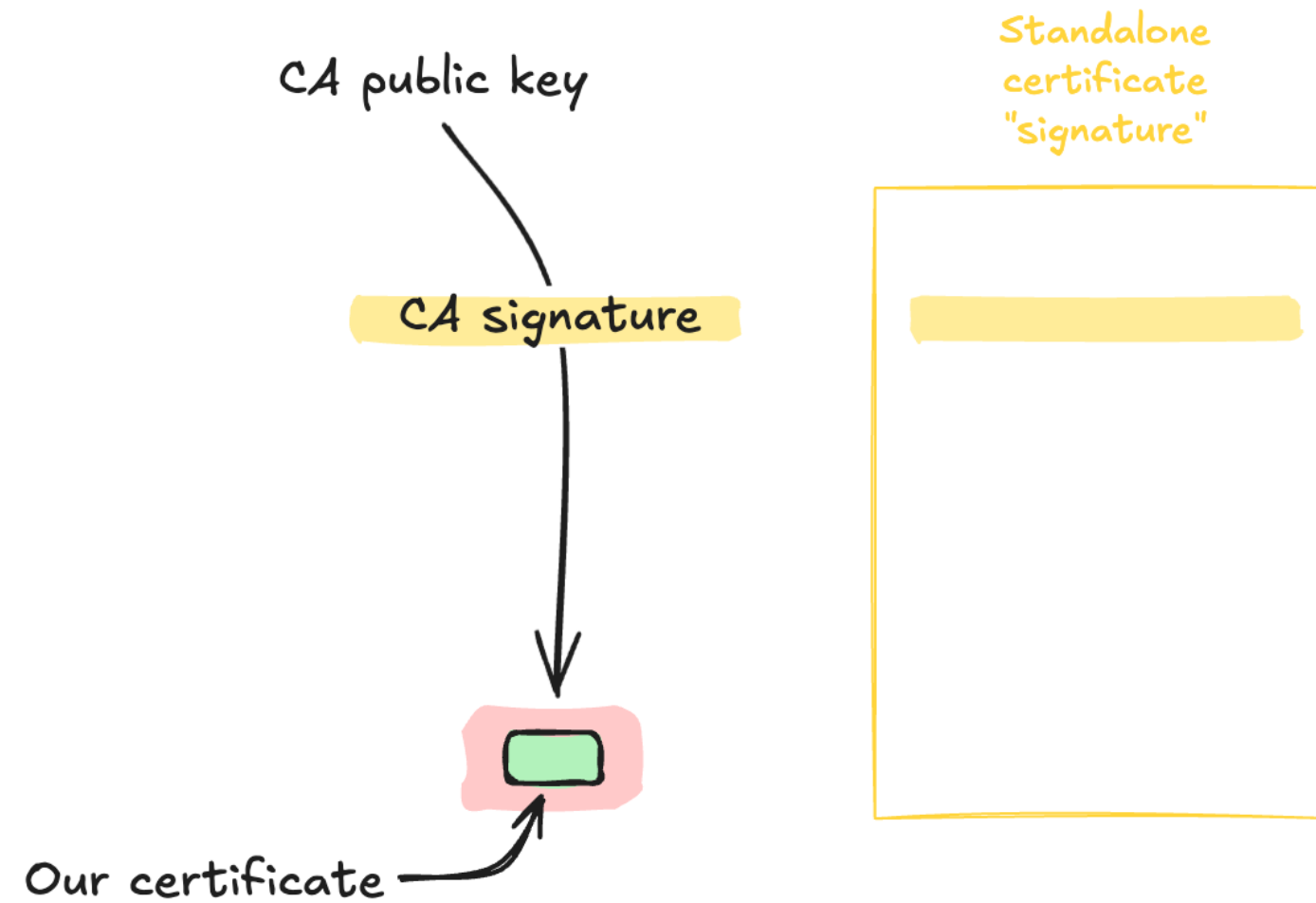
Step 2: drop the cosigners, you have one CA and no CT anyway



Step 3: make the subtree size 1



Step 4: forget there ever was a log (*the log index is the serial number, which has to be unique anyway*)

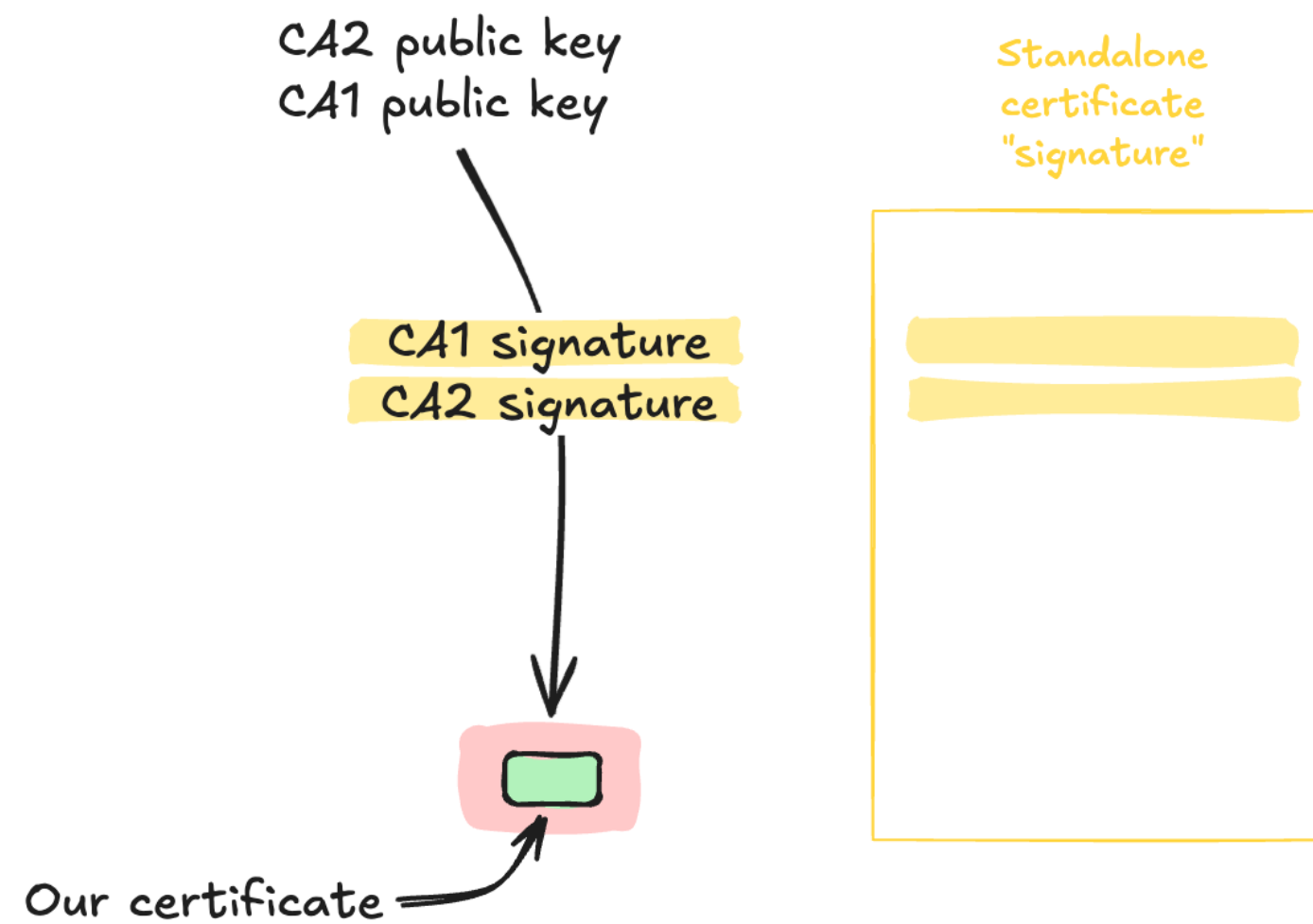


Just a weird encoding for a pure signature!

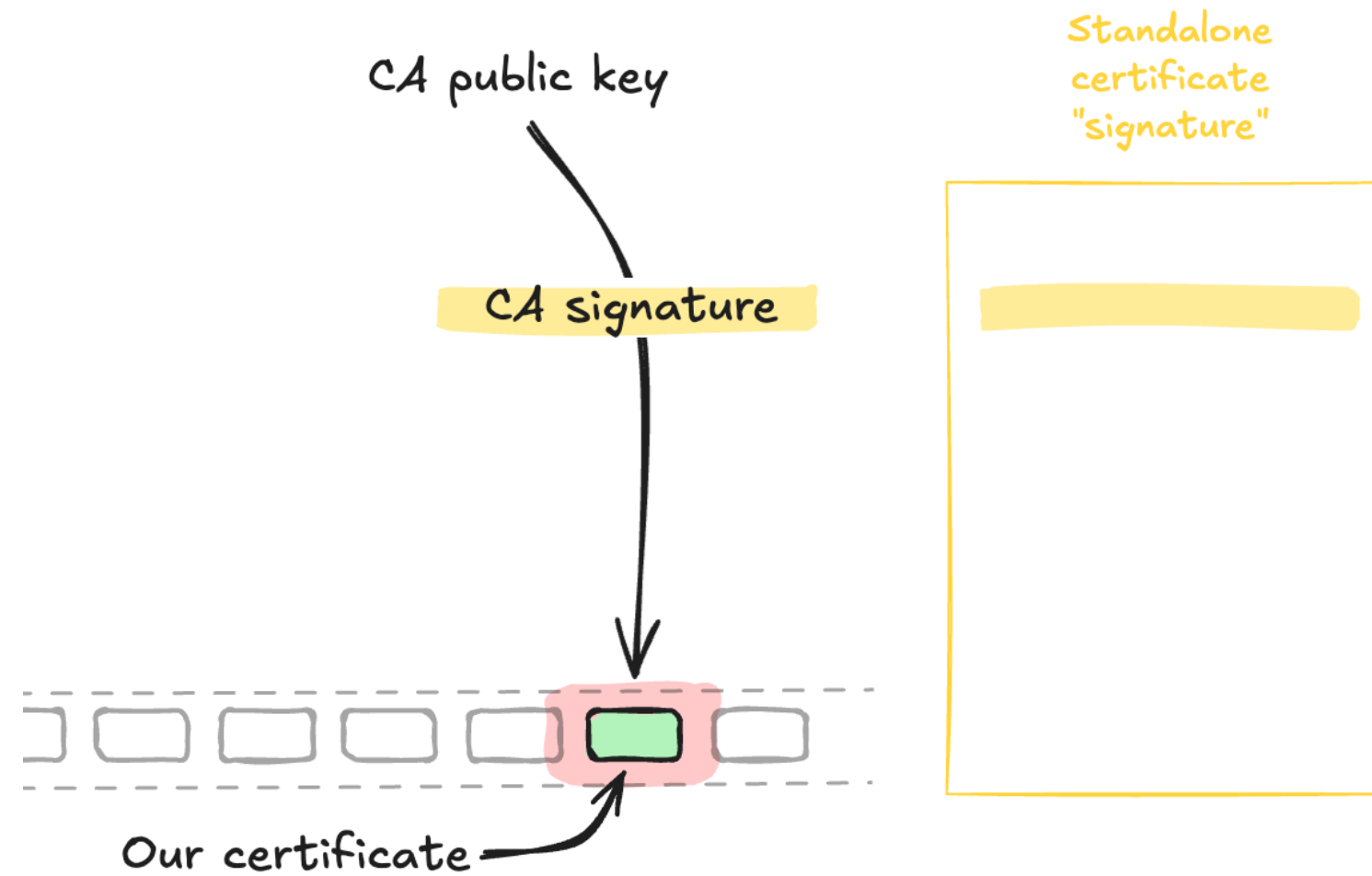
Ok, I believe you: MTC can fit wherever regular signed X.509 certificates fit, if you prune it down.
But *why* would I use MTC then?



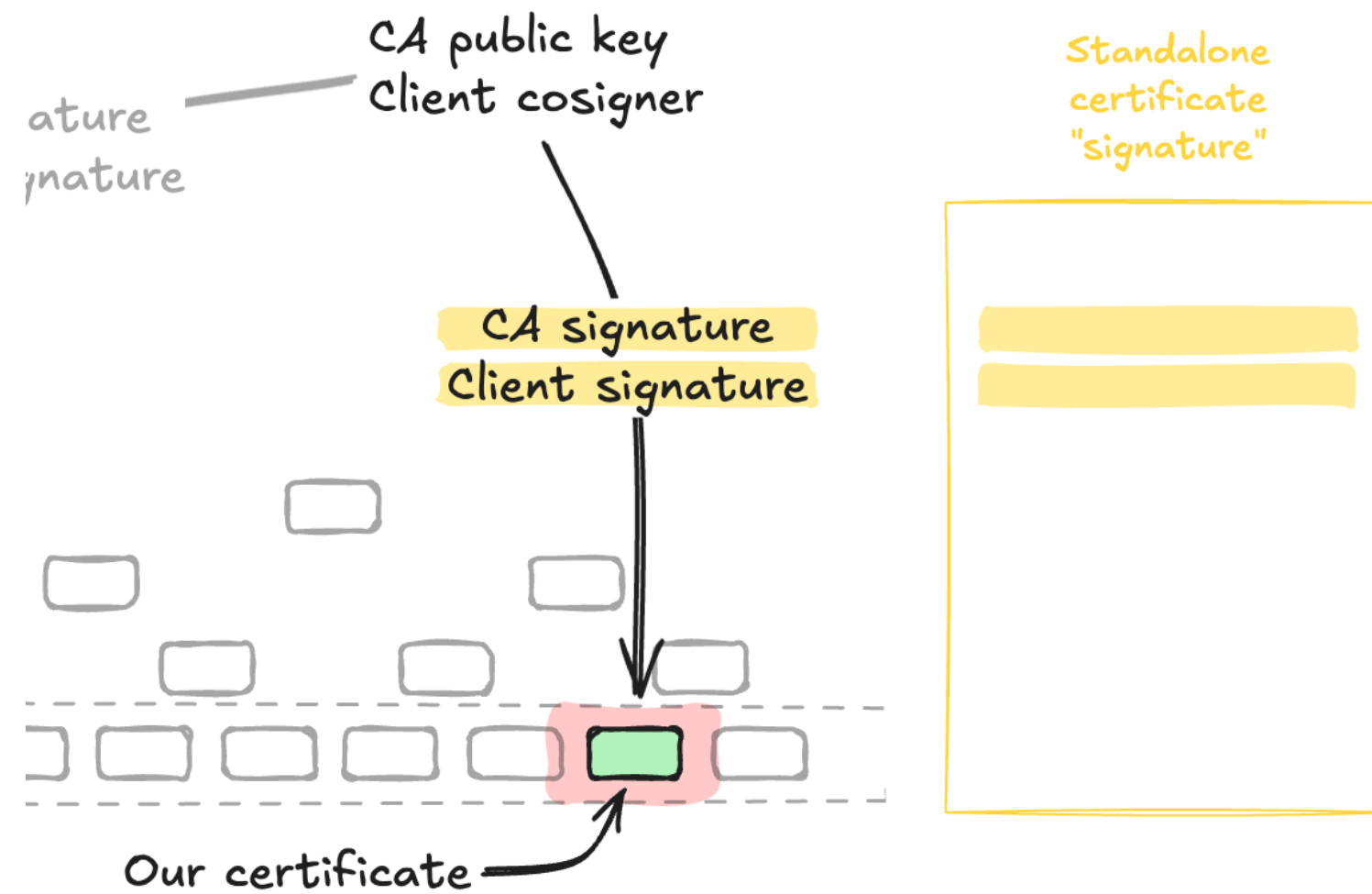
**Because you can
grow it back up!**



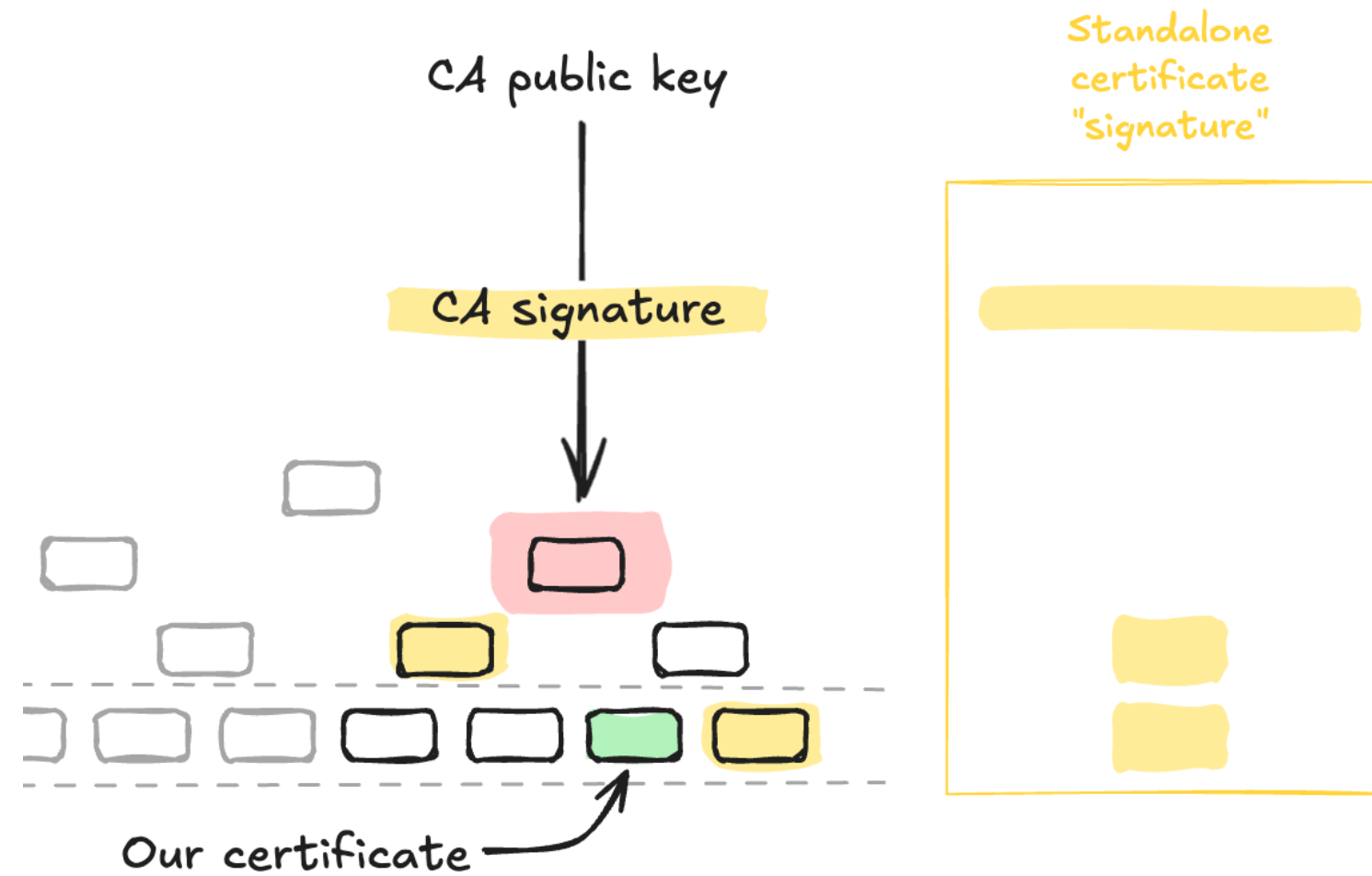
Get key rotation for free (if you don't care about size)



Add an audit log, and get WebPKI-style PQ downgrade protection



Let a client witness the log with off-the-shelf cosigner software and verifiers (**CT for private PKIs!**)



Do batch signing with a slow HSM (or, of course, roll out landmarks)

“Subtree size 1” standalone MTC work everywhere

Does that mean we *could* use MTC as the only PQ PKI code path, both on the CA and on the verifier side?

Does that mean that we *should*?

Maybe not, but hopefully this exercise gives you the idea of how standalone MTC works.

It's not scary! It's just a cryptid (for now) signature encoding! 🌱