Reliable Transparency and Revocation Mechanisms

Brendan McMillion, Devon O’Brien, Dennis Jackson

Problem Statement

- Certificate Transparency’s security is based on an assumption that log operators are perfect and independent, which has been broken repeatedly

- Monitoring CT is deeply unrealistic for “normal” people

- When CT detects a mis-issued certificate, there is no resolution!

What is Key Transparency?

Most transparency logs:

Key Transparency logs:

What is Key Transparency?

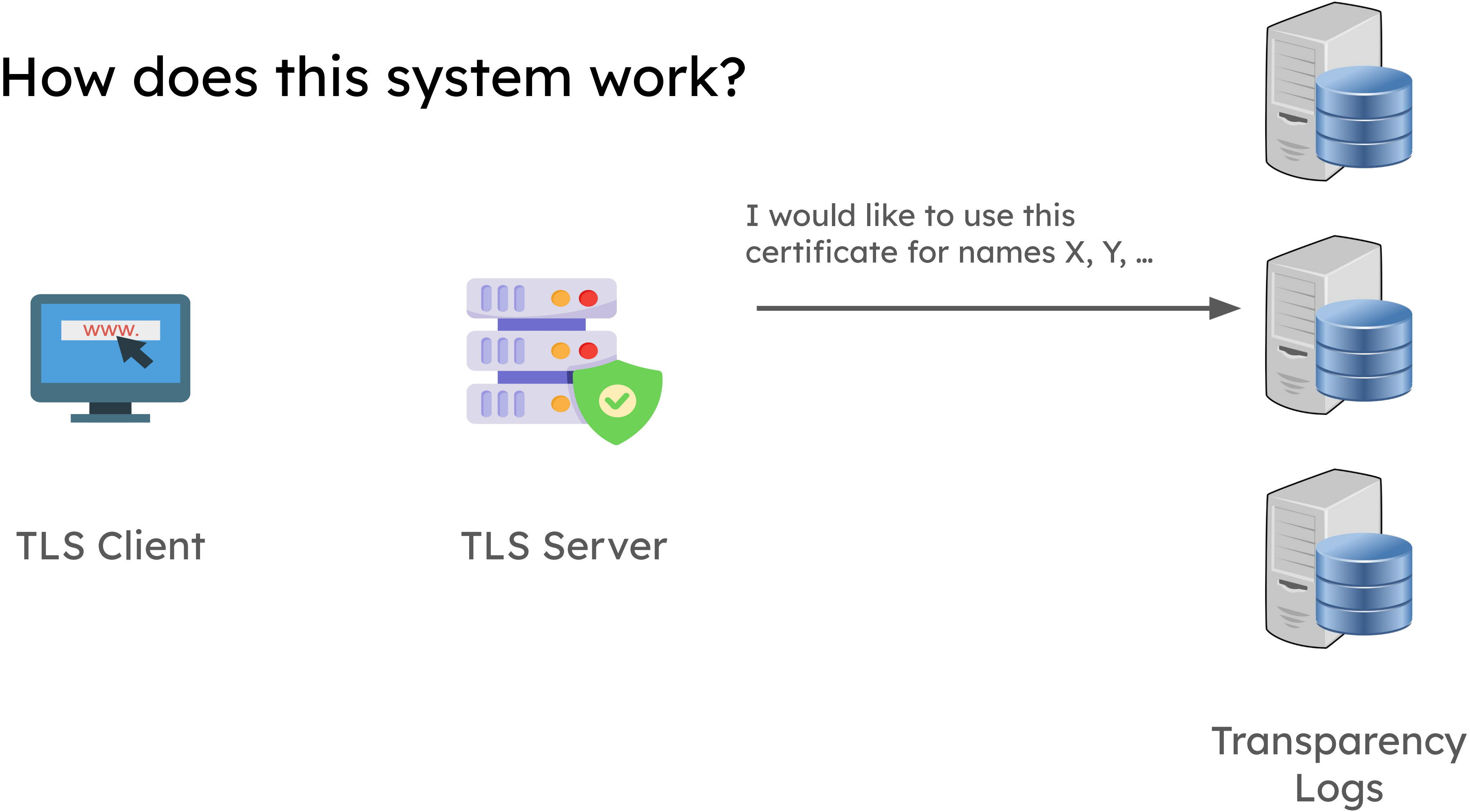
Most transparency logs:

Key Transparency logs:

Makes it easy to:

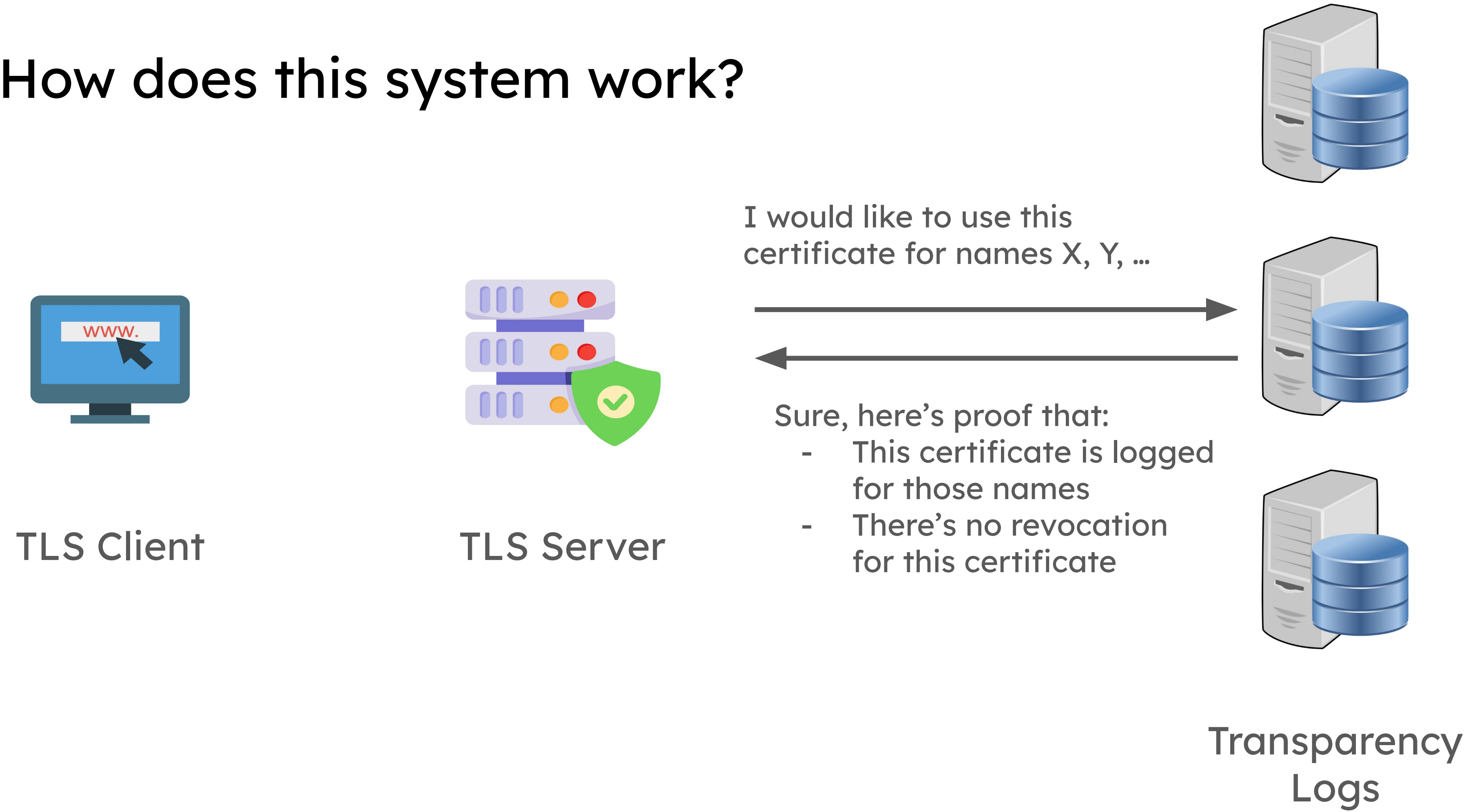
1. Find things that exist
2. **Find things that don’t exist**

How does this system work?



I would like to use this certificate for names X, Y, …

How does this system work?



TLS Client

TLS Server

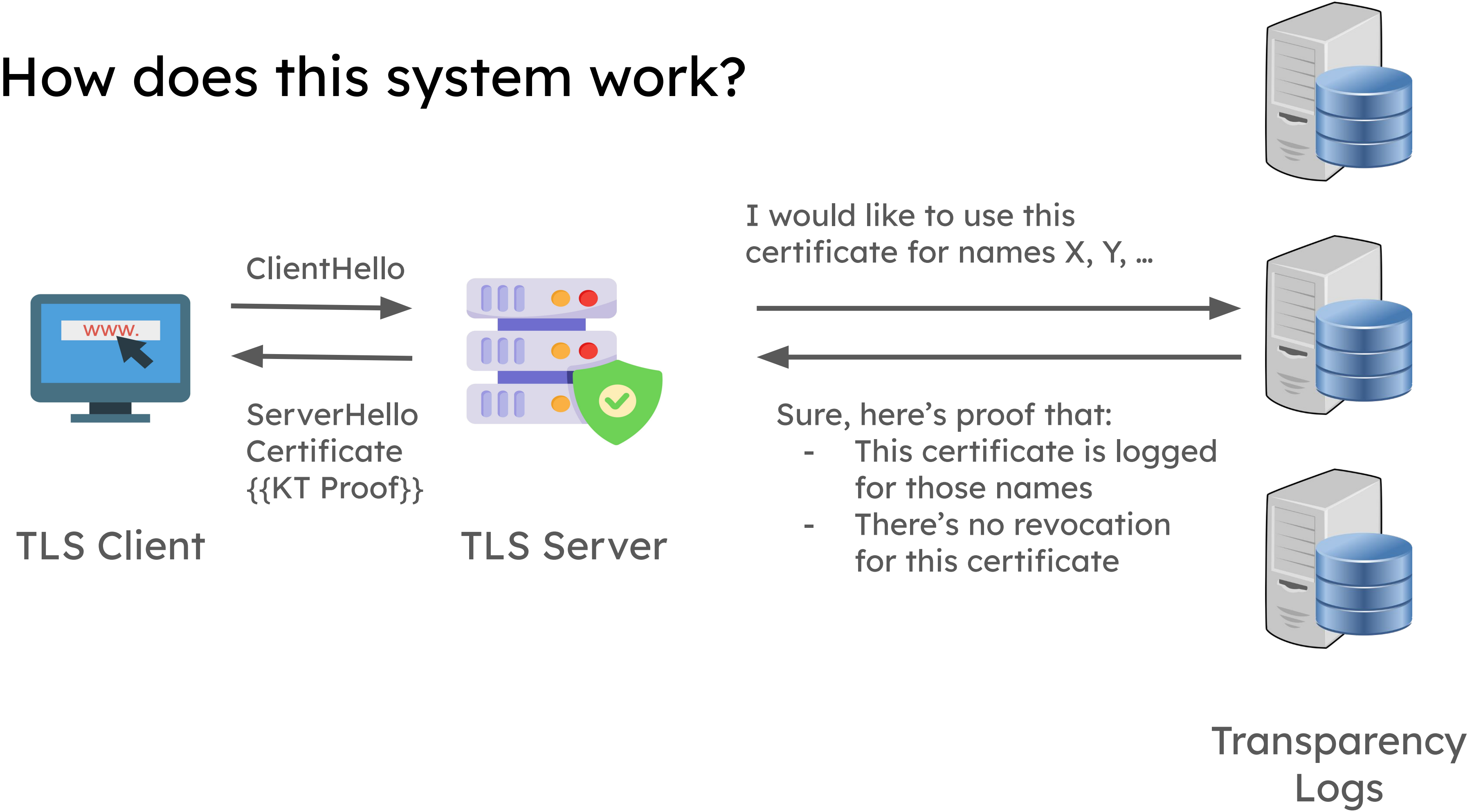
I would like to use this certificate for names X, Y, …

Sure, here’s proof that:

- This certificate is logged for those names
- There’s no revocation for this certificate

Transparency Logs

How does this system work?



ClientHello

ServerHello
Certificate
{{KT Proof}}

TLS Client

TLS Server

I would like to use this certificate for names X, Y, …

Sure, here’s proof that:

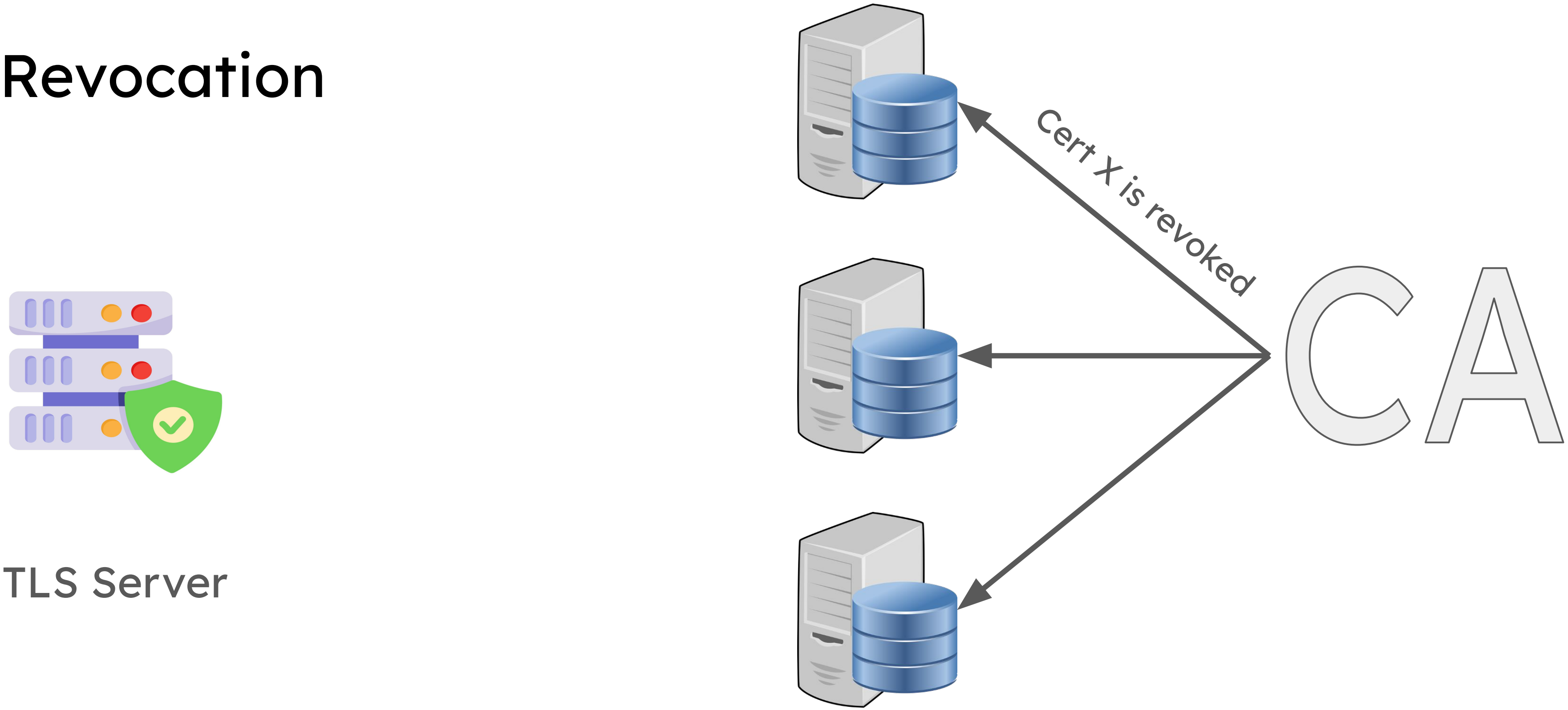
- This certificate is logged for those names
- There’s no revocation for this certificate

Transparency Logs

Security

	Certificate Transparency	This Draft
Stateless Client	Secure assuming no collusion	Secure assuming no collusion
Stateful Client	Secure assuming no collusion	Secure regardless of collusion

Revocation



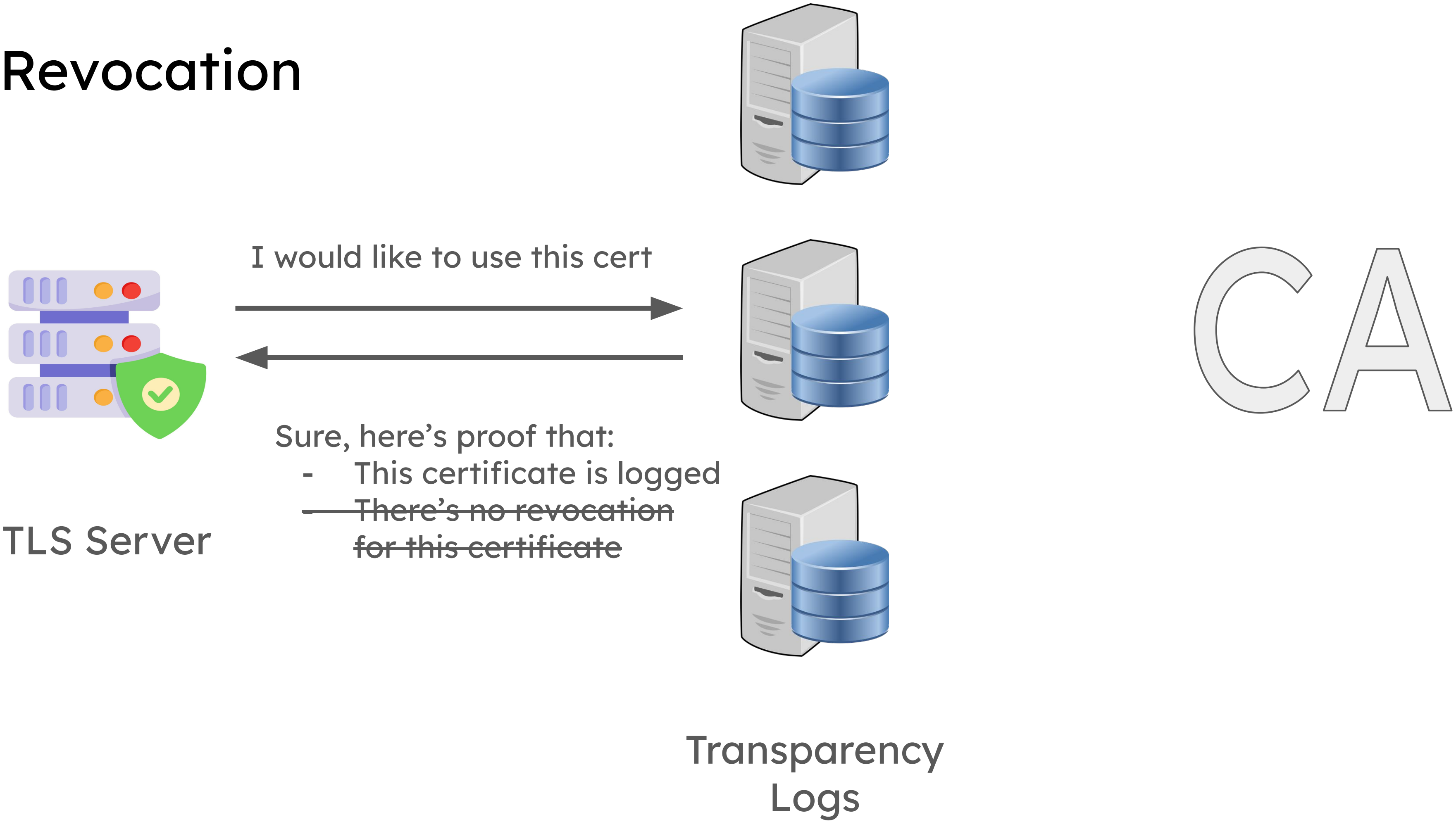
TLS Server

Cert X is revoked

CA

Transparency Logs

Revocation



I would like to use this cert

Sure, here’s proof that:

- This certificate is logged
- ~~There’s no revocation for this certificate~~

TLS Server

Transparency Logs

CA

Short-Lived Certificates

Reduction in certificate lifetime = corresponding increase in CT log throughput

CA renewal is a single point-of-failure
=> Revocation is slow

This Draft

Longer-lived certs are just as secure as shorter-lived certs & are less burden on logs

Automatic failover between logs prevents cascading outages

Questions?

Is this a good approach?

Should we keep working on it?