



Deploying MTCs at CDN Scale

Luke Valenta

Senior Research Engineer

Cloudflare

Transparency.dev Summit 2026

About Cloudflare

We run a **global network** spanning 335 cities in over 125 countries.

Started of as a **CDN** and **DDoS mitigation** company, we now offer many more services, including

- **1.1.1.1**, public DNS resolver
- **Workers**, developer platform
- **Zero trust**, to protect corporate networks

We serve more than **20% of all websites** and process 115 million HTTP requests per second.
>45% of Fortune 500 are paying customers.



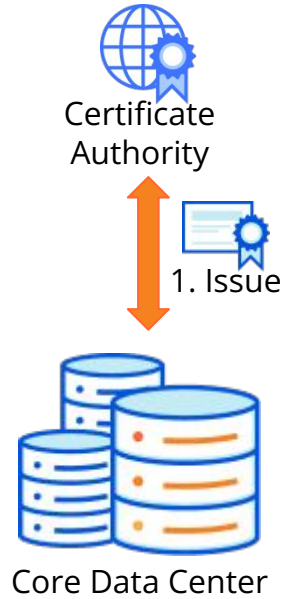
This talk

How MTC is going to let us PQ all those connections (for everyone, for free).

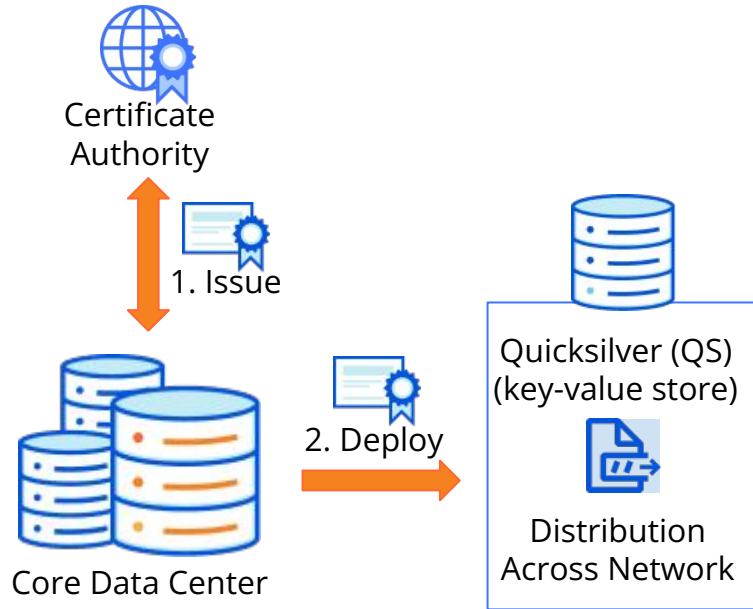
We'll use some tricks to make it less expensive.

WARNING: lots of hand-wavy napkin math

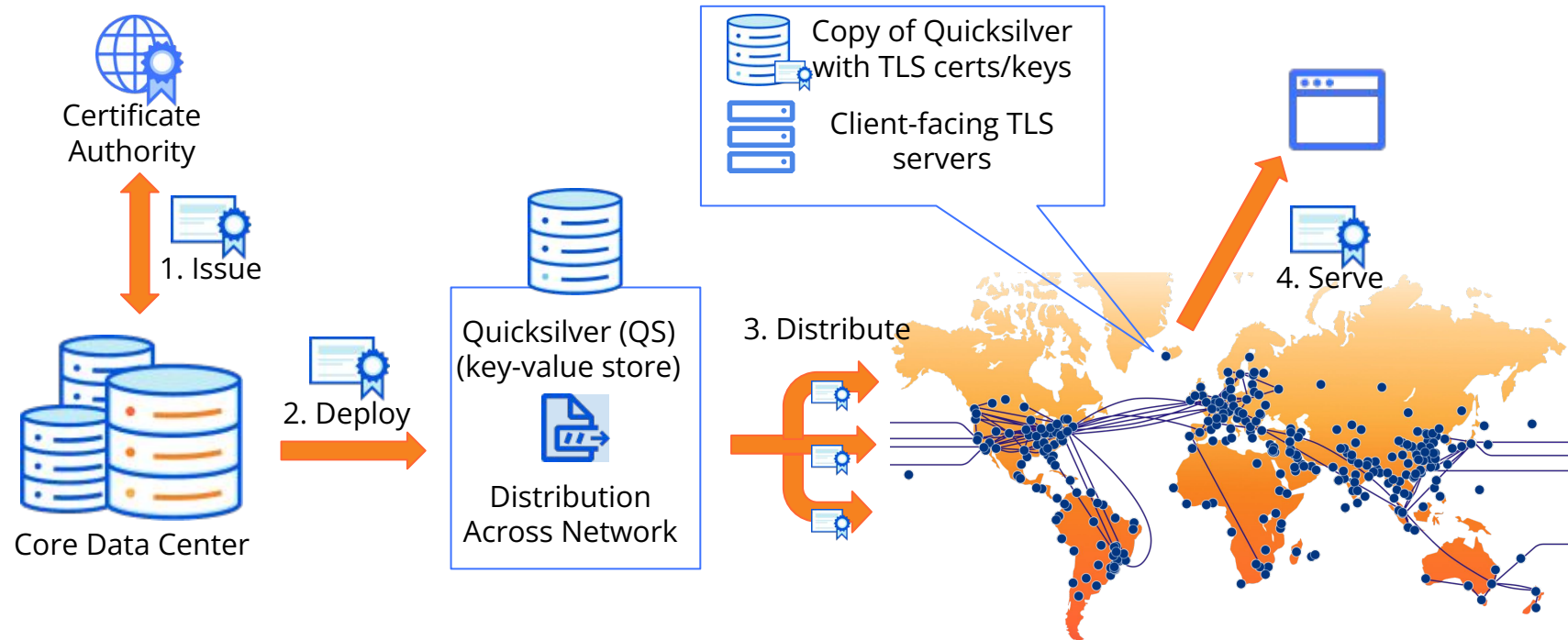
Cloudflare TLS cert lifecycle



Cloudflare TLS cert lifecycle



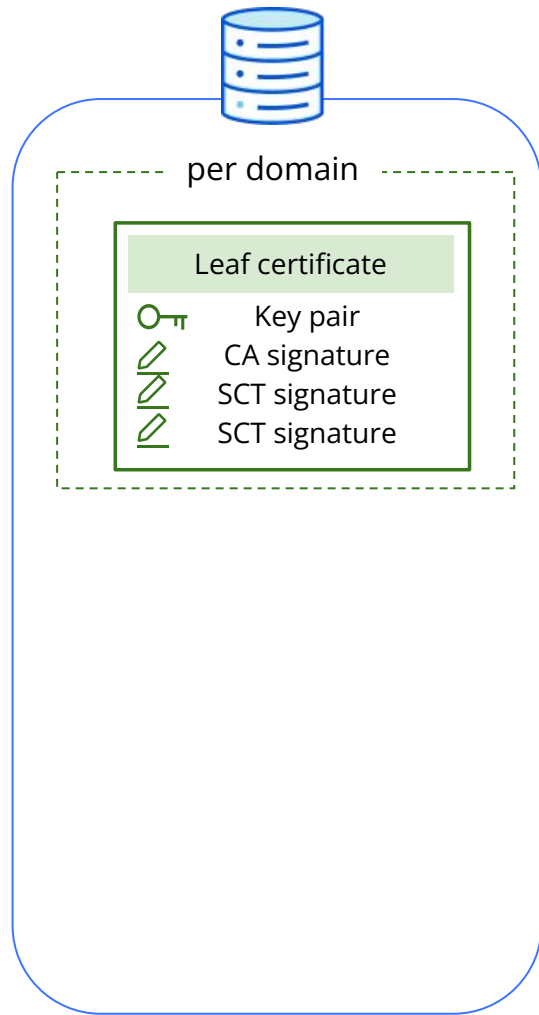
Cloudflare TLS cert lifecycle



Ballpark costs (classical)

Numbers for napkin math

- 100M domains
- Per domain
 - 1 P-256 key pair
 - 1 RSA-2048 CA signature
 - 2 P-256 SCT signatures
 - *Ignore other overheads (key wrapping, TbsCertificate, cert chain, ...)*



Ballpark costs (classical)

Numbers for napkin math

- 100M domains
- Per domain
 - 1 P-256 key pair
 - 1 RSA-2048 CA signature
 - 2 P-256 SCT signatures
 - *Ignore other overheads (key wrapping, TbsCertificate, cert chain, ...)*

Size in QS: 👍 **45 GB** $10^8 * (64+256+2*64)$



per domain

Leaf certificate



Key pair



CA signature



SCT signature



SCT signature

Ballpark costs (classical)

Numbers for napkin math

- 100M domains
- Per domain
 - 1 P-256 key pair
 - 1 RSA-2048 CA signature
 - 2 P-256 SCT signatures
 - *Ignore other overheads (key wrapping, TbsCertificate, cert chain, ...)*

Size in QS: 👍 **45 GB** $10^8 * (64+256+2*64)$

With ML-DSA-44: 👎 **857 GB** $10^8 * (1312+3*2420)$



per domain

Leaf certificate



Key pair



CA signature



SCT signature

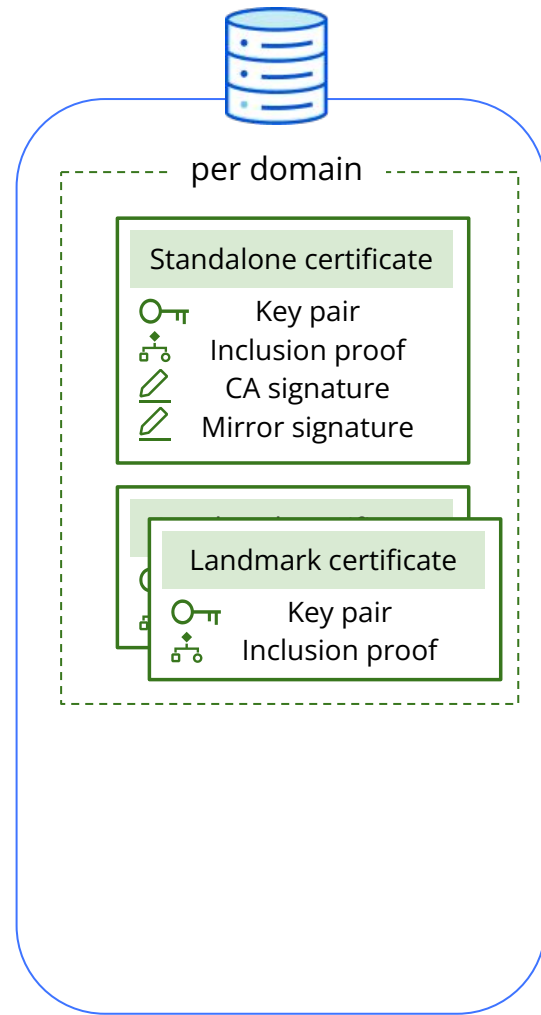


SCT signature

What changes with PQ MTC?

Have to deploy multiple certs per domain

- 1 *standalone* cert for old clients, new domains, etc.
- 2 *landmark-relative* (LR) certs for up-to-date clients.
 - Only usable after landmarks are distributed to clients, so keep an old one around.



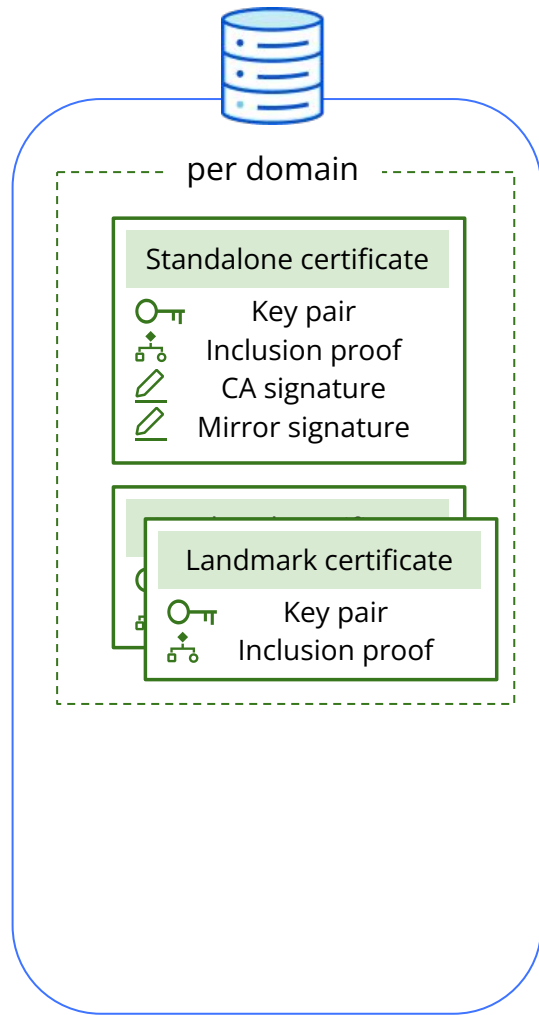
What changes with PQ MTC?

Have to deploy multiple certs per domain

- 1 *standalone* cert for old clients, new domains, etc.
- 2 *landmark-relative* (LR) certs for up-to-date clients.
 - Only usable after landmarks are distributed to clients, so keep an old one around.

ML-DSA-44 signatures/public keys are big

- 32-byte private keys (seed format)
- 2420-byte signatures (38x increase over P-256)
- 1312-byte public keys (20x increase over P-256)



What changes with PQ MTC?

Have to deploy multiple certs per domain

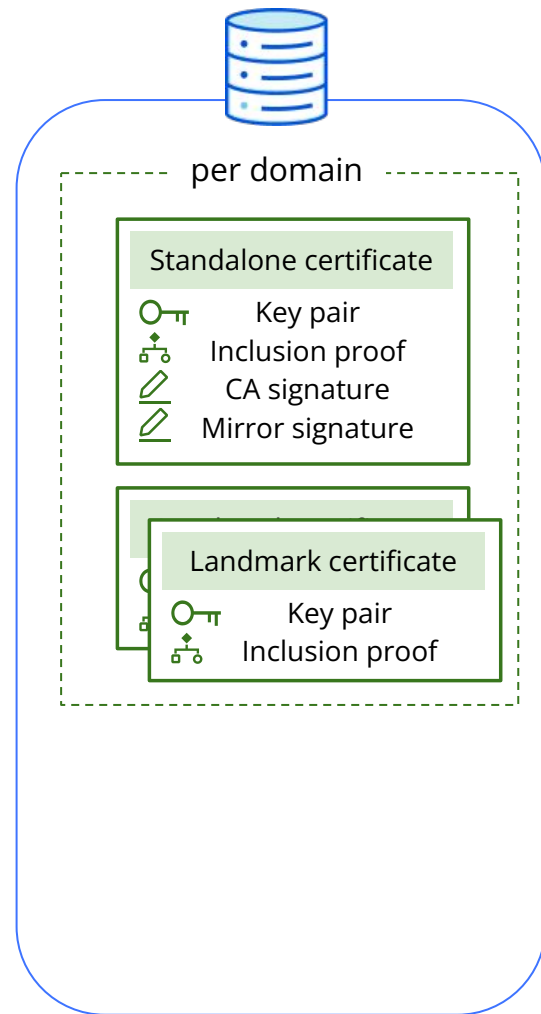
- 1 *standalone* cert for old clients, new domains, etc.
- 2 *landmark-relative* (LR) certs for up-to-date clients.
 - Only usable after landmarks are distributed to clients, so keep an old one around.

ML-DSA-44 signatures/public keys are big

- 32-byte private keys (seed format)
- 2420-byte signatures (38x increase over P-256)
- 1312-byte public keys (20x increase over P-256)

Batch signing

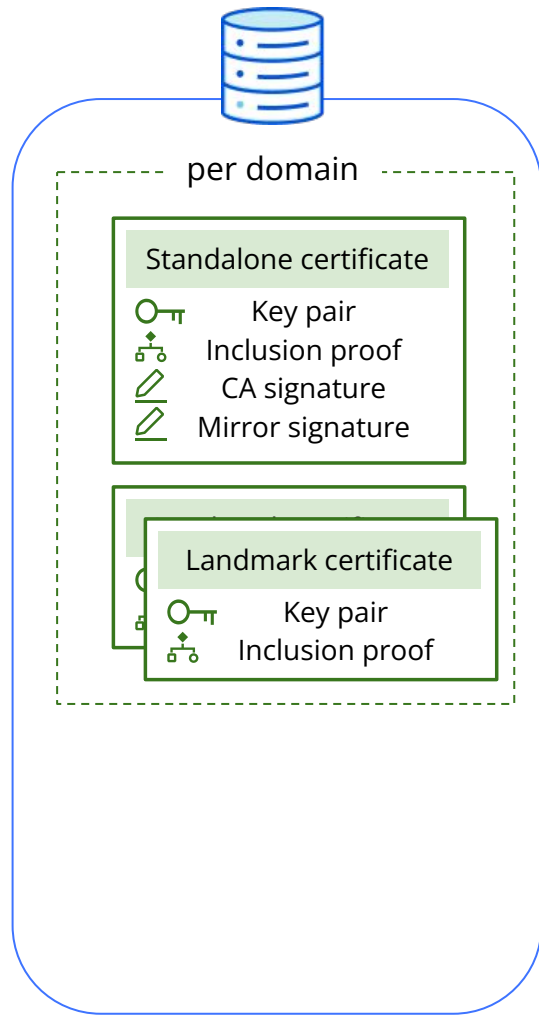
- One set of cosignatures covers many certs
- Unique inclusion proof per cert



Ballpark costs (naïve PQ MTC)

Numbers for napkin math

- 100M domains
- Single CA issuance log
- MTC checkpoints every 1s, landmarks every 1h
- 7-day max cert lifetime, renew every 3.5 days
- Per domain
 - 2 ML-DSA-44 cosignatures
 - 1 standalone incl. proof (256 B)
 - 3 ML-DSA-44 key pairs
 - 2 landmark incl. proofs (600 B)
 - *Still ignore other overheads*

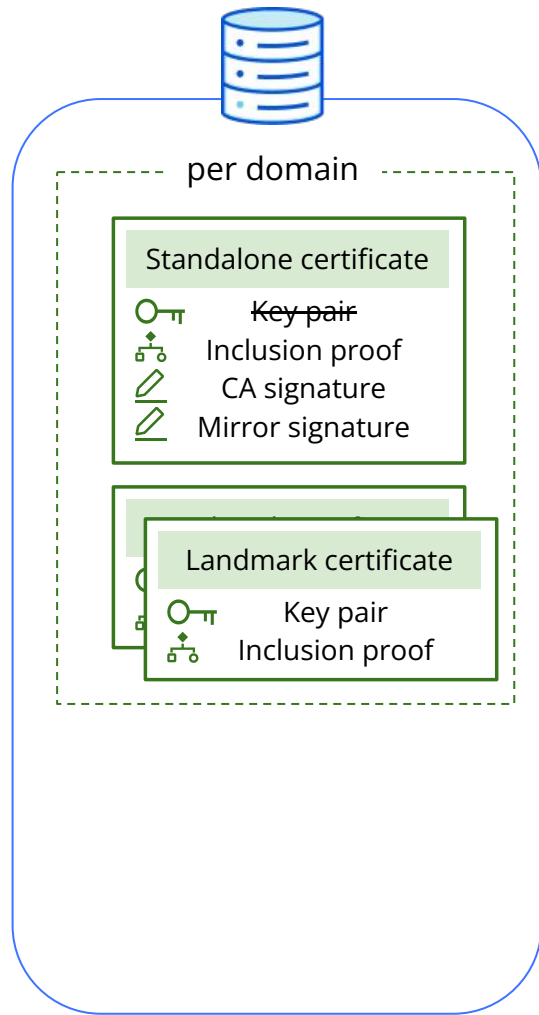


Ballpark costs (naïve PQ MTC)

Numbers for napkin math

- 100M domains
- Single CA issuance log
- MTC checkpoints every 1s, landmarks every 1h
- 7-day max cert lifetime, renew every 3.5 days
- Per domain
 - 2 ML-DSA-44 cosignatures
 - 1 standalone incl. proof (256 B)
 - ~~2~~ 2 ML-DSA-44 key pairs*
 - 2 landmark incl. proofs (600 B)
 - *Still ignore other overheads*

* Only need 2 key pairs, since the standalone cert shares a key pair with one of the landmark certs.



Ballpark costs (naïve PQ MTC)

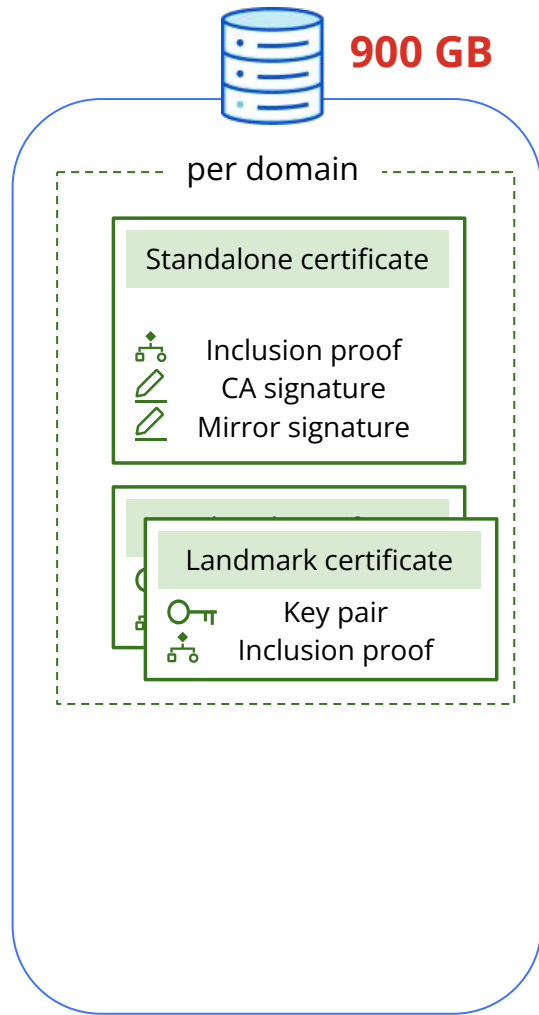
Numbers for napkin math

- 100M domains
- Single CA issuance log
- MTC checkpoints every 1s, landmarks every 1h
- 7-day max cert lifetime, renew every 3.5 days
- Per domain
 - 2 ML-DSA-44 cosignatures
 - 1 standalone incl. proof (256 B)
 - ~~2~~ 2 ML-DSA-44 key pairs*
 - 2 landmark incl. proofs (600 B)
 - *Still ignore other overheads*

* Only need 2 key pairs, since the standalone cert shares a key pair with one of the landmark certs.

Cost: 🖐️ **900 GB** $10^8 * (2*32+2*1312+2*2420+256+2*640)$

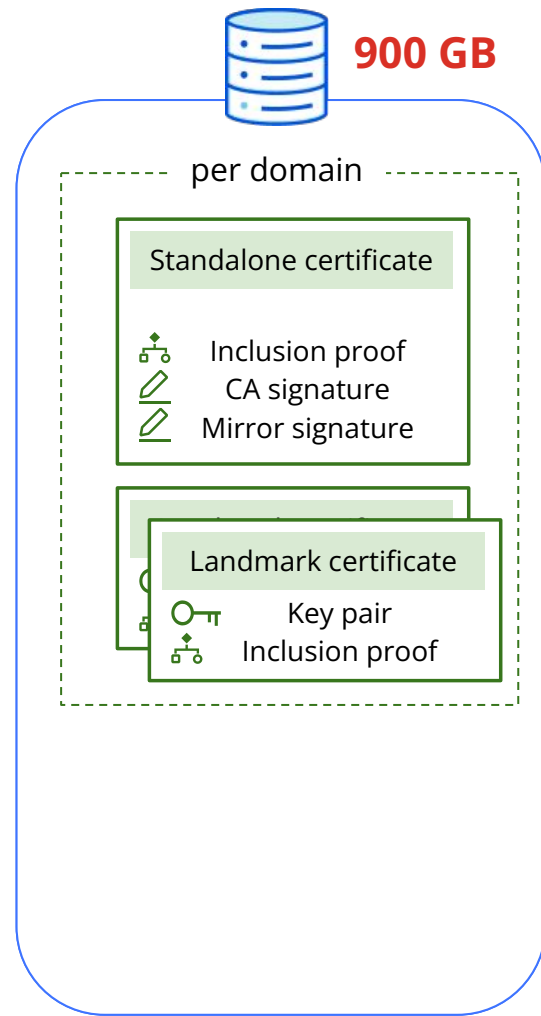
Can we do better?



Trick 1: Dedup subtree cosignatures

Observation

- A limited number of checkpoint subtrees (1.2M) cover all active certs.
- Many standalone certs will share the same signatures.



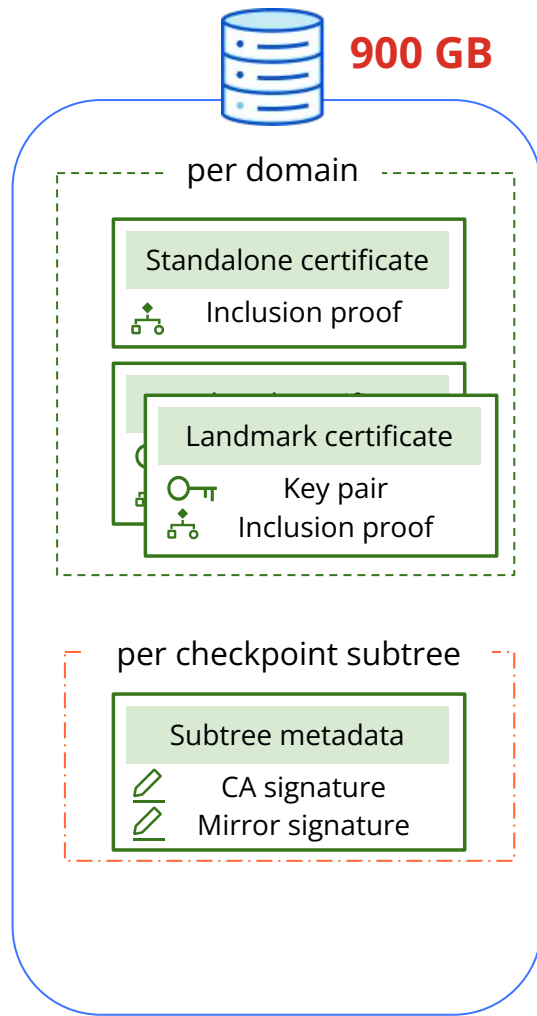
Trick 1: Dedup subtree cosignatures

Observation

- A limited number of checkpoint subtrees (1.2M) cover all active certs.
- Many standalone certs will share the same signatures.

Approach

- Don't store cosignature in per-domain records
 - Save **484 GB** $10^8 * 2 * 2420$
- Store copy of signatures checkpoint subtree
 - Add **6 GB** $2 * 604,800 * 2 * 2420$
- Reconstruct standalone certs during TLS handshake



Trick 1: Dedup subtree cosignatures

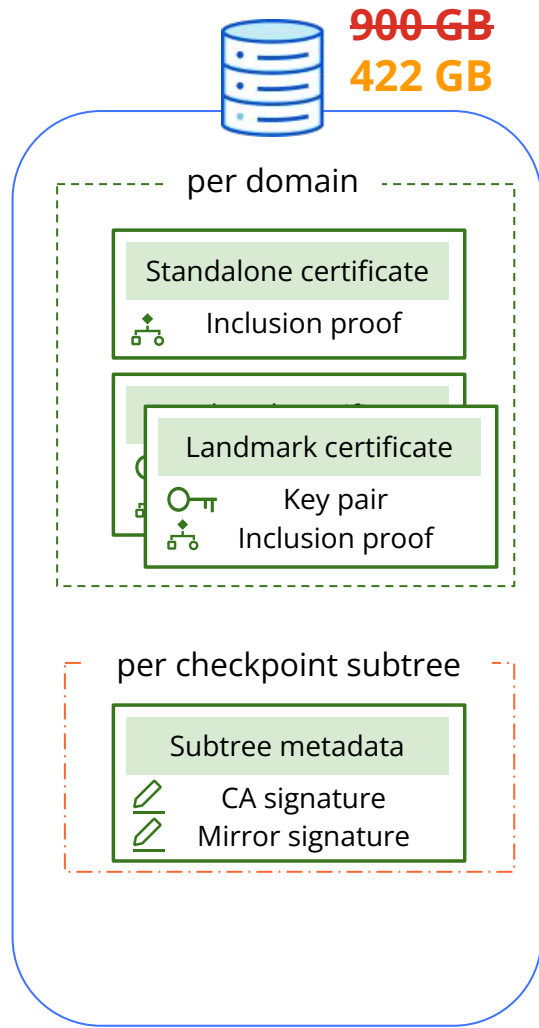
Observation

- A limited number of checkpoint subtrees (1.2M) cover all active certs.
- Many standalone certs will share the same signatures.

Approach

- Don't store cosignature in per-domain records
 - Save **484 GB** $10^8 * 2 * 2420$
- Store copy of signatures checkpoint subtree
 - Add **6 GB** $2 * 604,800 * 2 * 2420$
- Reconstruct standalone certs during TLS handshake

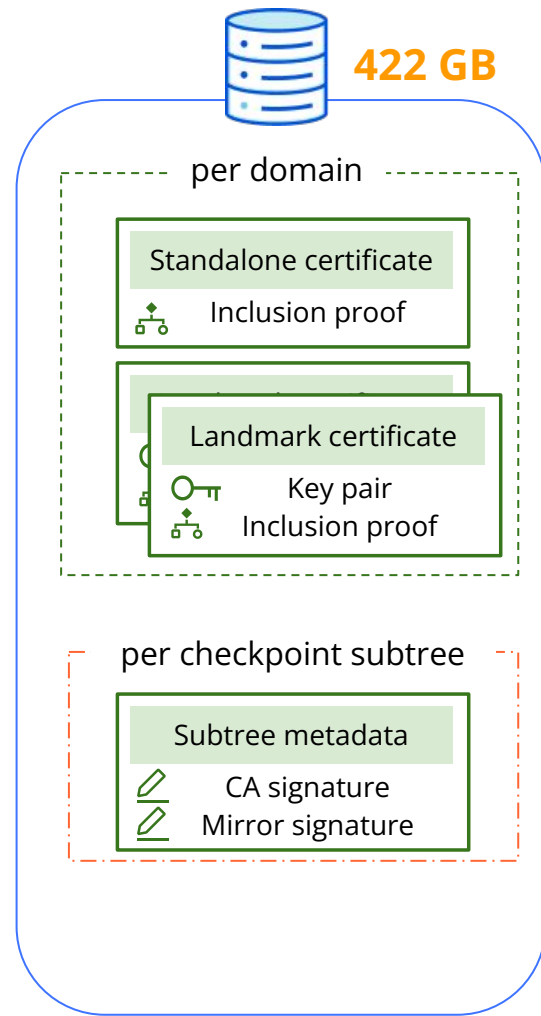
Cost drops over by half to **422 GB**.



Trick 2: Derive pubkeys from privkeys

Observation (not MTC-specific)

- Already need cert private key to terminate TLS handshake.
- Can derive public key from private key when needed.



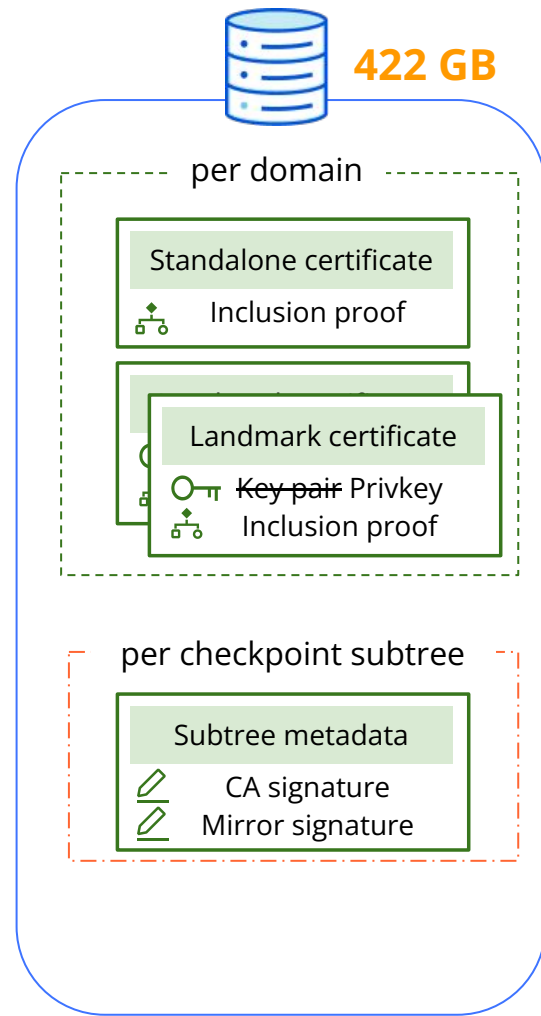
Trick 2: Derive pubkeys from privkeys

Observation (not MTC-specific)

- Already need cert private key to terminate TLS handshake.
- Can derive public key from private key when needed.

Approach

- Don't store public keys with certs in QS (only wrapped private keys)
 - Save **262 GB** $10^8 * 2 * 1312$
- Reconstruct public key during TLS handshake



Trick 2: Derive pubkeys from privkeys

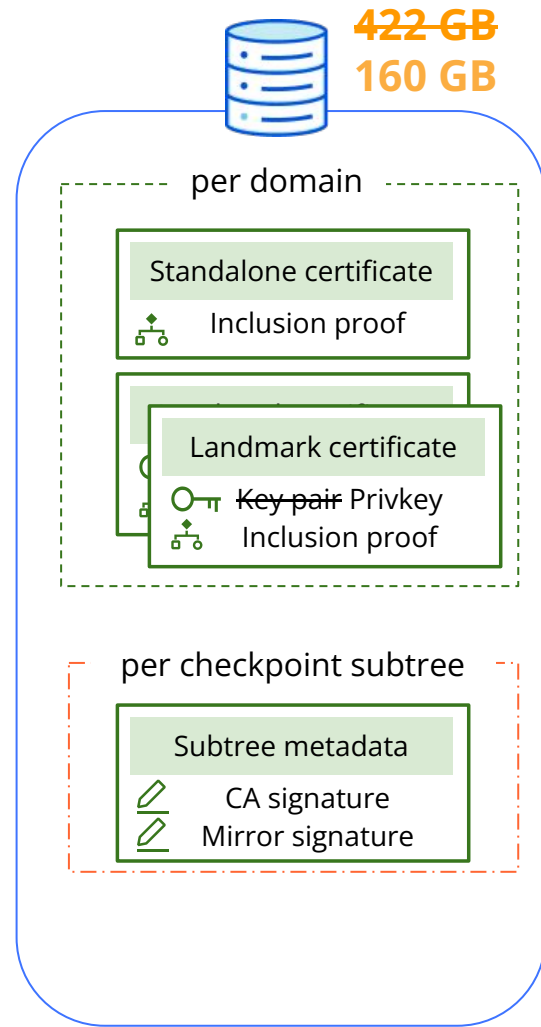
Observation (not MTC-specific)

- Already need cert private key to terminate TLS handshake.
- Can derive public key from private key when needed.

Approach

- Don't store public keys with certs in QS (only wrapped private keys)
 - Save **262 GB** $10^8 * 2 * 1312$
- Reconstruct public key during TLS handshake

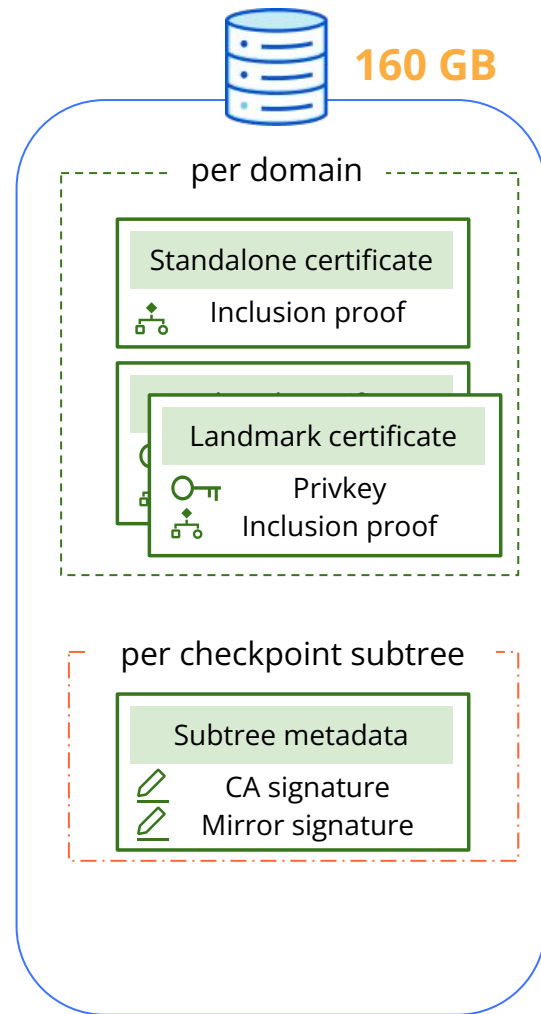
Down to **160 GB**... can we do better?



Trick 3: Dedup inclusion proofs

Observation

- A few (338) landmark subtrees cover all certs.
- Many landmark inclusion proofs will share hashes.



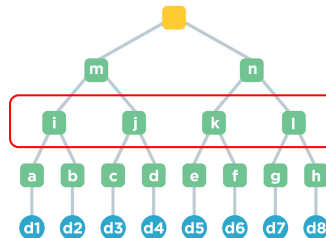
Trick 3: Dedup inclusion proofs

Observation

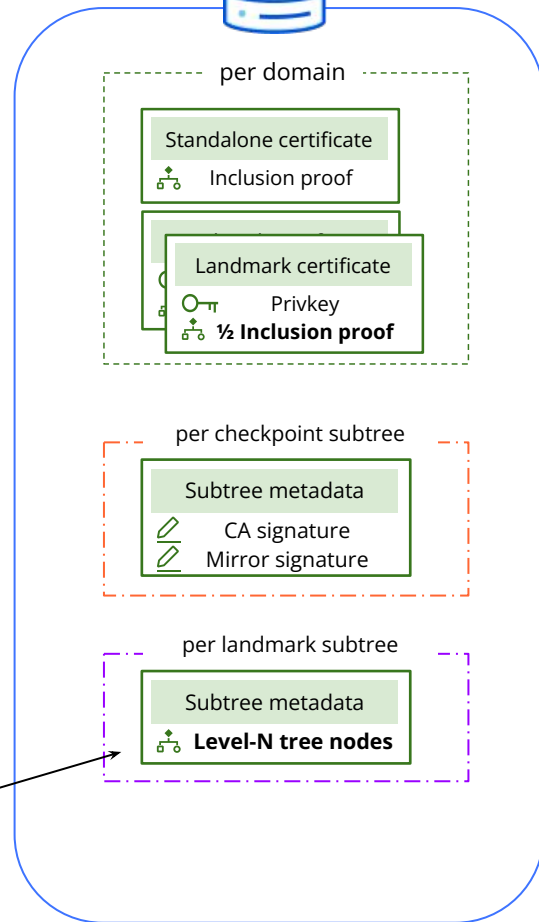
- A few (338) landmark subtrees cover all certs.
- Many landmark inclusion proofs will share hashes.

Approach

- Store nodes at level N of each landmark subtree once.
 - With $N=10$, add **6 MB** $338 * \text{ceil}(10^8/168/2^{10}) * 32$
- Store (height - N) fewer inclusion proof hashes per landmark cert
 - With $N=10$, save **64 GB** $10^8 * 2 * 10 * 32$
- Reconstruct inclusion proofs during TLS handshake



160 GB



Trick 3: Dedup inclusion proofs

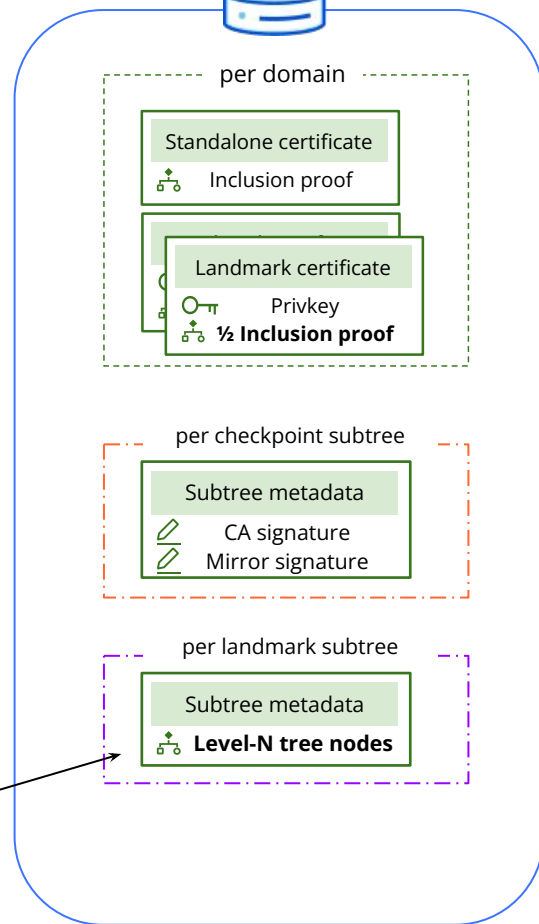
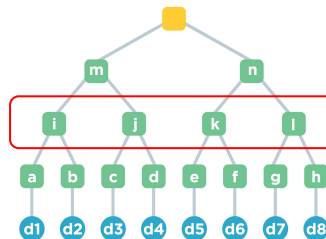
Observation

- A few (338) landmark subtrees cover all certs.
- Many landmark inclusion proofs will share hashes.

Approach

- Store nodes at level N of each landmark subtree once.
 - With $N=10$, add **6 MB** $338 * \text{ceil}(10^8/168/2^{10}) * 32$
- Store (height - N) fewer inclusion proof hashes per landmark cert
 - With $N=10$, save **64 GB** $10^8 * 2 * 10 * 32$
- Reconstruct inclusion proofs during TLS handshake

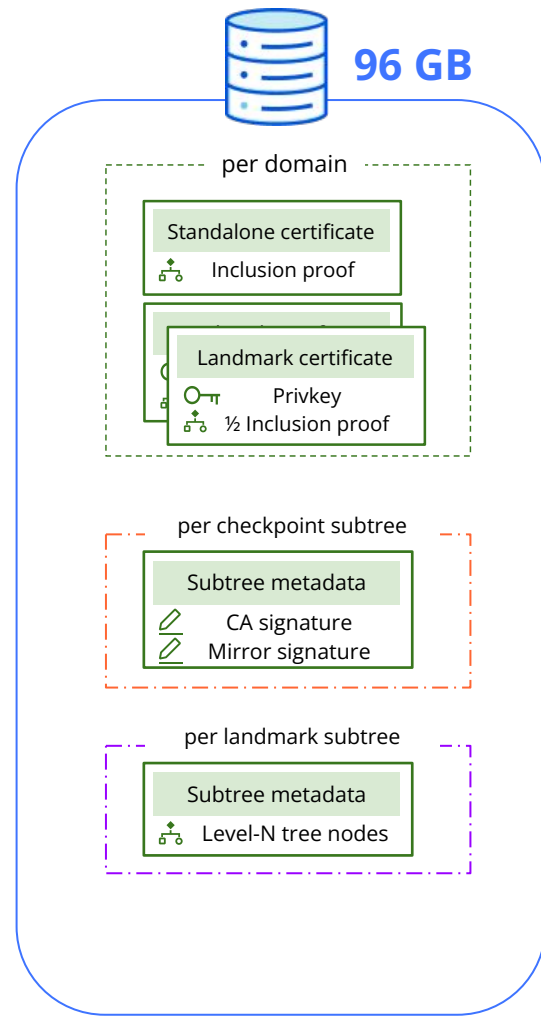
Almost halved again to **96 GB**. Keep going?



Trick 4: Landmark-rooted standalone

Observation

- Landmark + cosignatures = “landmark-rooted” standalone.
- Only 0.01 - 0.1% of certs require immediate issuance [1].



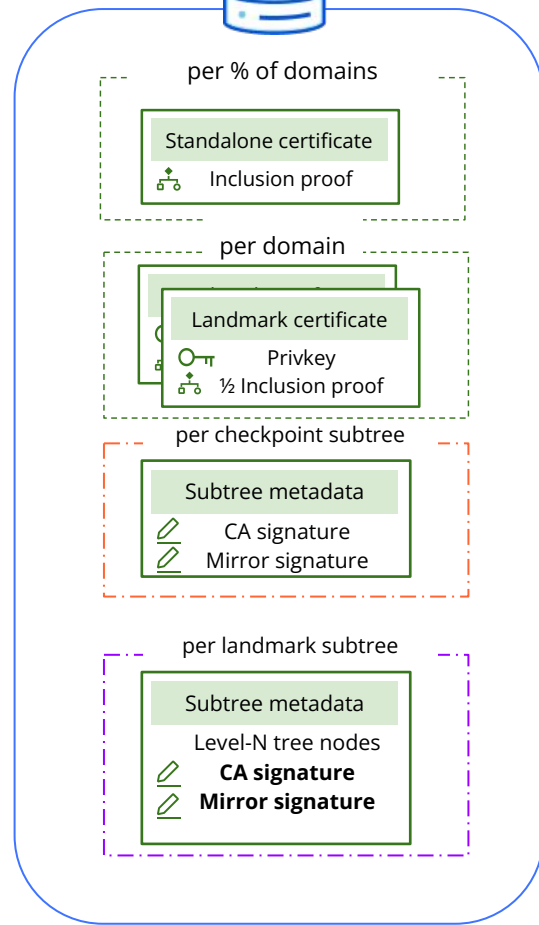
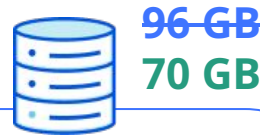
Trick 4: Landmark-rooted standalone

Observation

- Landmark + cosignatures = “landmark-rooted” standalone.
- Only 0.01 - 0.1% of certs require immediate issuance [1].

Approach

- Use sign-subtree API to get landmark subtree cosignatures
 - Add **1.6 MB** $338 * 2 * 2420$
- At issuance, only deploy standalone when immediate issuance is needed.
- One landmark interval after issuance, swap standalone cert with *landmark-rooted* standalone.
 - Save **26 GB** $10^8 * (1 - 1/1000/168) * 256$



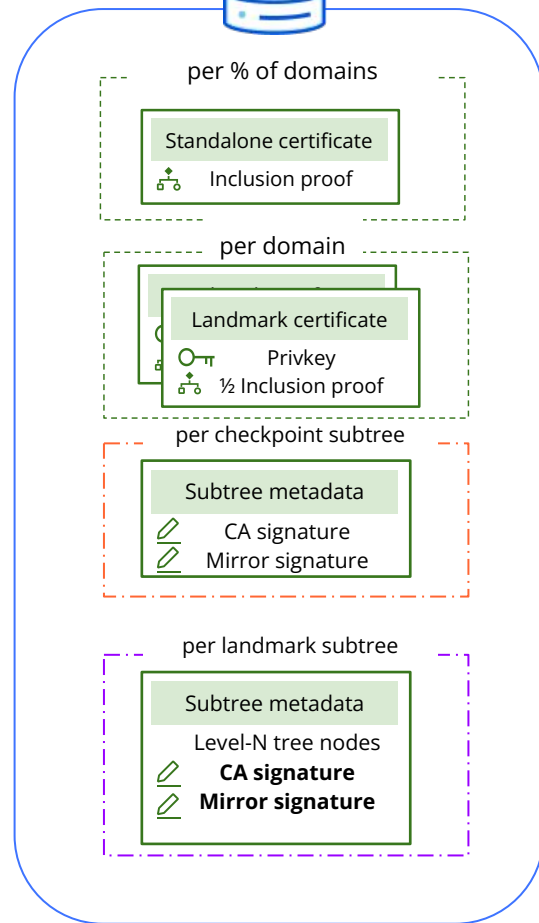
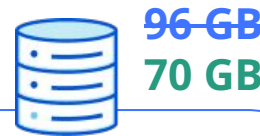
Trick 4: Landmark-rooted standalone

Observation

- Landmark + cosignatures = “landmark-rooted” standalone.
- Only 0.01 - 0.1% of certs require immediate issuance [1].

Approach

- Use sign-subtree API to get landmark subtree cosignatures
 - Add **1.6 MB** $338 * 2 * 2420$
- At issuance, only deploy standalone when immediate issuance is needed.
- One landmark interval after issuance, swap standalone cert with *landmark-rooted* standalone.
 - Save **26 GB** $10^8 * (1 - 1/1000/168) * 256$
- Con: Landmark inclusion proofs are bigger on the wire, but standalone is already the slow case
- Con: Can't fetch landmark cosignatures via ACME



Summing up

A naïve PQ MTC deployment would be expensive for us, and delay our ability to offer it for free to all Cloudflare customers.

With a few tricks, we can cut the storage impact of PQ MTC by over 10x (**900 GB** to **70GB** per data center).

Questions?

