The Web PKI: Fundamental, Fragile, Fixable?

Thyla van der Merwe

8 November 2019

moz://a

amazon

citi

Gmail by Google

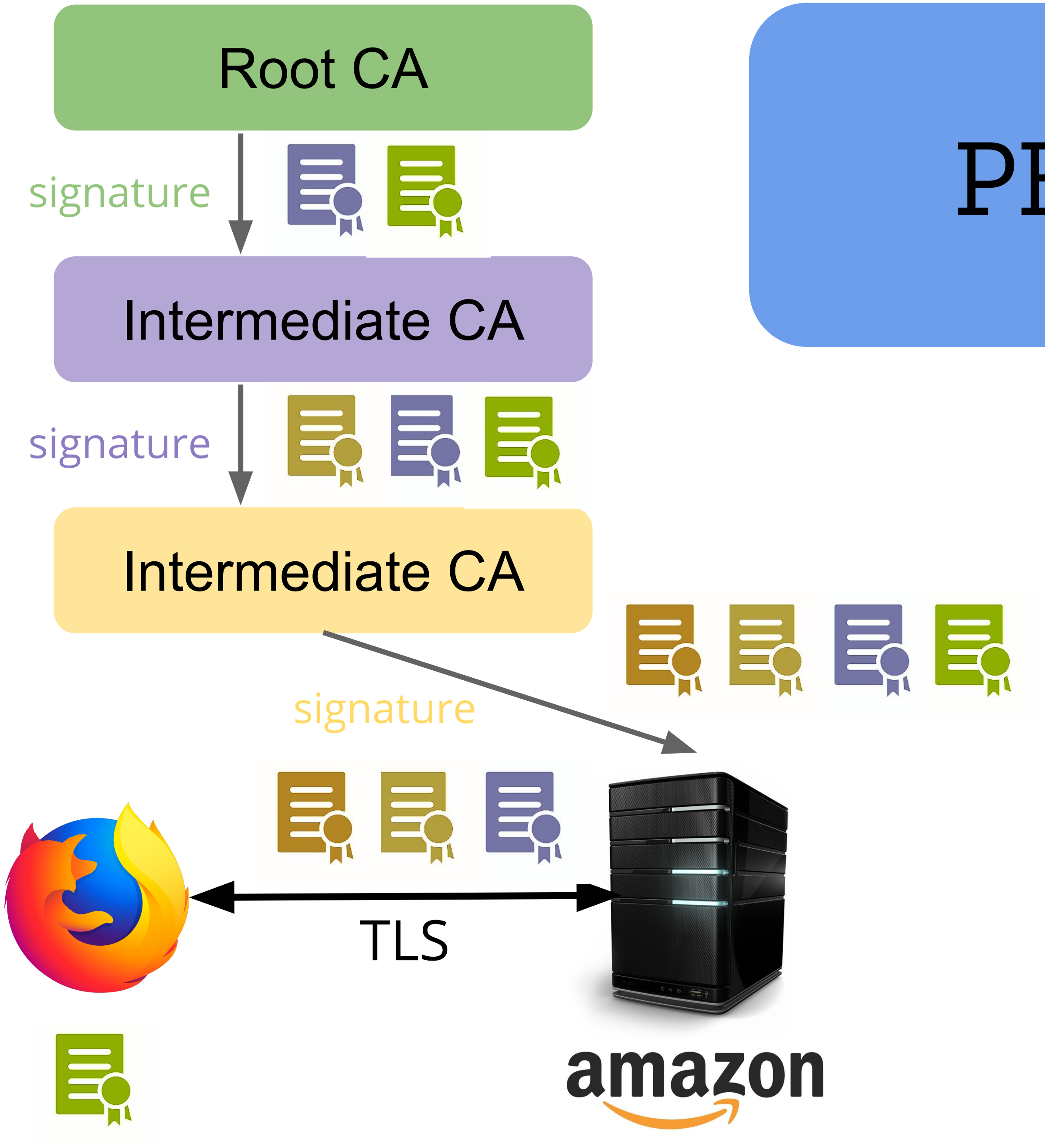
amazon

citi

Gmail
by Google

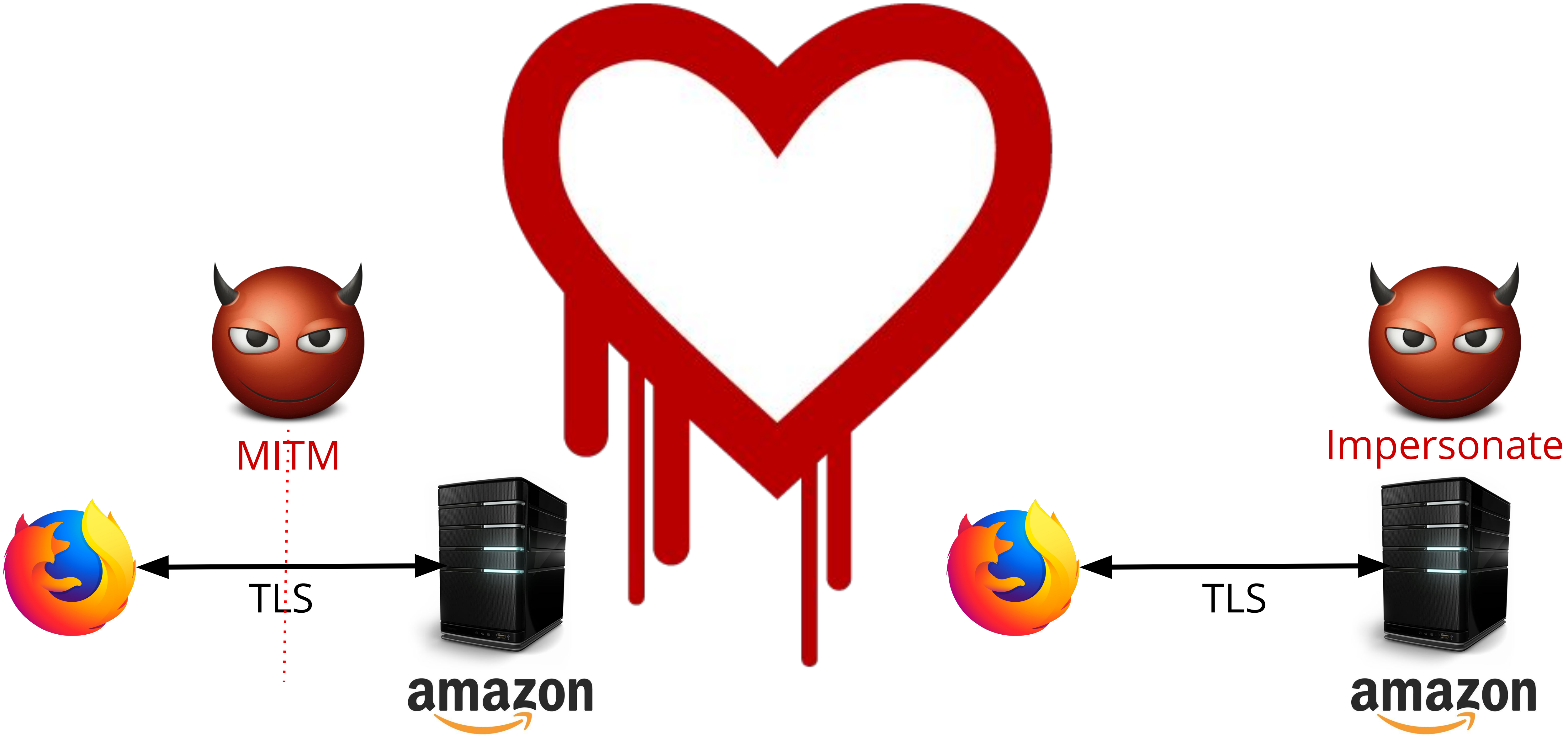
Trust = TLS + PKI

PKI


Root CA
signature
Intermediate CA
signature
Intermediate CA
signature
TLS
amazon

Fundamental to security on the Web!

> 500 000 private keys compromised!



> 500 000 private keys compromised!

The Register®
Biting the hand that feeds IT

Security

Trustwave to escape 'death penalty' for SSL skeleton key

Moz likely to spare certificate-confession biz same fate as DigiNotar

By John Leyden 14 Feb 2012 at 09:28

25 SHARE ▼

Analysis Trustwave's admission that it issued a digital "skeleton key" that allowed an unnamed private biz to spy on SSL-encrypted connections within its corporate network has sparked a fiery debate about trust on the internet.

Trustwave, an SSL certificate authority, confessed to supplying a subordinate root certificate as part of an information security product that allowed a customer to monitor employees' web communications - even if the staffers relied on HTTPS. Trustwave said the man-in-the-middle (MitM) gear was designed both to be tamper-proof and to work only within its unnamed client's compound. Despite these precautions, This system is pretty fragile!

We care about making it more robust.

Kathleen Wilson

Wayne Thayer

Dana Keeler

J.C. Jones

Kevin Jacobs

Infrastructure *CCADB, CAB Forum*	Revocation *CRLite*	Delegated Credentials *For TLS 1.3*

We care about making it more robust.

Kathleen Wilson

Wayne Thayer

Dana Keeler

J.C. Jones

Kevin Jacobs

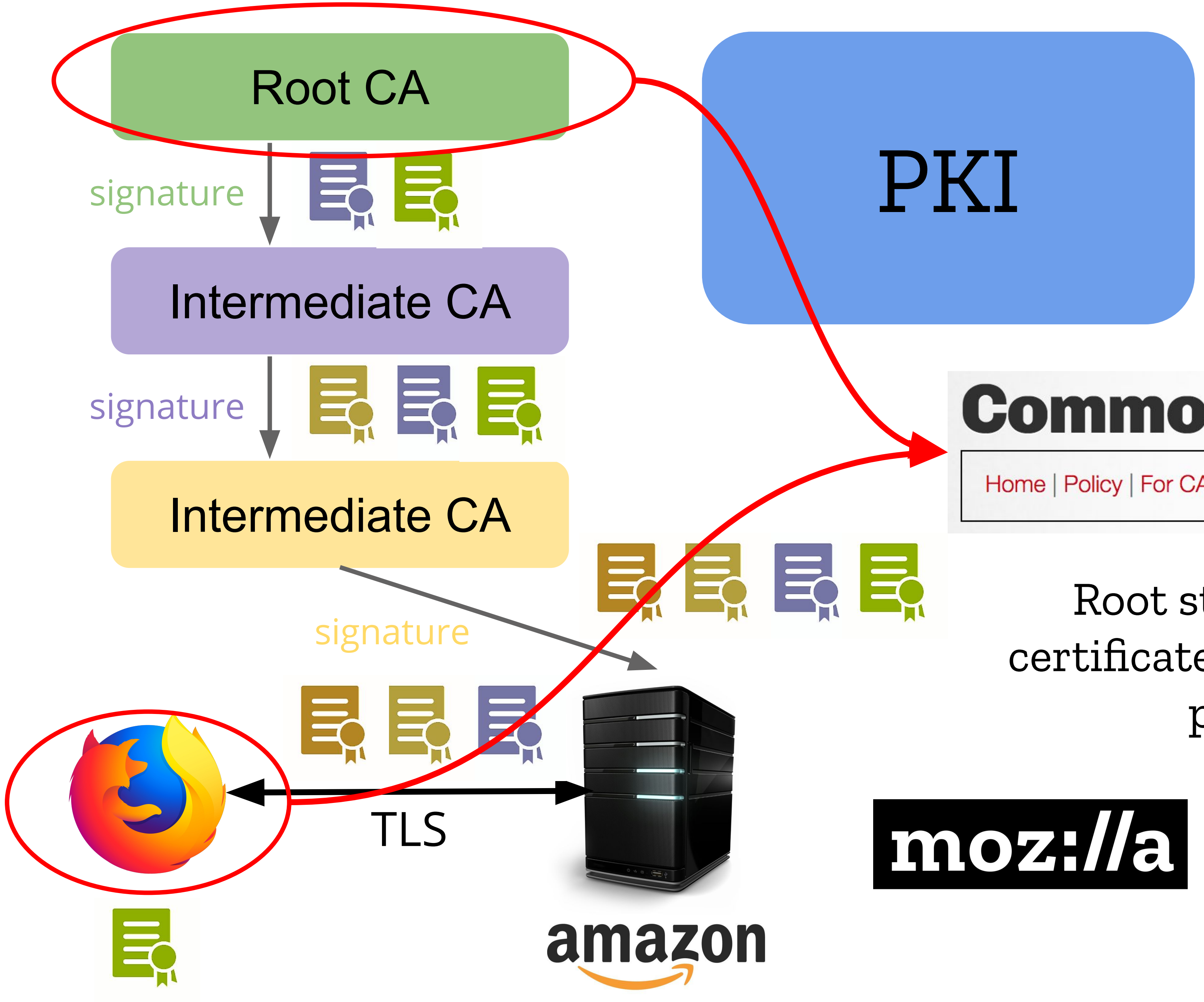
Franziskus Kiefer

Moritz Birghan

Dan Veditz

Marcus Burghardt

Infrastructure *CCADB, CAB Forum*	Revocation *CRLite*	Delegated Credentials *For TLS 1.3*



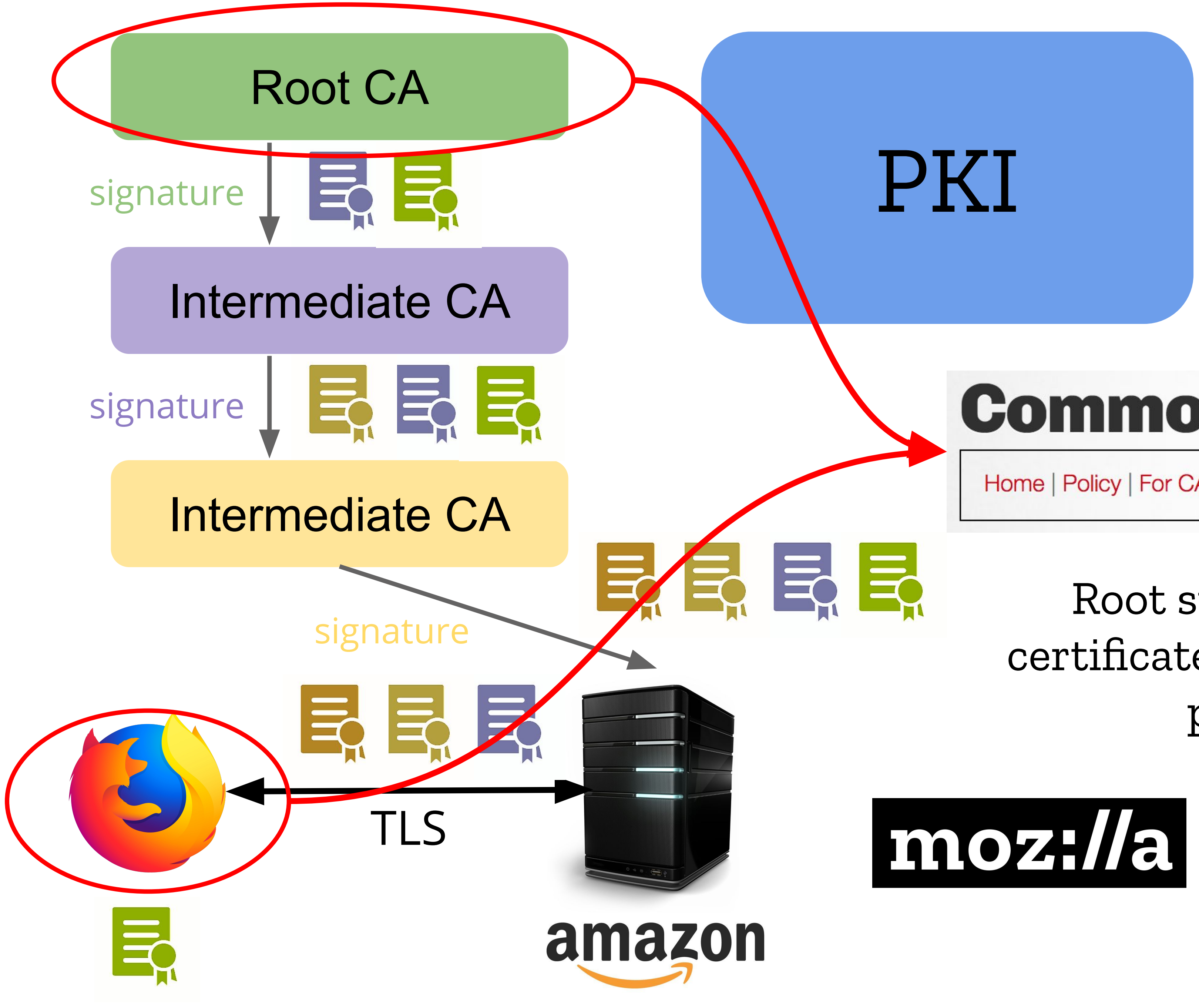
Common CA Database

Home | Policy | For CAs | For Root Stores | Resources

CCADB

Root store: List of trusted CA root certificates - strict application/inclusion process per root store.

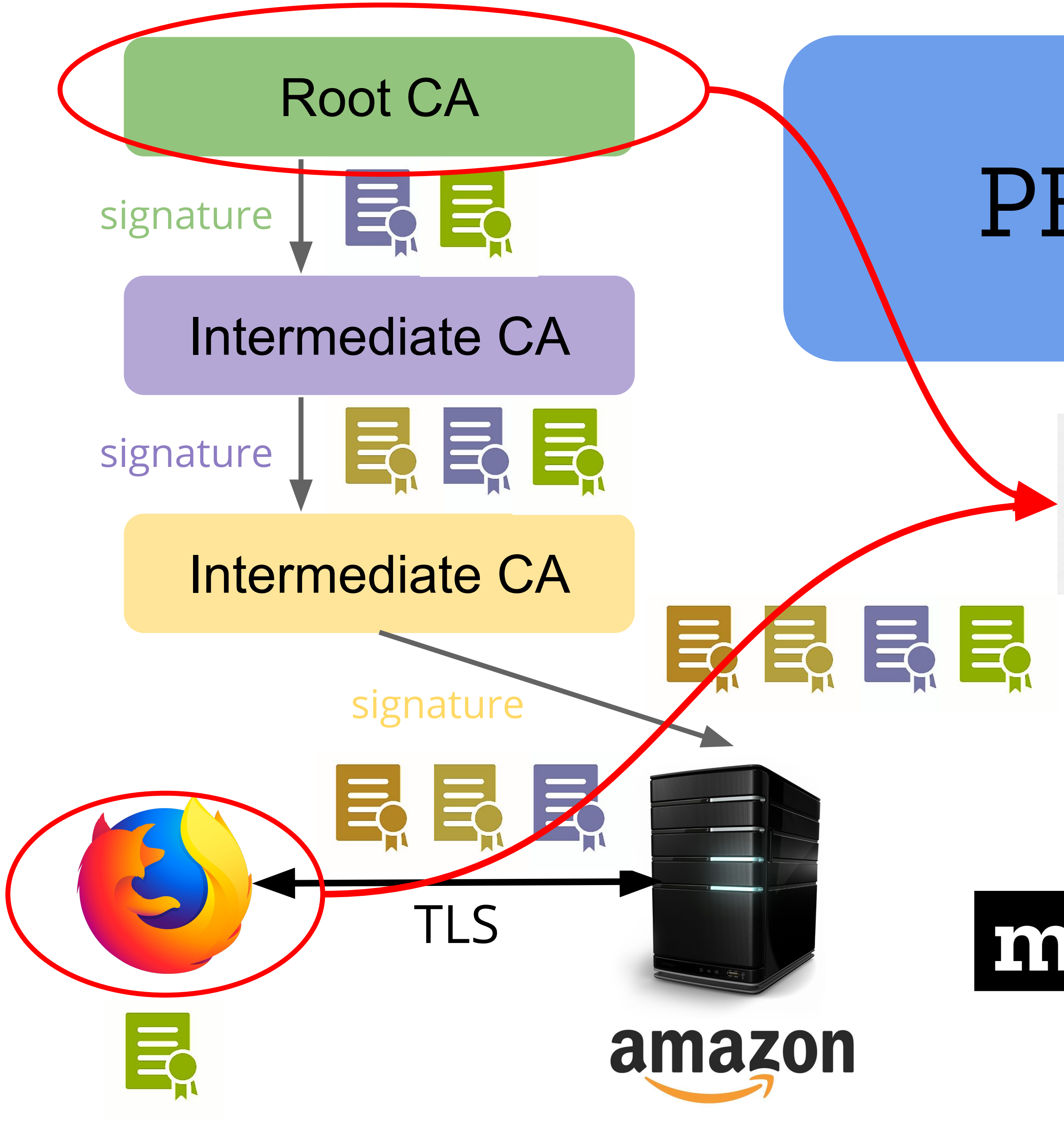
moz://a Microsoft Google



Repository of information about CAs and their certificates.

Root store: List of trusted CA root certificates - strict application/inclusion process per root store.

moz://a Microsoft Google



PKI

Managing root stores not always seen as a core business activity.

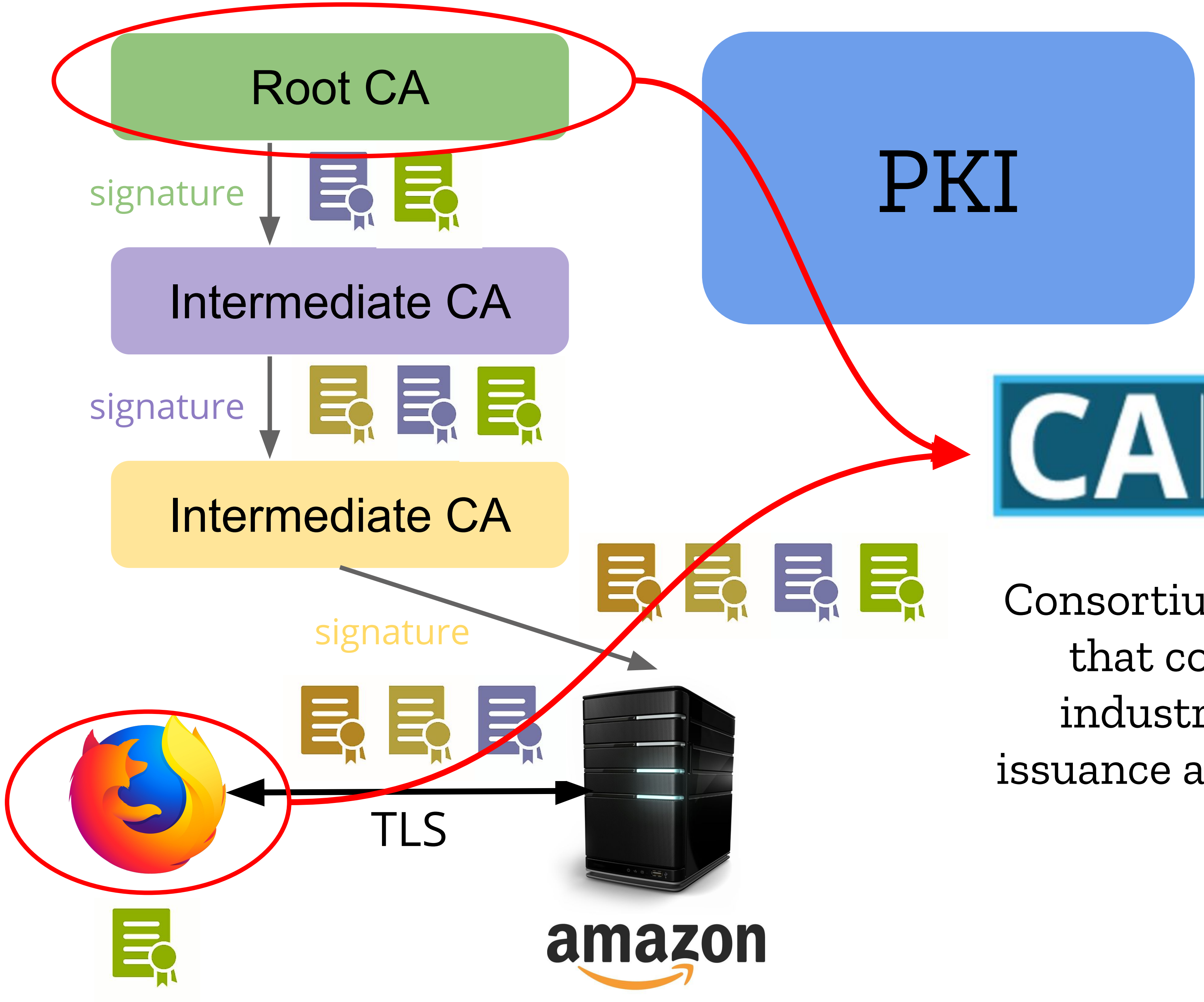
Common CA Database

Home | Policy | For CAs | For Root Stores | Resources

CCADB

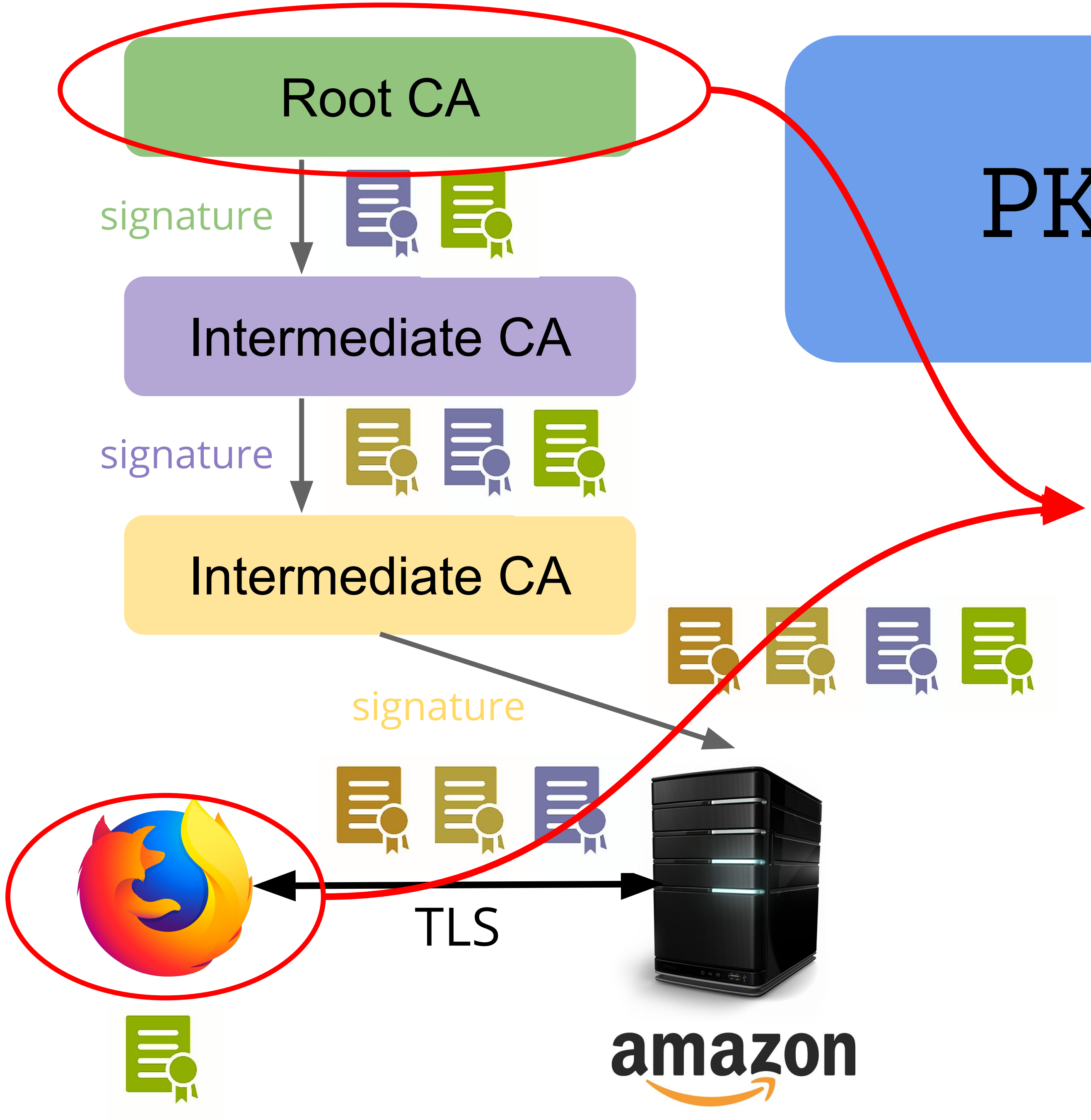
Root store: List of trusted CA root certificates - strict application/inclusion process per root store.

moz://a Microsoft Google



CA/BROWSER FORUM

Consortium of CAs and browser vendors that comes up with and maintains industry guidelines concerning the issuance and management of certificates.



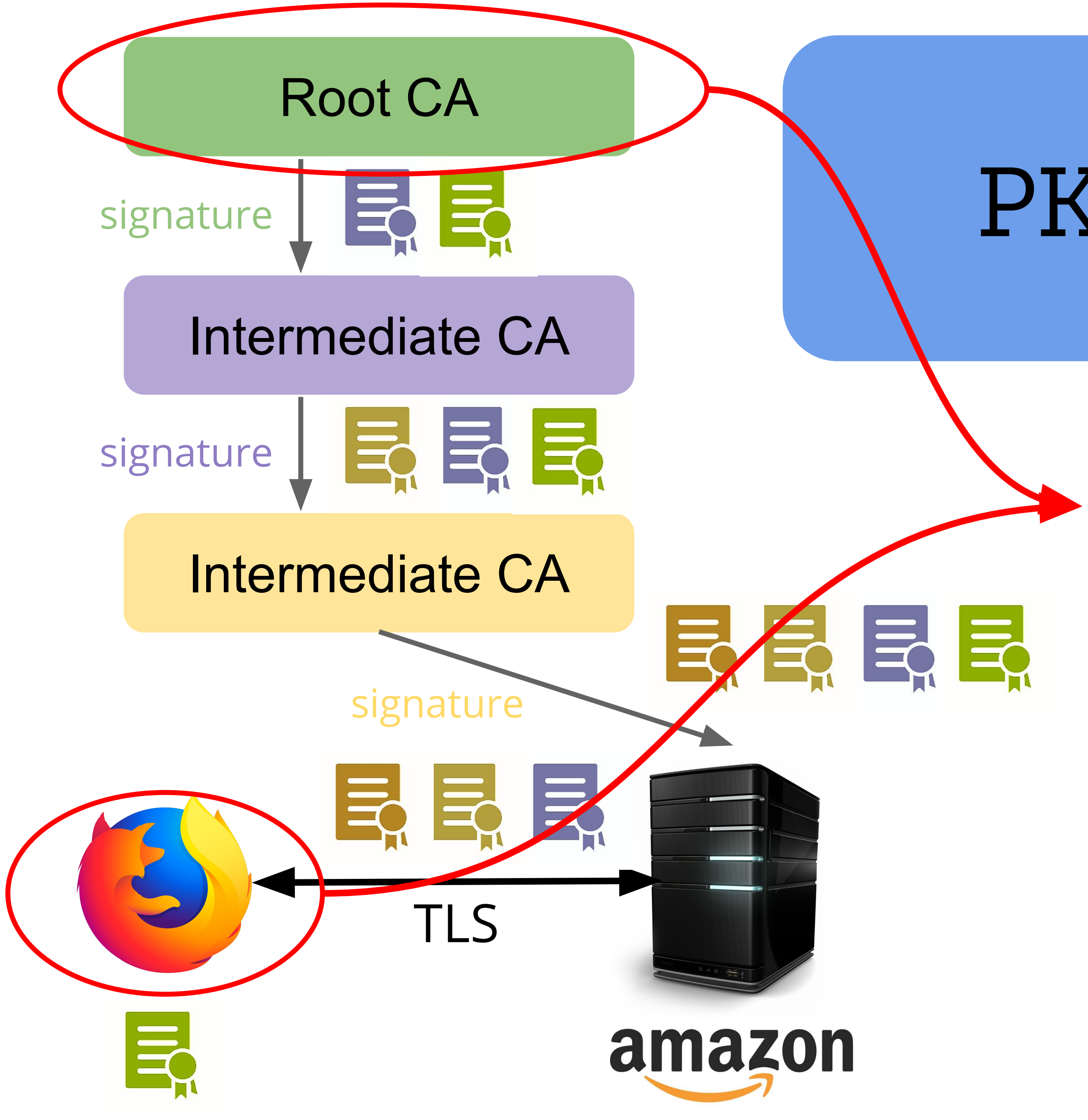
PKI

Browser proposal to shorten certificate lifetimes - good for security but fewer profits for CAs?

CA/BROWSER FORUM

Consortium of CAs and browser vendors that comes up with and maintains industry guidelines concerning the issuance and management of certificates.

moz://a



PKI

usenix THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

The Web's Identity Crisis: Understanding the Effectiveness of Website Identity Indicators

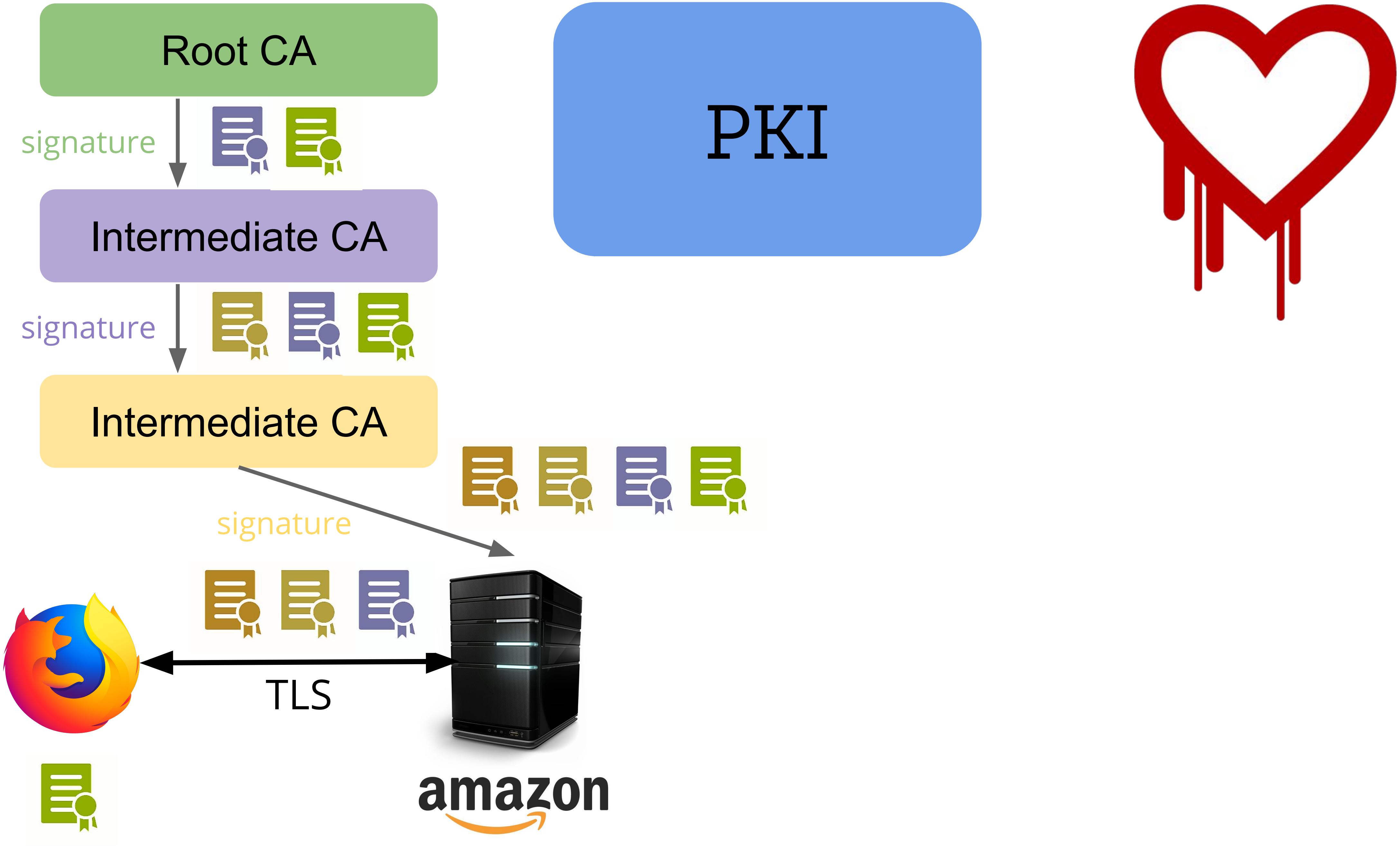
Christopher Thompson, Martin Shelton, Emily Stark, Maximilian Walker, Emily Schechter, and Adrienne Porter Felt, *Google*

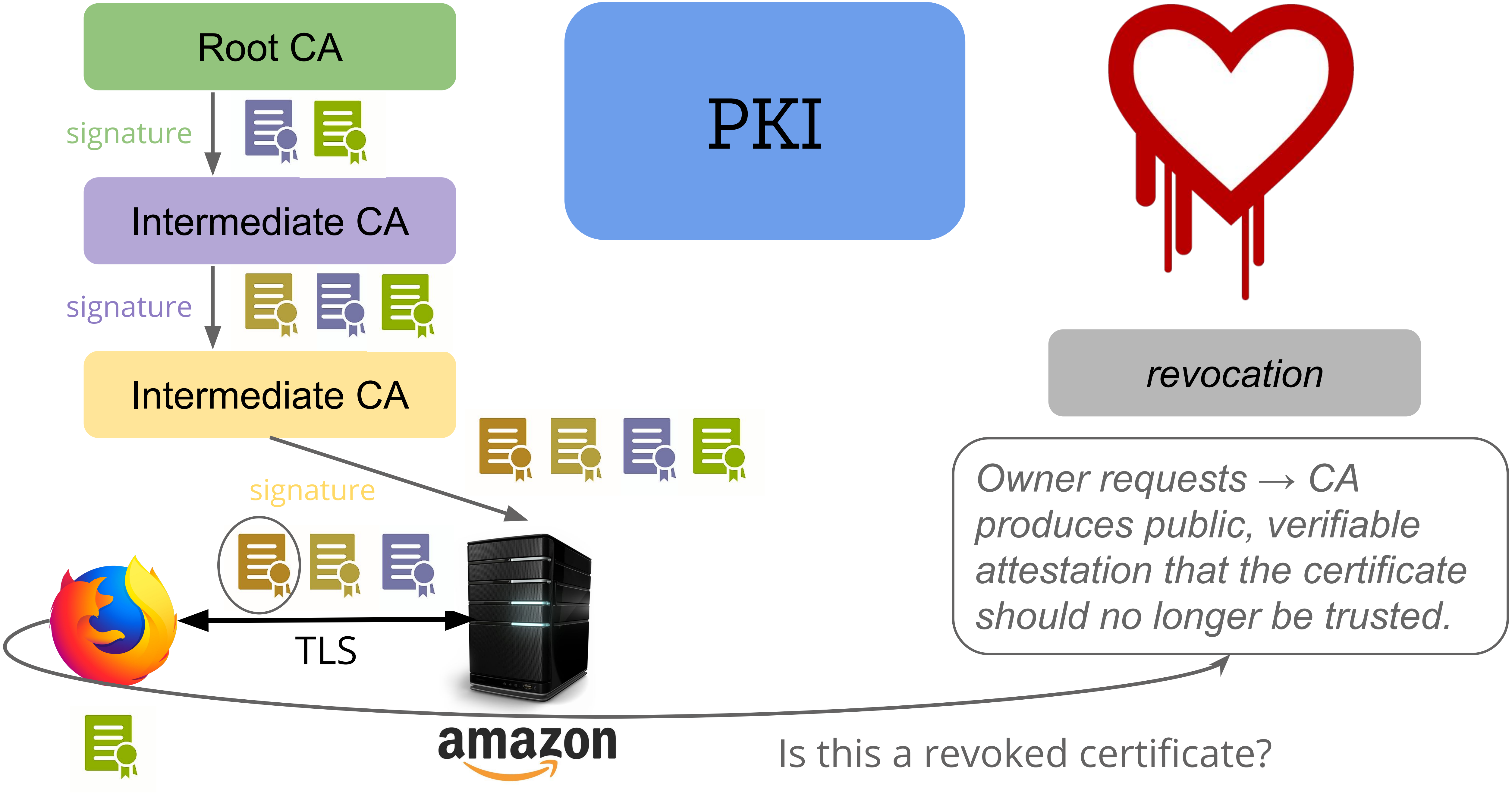
https://www.usenix.org/conference/usenixsecurity19/presentation/thompson

CAB CA/BROWSER FORUM

Consortium of CAs and browser vendors that comes up with and maintains industry guidelines concerning the issuance and management of certificates.

moz://a


Root CA
signature
Intermediate CA
signature
Intermediate CA
signature
TLS
amazon
PKI


Root CA
signature
Intermediate CA
signature
Intermediate CA
signature
TLS
amazon
PKI
revocation
Owner requests → CA produces public, verifiable attestation that the certificate should no longer be trusted.
Is this a revoked certificate?

Revocation is important!

Revocation is important!

Revocation is broken!

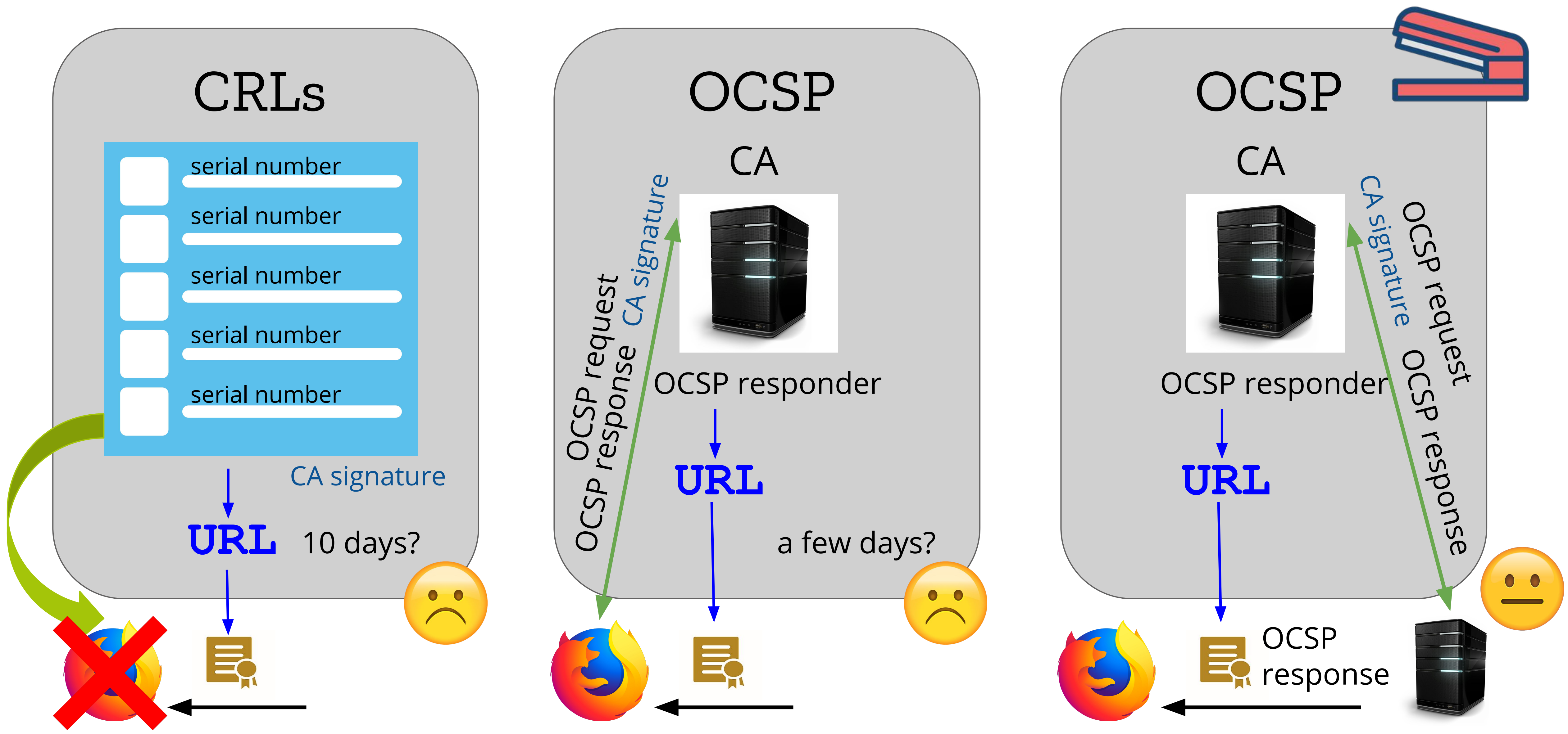
Revocation is important!

Revocation is broken!

38th IEEE Symposium on Security and Privacy

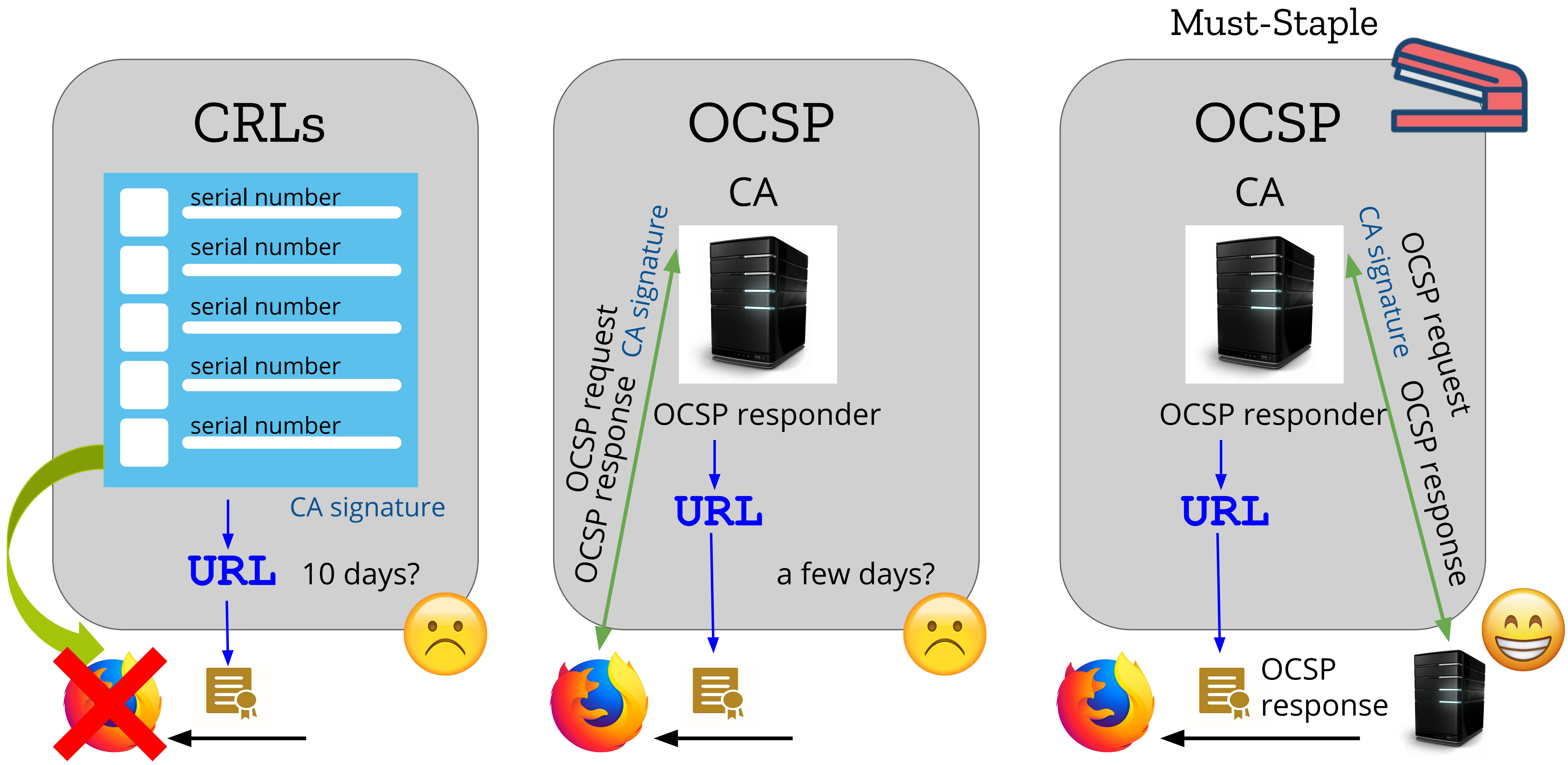
CRLite: A Scalable System for Pushing All TLS Revocations to All Browsers

James Larisch* David Choffnes* Dave Levin†
Bruce M. Maggs‡ Alan Mislove* Christo Wilson*
* Northeastern University † University of Maryland ‡ Duke University and Akamai Technologies



CRLs
serial number
serial number
serial number
serial number
serial number
CA signature
URL
10 days?
OCSP
CA
OCSP request
CA signature
OCSP response
OCSP responder
URL
a few days?
OCSP
CA
CA signature
OCSP request
OCSP response
OCSP responder
URL
OCSP
response

Fail-open vs Fail-closed


Must-Staple
CRLs
serial number
serial number
serial number
serial number
serial number
CA signature
URL
10 days?
OCSP
CA
OCSP request
OCSP response
CA signature
OCSP responder
URL
a few days?
OCSP
CA
CA signature
OCSP request
OCSP response
OCSP responder
URL
OCSP
response

Delays

200 ms

Fail-open

Privacy concerns

Push all revocation information to all clients?

CRLSet

OneCRL

Push all revocation information to all clients?

CRLSet

Google

OneCRL

moz://a

Size??

CRLite

Certificate Transparency

zmap

RAPID7

Certificate Ecosystem

Certificate Transparency

zmap

RAPID7

Certificate Ecosystem

Use a probabilistic data structure that supports queries for the finite set of unexpired certificates.

Certificate Transparency

zmap

RAPID7

Certificate Ecosystem

Use a probabilistic data structure that supports queries for the finite set of unexpired certificates.

Cascading Bloom Filters

Bloom Filters

0	0	0	0	0	0	0	0	0	0	0	0

$m = 12$

$k = 4$

for array indices

Bloom Filters

0	0	0	0	0	0	0	0	0	0	0	0

$m = 12$ $k = 4$

Let's put data item d in the filter:

Bloom Filters

0	0	0	0	1	0	0	0	0	0	0	0

$m = 12$ $k = 4$

Let's put data item *d* in the filter:

Compute *h_1*(d) = 4 → set bit in index 4 to 1.

Bloom Filters

0	0	0	0	1	0	0	0	0	0	0	1

m = 12 *k = 4*

Let's put data item *d* in the filter:

Compute *h_1*(d) = 4 → set bit in index 4 to 1.

Compute *h_2*(d) = 11 → set bit in index 11 to 1.

Bloom Filters

0	0	0	0	1	0	0	0	0	1	0	1

$m = 12$

$k = 4$

Let's put data item *d* in the filter:

Compute $h\_1(d) = 4 \rightarrow$ set bit in index 4 to 1.

Compute $h\_2(d) = 11 \rightarrow$ set bit in index 11 to 1.

Compute $h\_3(d) = 9 \rightarrow$ set bit in index 9 to 1.

Bloom Filters

0	0	1	0	1	0	0	0	0	1	0	1

$m = 12$ $k = 4$

Let's put data item *d* in the filter:

Compute *h_1*(d) = 4 → set bit in index 4 to 1.

Compute *h_2*(d) = 11 → set bit in index 11 to 1.

Compute *h_3*(d) = 9 → set bit in index 9 to 1.

Compute *h_4*(d) = 2 → set bit in index 2 to 1.

Bloom Filters

0	0	1	0	1	0	0	0	0	1	0	1

$m = 12$

$k = 4$

Let's put data item d in the filter:

Compute $h\_1(d) = 4 \rightarrow$ set bit in index 4 to 1.

Compute $h\_2(d) = 11 \rightarrow$ set bit in index 11 to 1.

Compute $h\_3(d) = 9 \rightarrow$ set bit in index 9 to 1.

Compute $h\_4(d) = 2 \rightarrow$ set bit in index 2 to 1.

Add another item d'?

Bloom Filters

0	1	1	0	1	1	0	1	0	1	0	1

m = *12*

k = *4*

Let's put data item *d* in the filter:

Compute *h_1*(*d*) = 4 → set bit in index 4 to 1.

Compute *h_2*(*d*) = 11 → set bit in index 11 to 1.

Compute *h_3*(*d*) = 9 → set bit in index 9 to 1.

Compute *h_4*(*d*) = 2 → set bit in index 2 to 1.

Add another item *d'*?

Bloom Filters

0	1	1	0	1	1	0	1	0	1	0	1

$m = 12$ $k = 4$

Is $d*$ in the filter?

If any of the $h\_i(d*)$ values is 0 then **DEFINITELY NOT** in the filter.

If all of the $h\_i(d*)$ values are 1 then **MAYBE** in the filter.

Bloom Filters

0	1	1	0	1	1	0	1	0	1	0	1

$m = 12$

$k = 4$

Is d^* in the filter?

If any of the $h_i(d^*)$ values is 0 then **DEFINITELY NOT** in the filter.

If all of the $h_i(d^*)$ values are 1 then **MAYBE** in the filter.

So maybe it's a legitimate insertion, maybe it's not.

Bloom Filters

0	1	1	0	1	1	0	1	0	1	0	1

$m = 12$

$k = 4$

Will have false positives → rate p determined by m, k, occupancy.

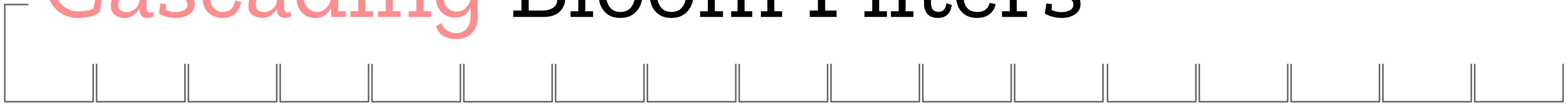
Say we want to store $R \subseteq U$. R is the set of revoked certificates, and U is the finite set of unexpired certificates. $R \cup S = U$.

But there will be false positives!

Say we want to store $R \subseteq U$. R is the set of revoked certificates, and U is the finite set of unexpired certificates. $R \cup S = U$.

Store those in another bloom filter.

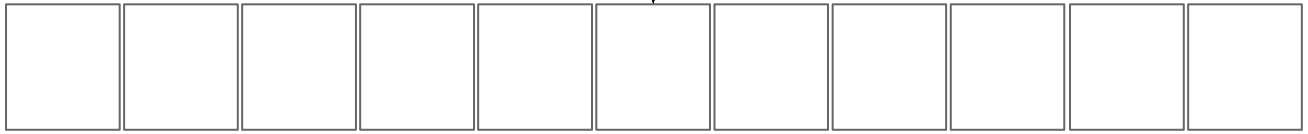
Cascading Bloom Filters



BF1

But there are going to be

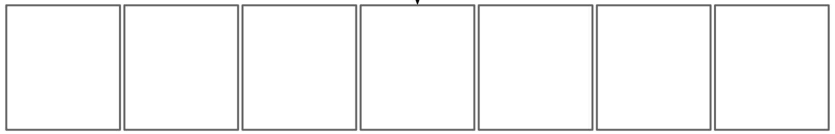
false positives



BF2

But there are going to be

false positives

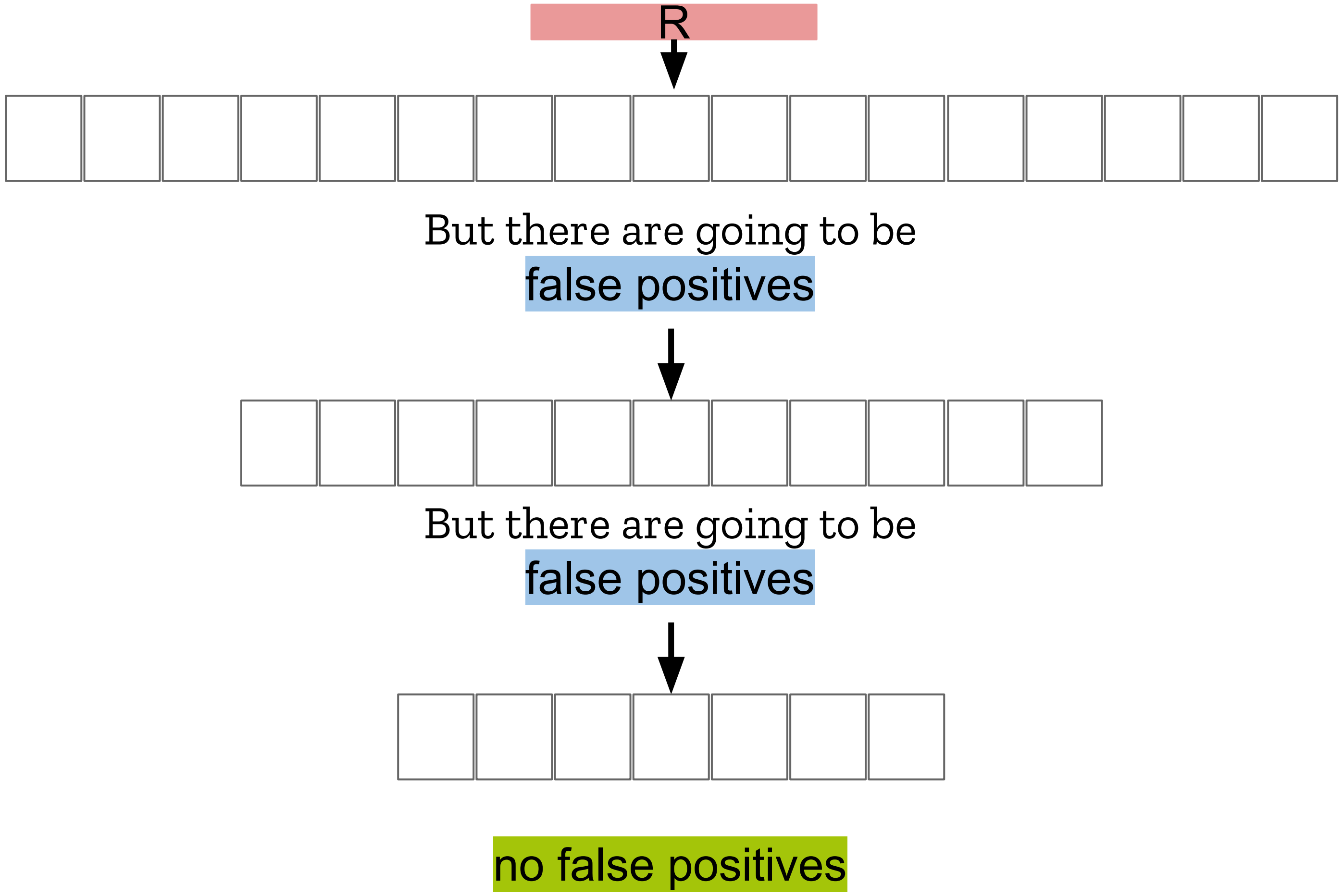


BFx

no false positives*

* chance of false positives is negligible

Cascading Bloom Filters



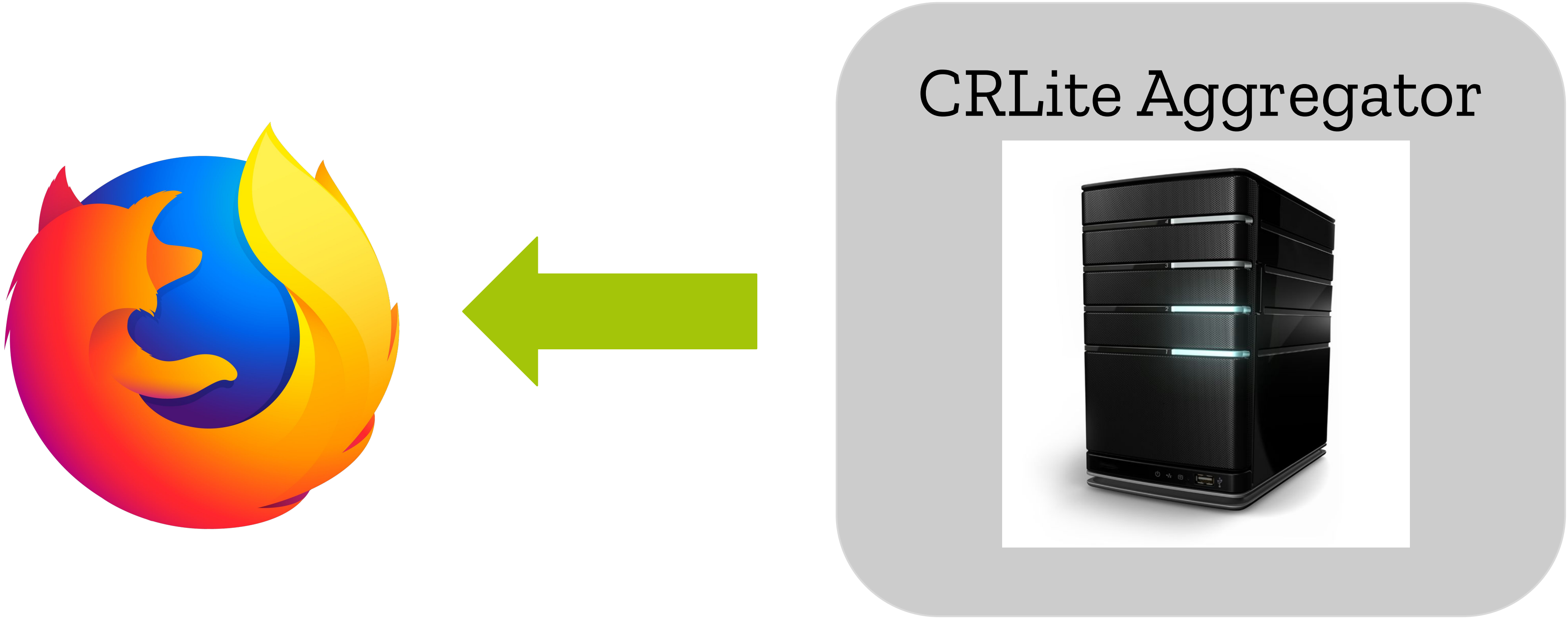
3 levels

If *cert* in not in BF1, then definitely not in *R*. If *cert* is in BF1, then we don't know.

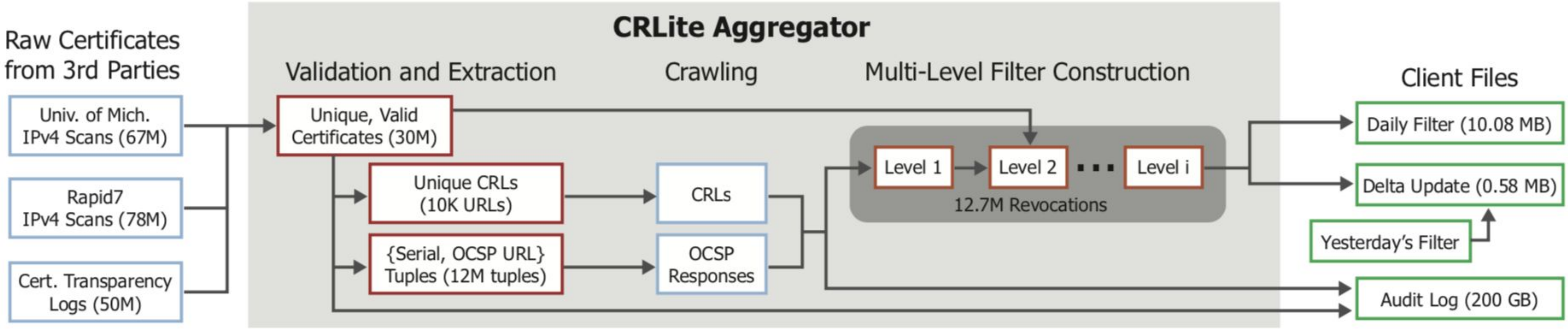
If *cert* in BF1 but not in BF2, then in *R*. If *cert* is in BF1 and BF2, then we don't know.

If *cert* in BF1 and BF2 but not in BF3, then definitely not in *R*. If *cert* is all three, then in *R*.

CRLite Architecture

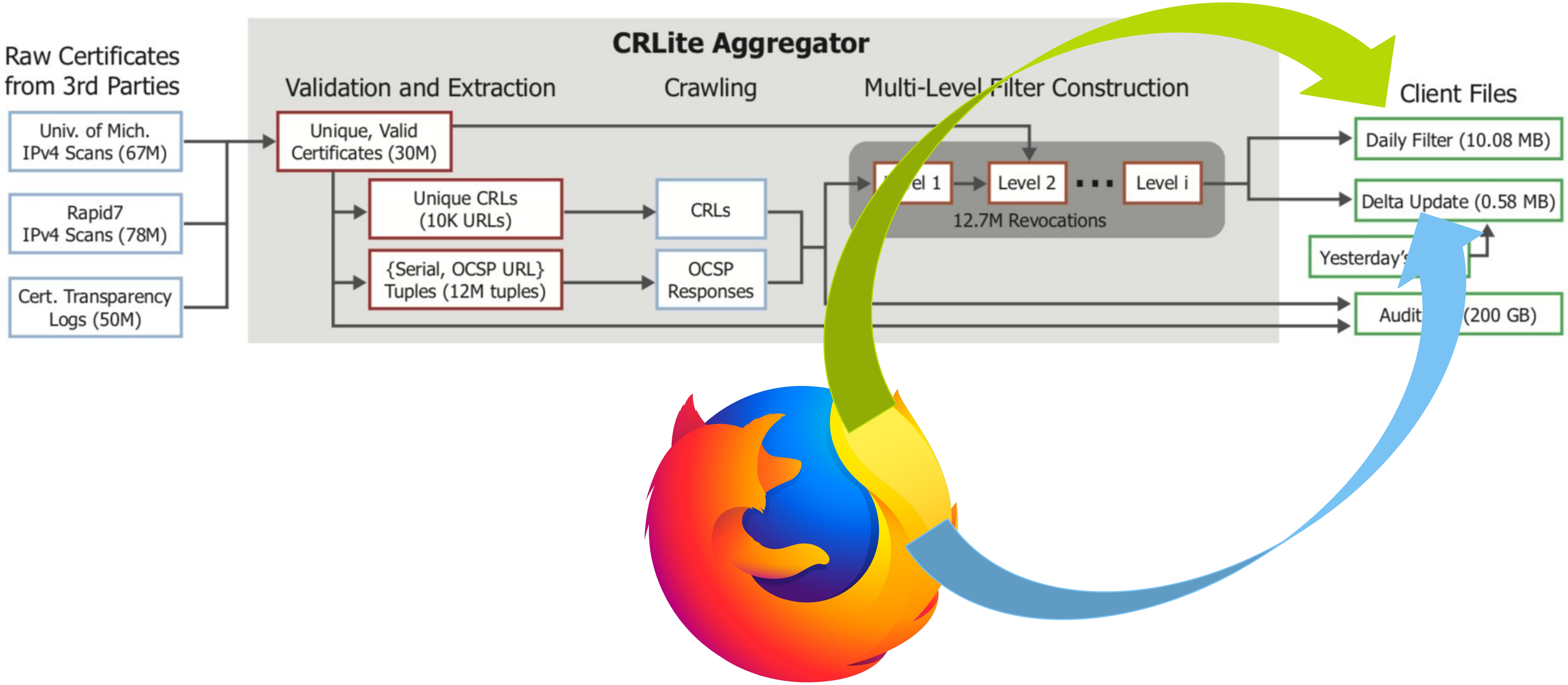


CRLite Architecture



CRLite: A Scalable System for Pushing All TLS Revocations to All Browsers. Larisch et al. IEEE S&P 2017

CRLite Architecture



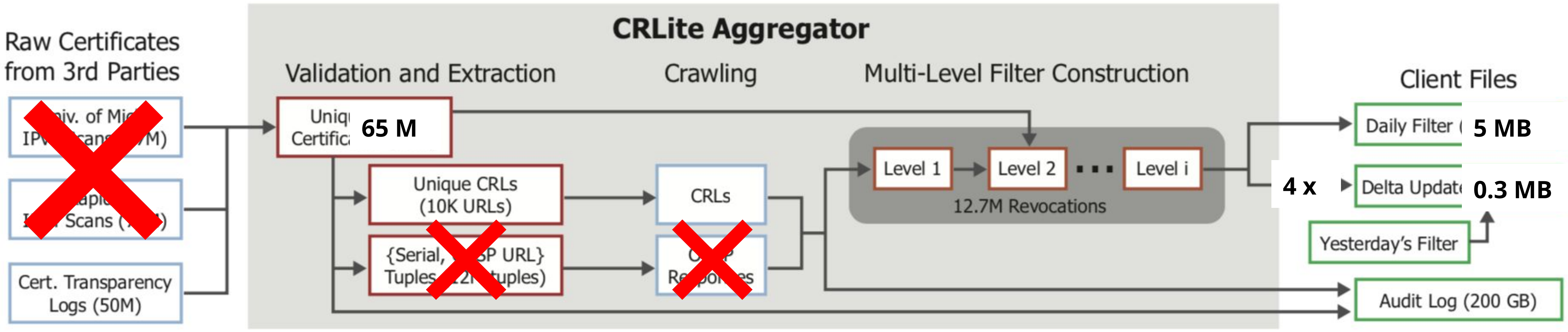
Implementing CRLite for Firefox

Principle 4

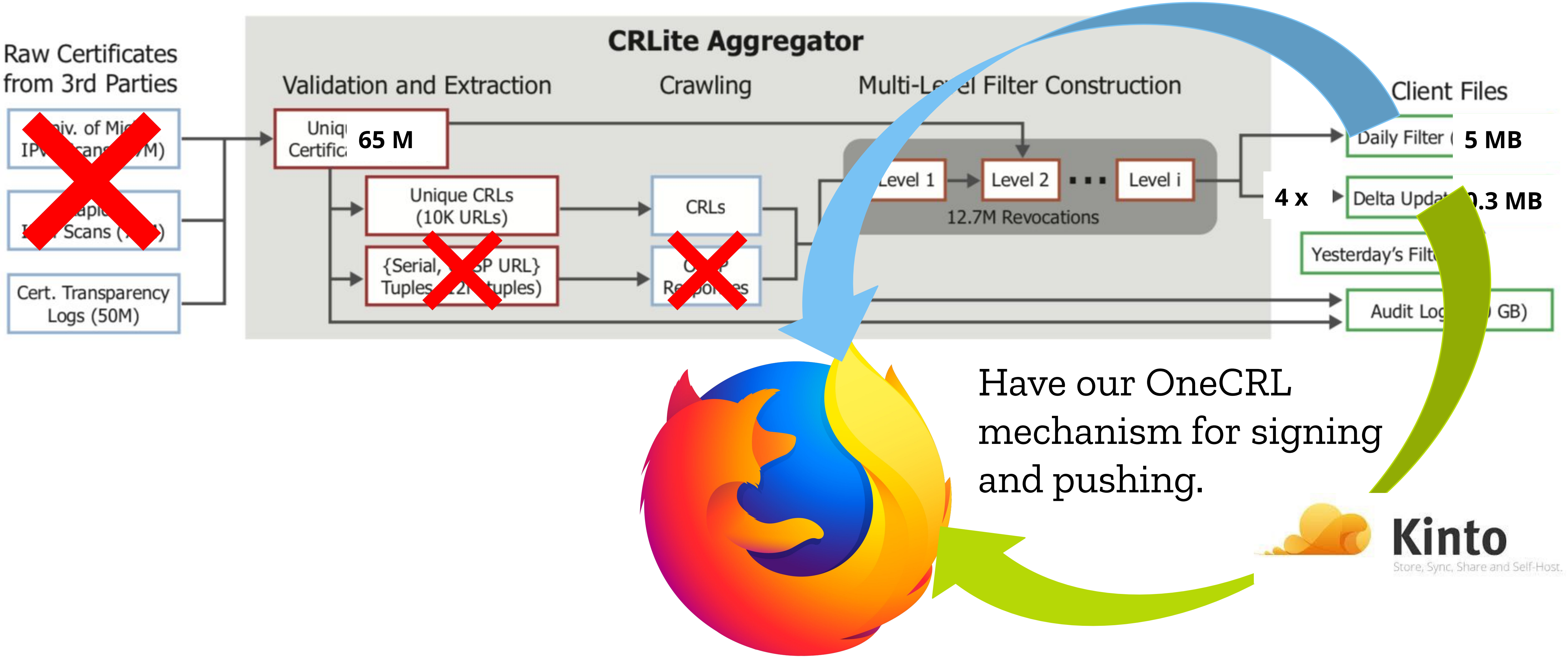
Individuals' security and privacy on the Internet are fundamental and must not be treated as optional.

- CRL-like properties
- Small data sizes (fast to parse)
- Incremental updates
- Scales well
- Builds on useful properties of CT

Implementing CRLite for Firefox



Implementing CRLite for Firefox



Have our OneCRL mechanism for signing and pushing.

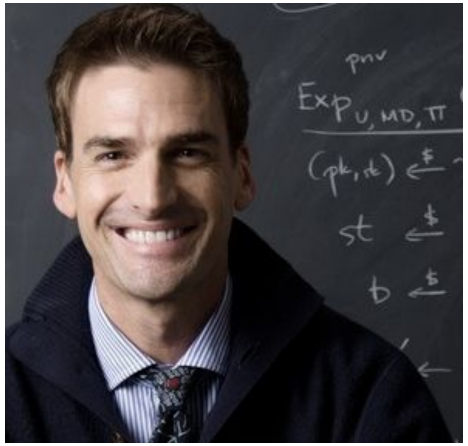
Implementing CRLite for Firefox

Paper did have a prototype using Firefox → built as a Firefox extension.

Academic Prototype	Mozilla Prototype
TLS APIs for cert checking - JavaScript (11.9MB memory)	Native code (C++, Rust, some JS)
10ms to check a cert chain (6ms with cache tricks) - includes parsing certs (API provides unparsed certs)	0.01 - 0.04 ms - We check end-entity certs - Use OneCRL -> intermediates

Are we done yet?

The proceedings version of this paper appears at CCS '19. This is the full version.



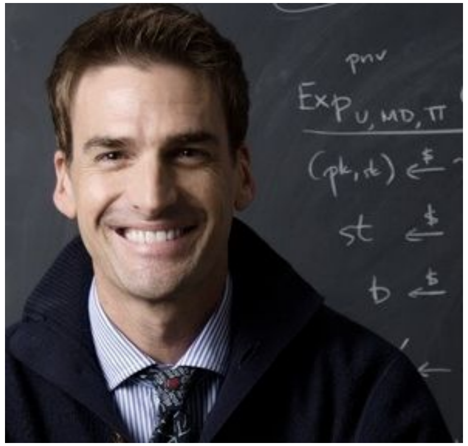
Probabilistic Data Structures in Adversarial Environments

David Clayton, Christopher Patton, and Thomas Shrimpton

Florida Institute for Cybersecurity Research
Computer and Information Science and Engineering
University of Florida

{davidclayton,cjpatton,teshrim}@ufl.edu

Are we done yet?



The proceedings version of this paper appears at CCS '19. This is the full version.

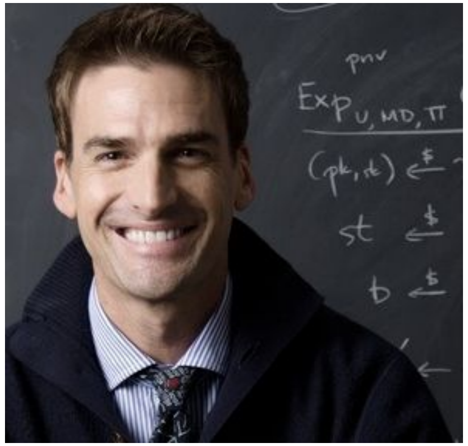
Probabilistic Data Structures in Adversarial Environments

David Clayton, Christopher Patton, and Thomas Shrimpton

Florida Institute for Cybersecurity Research
Computer and Information Science and Engineering
University of Florida

{davidclayton,cjpatton,teshrim}@ufl.edu

Are we done yet?



The proceedings version of this paper appears at CCS '19. This is the full version.

Probabilistic Data Structures in Adversarial Environments

David Clayton, Christopher Patton, and Thomas Shrimpton

Florida Institute for Cybersecurity Research
Computer and Information Science and Engineering
University of Florida

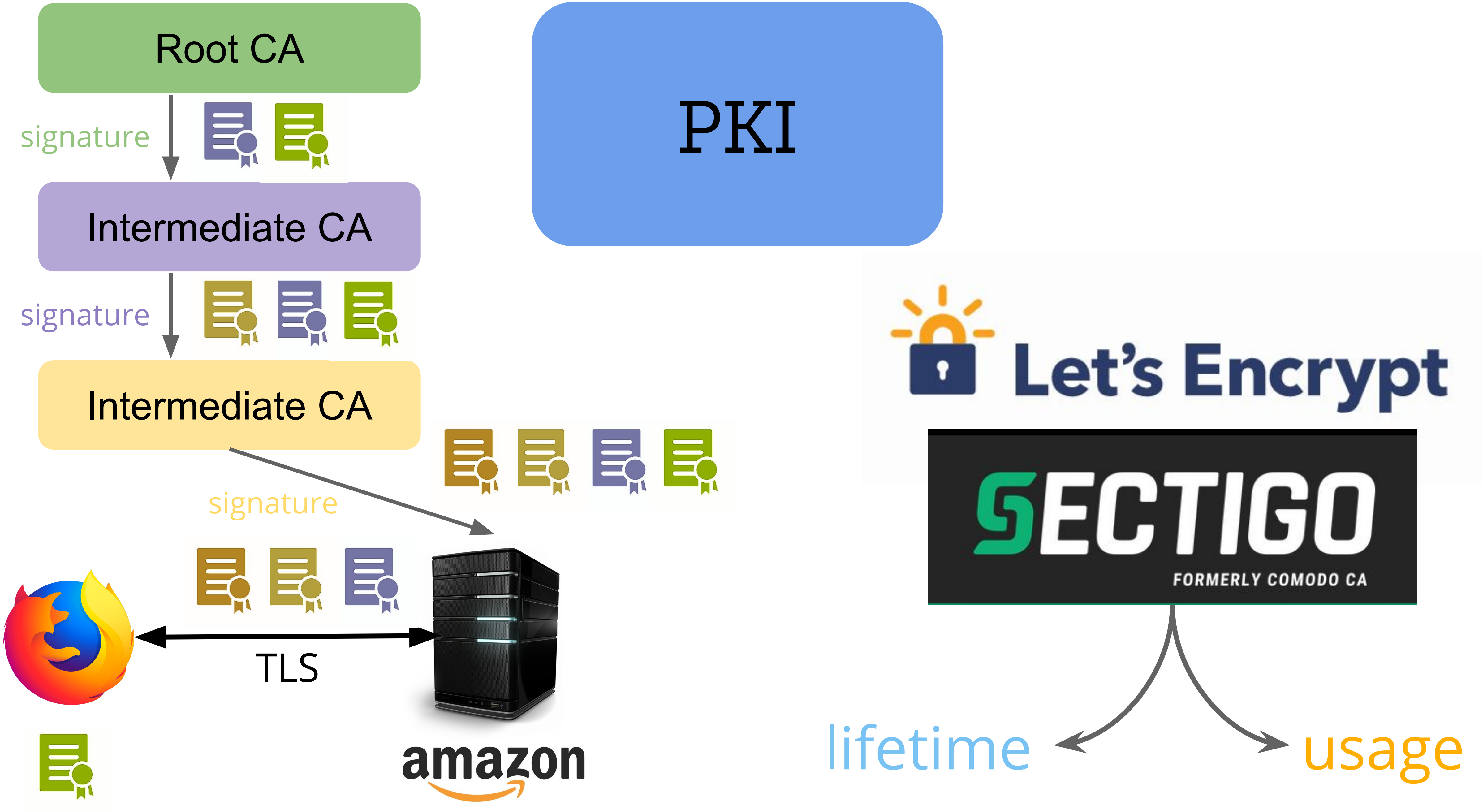
{davidclayton,cjpatton,teshrim}@ufl.edu

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 6, 2020

R. Barnes
Cisco
S. Iyengar
Facebook
N. Sullivan
Cloudflare
E. Rescorla
Mozilla

November 03, 2019

Delegated Credentials for TLS
draft-ietf-tls-subcerts-05


PKI
Root CA
signature
Intermediate CA
signature
Intermediate CA
signature
TLS
amazon
Let's Encrypt
SECTIGO
FORMERLY COMODO CA
lifetime
usage

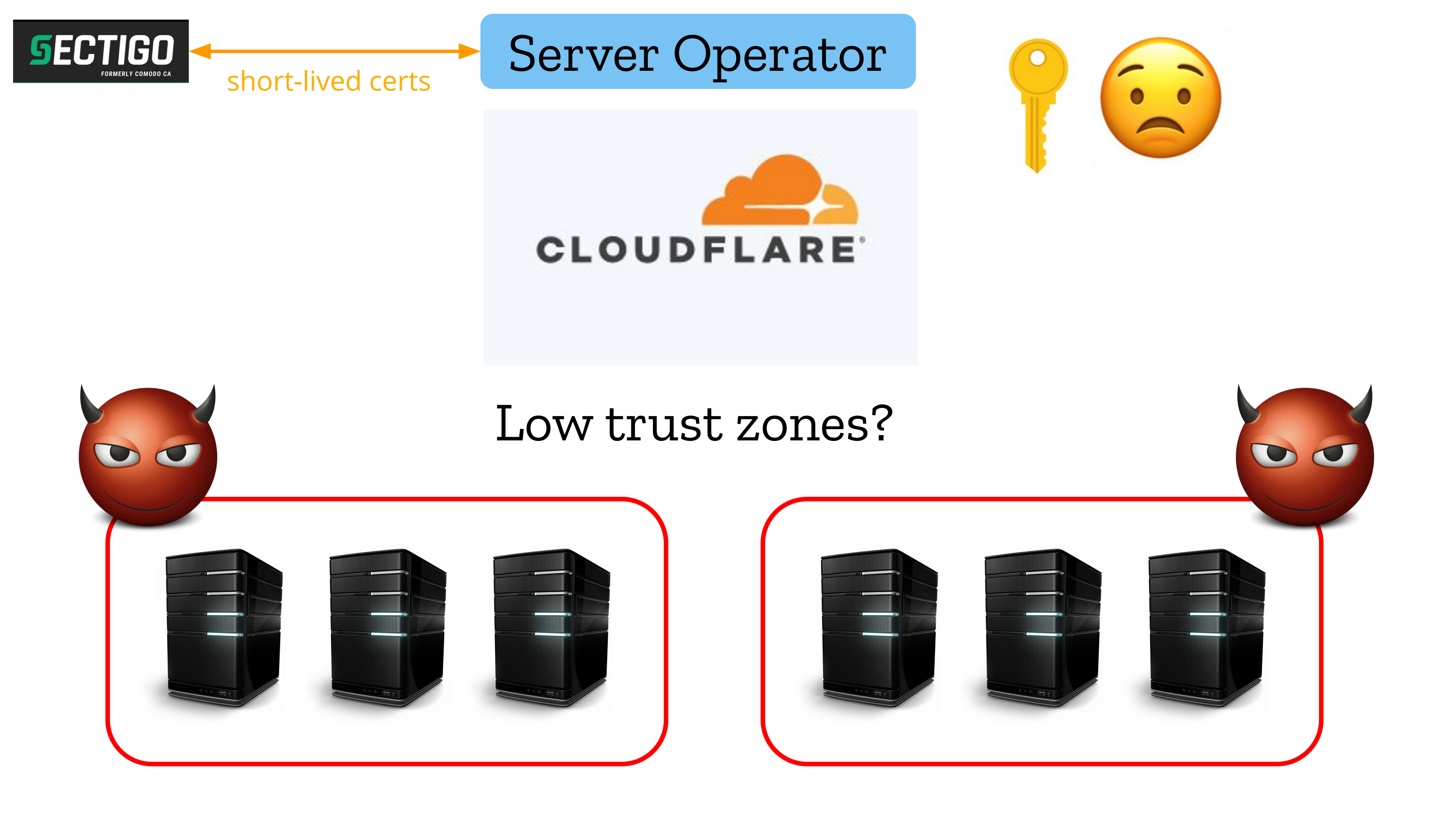
Server Operator

CLOUDFLARE

Low trust zones?

Server Operator

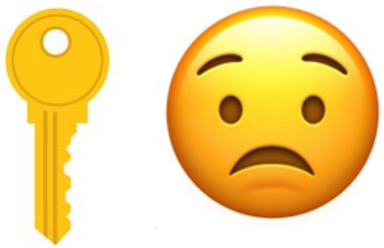
Low trust zones?


SECTIGO
FORMERLY COMODO CA
short-lived certs
Server Operator
CLOUDFLARE
Low trust zones?

SECTIGO
FORMERLY COMODO CA

~~short-lived certs~~

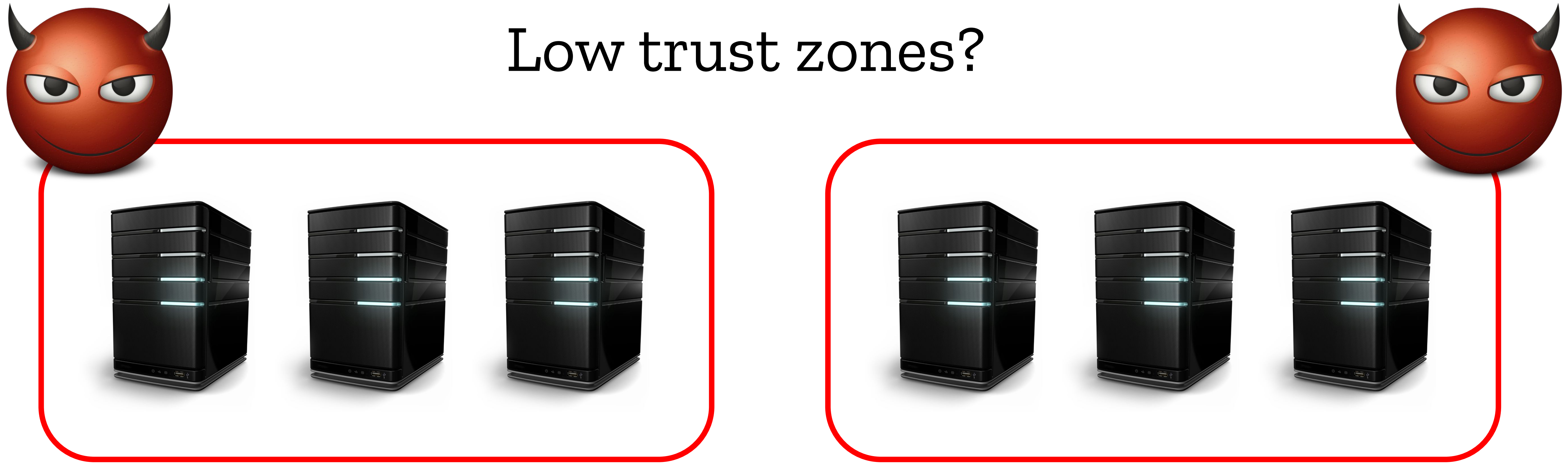
Server Operator

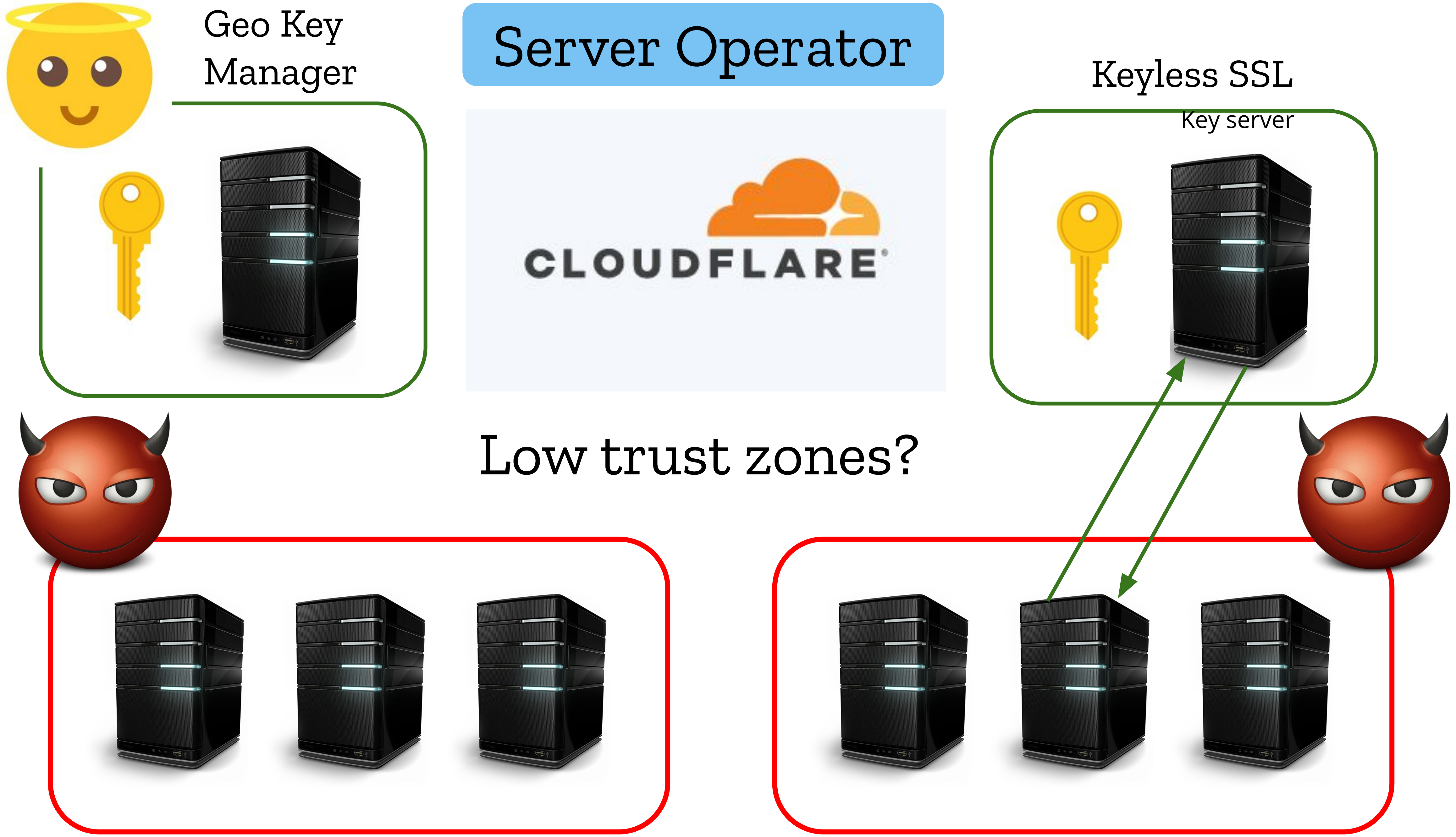


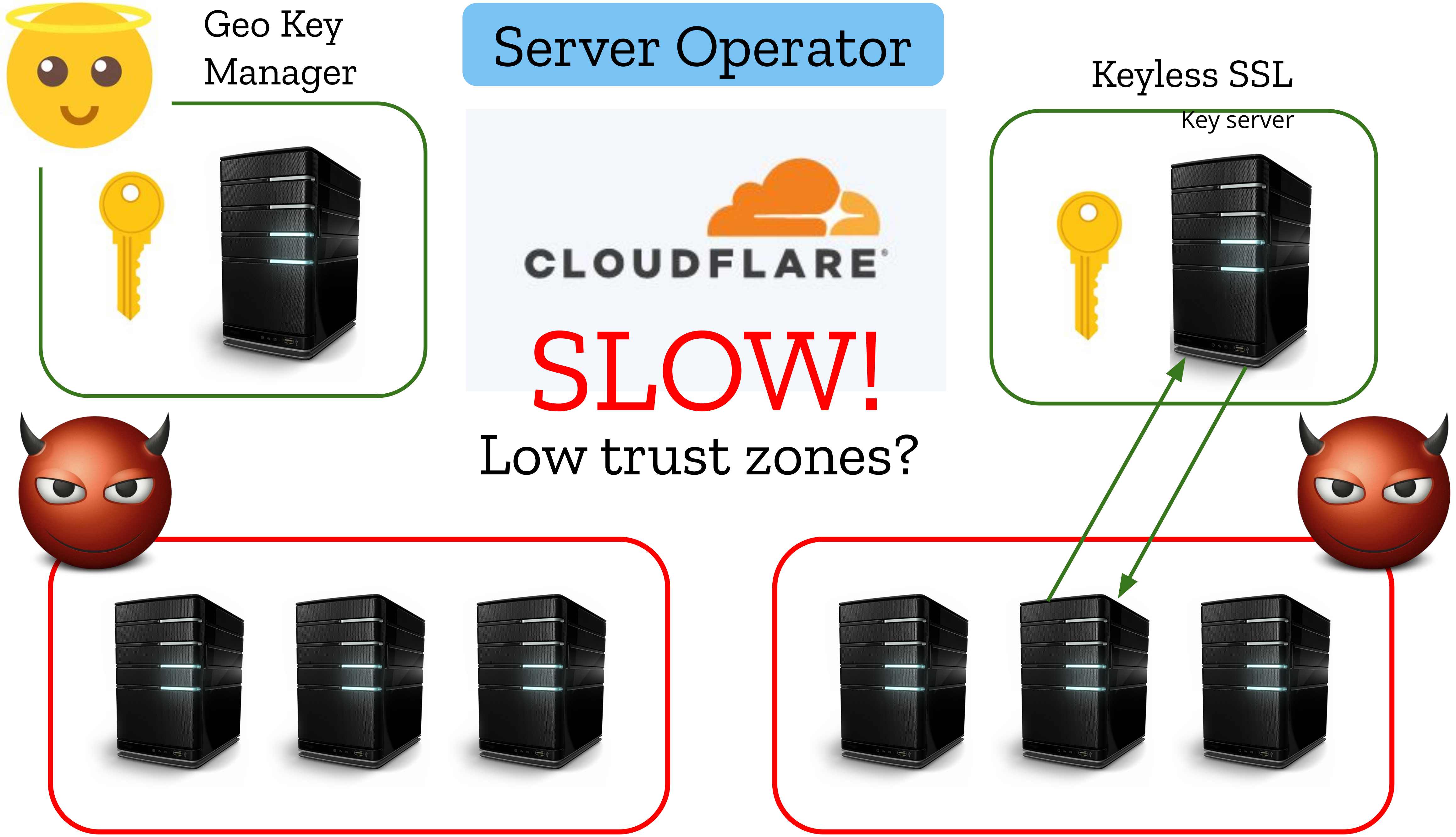
CLOUDFLARE

Operationally costly!

Low trust zones?

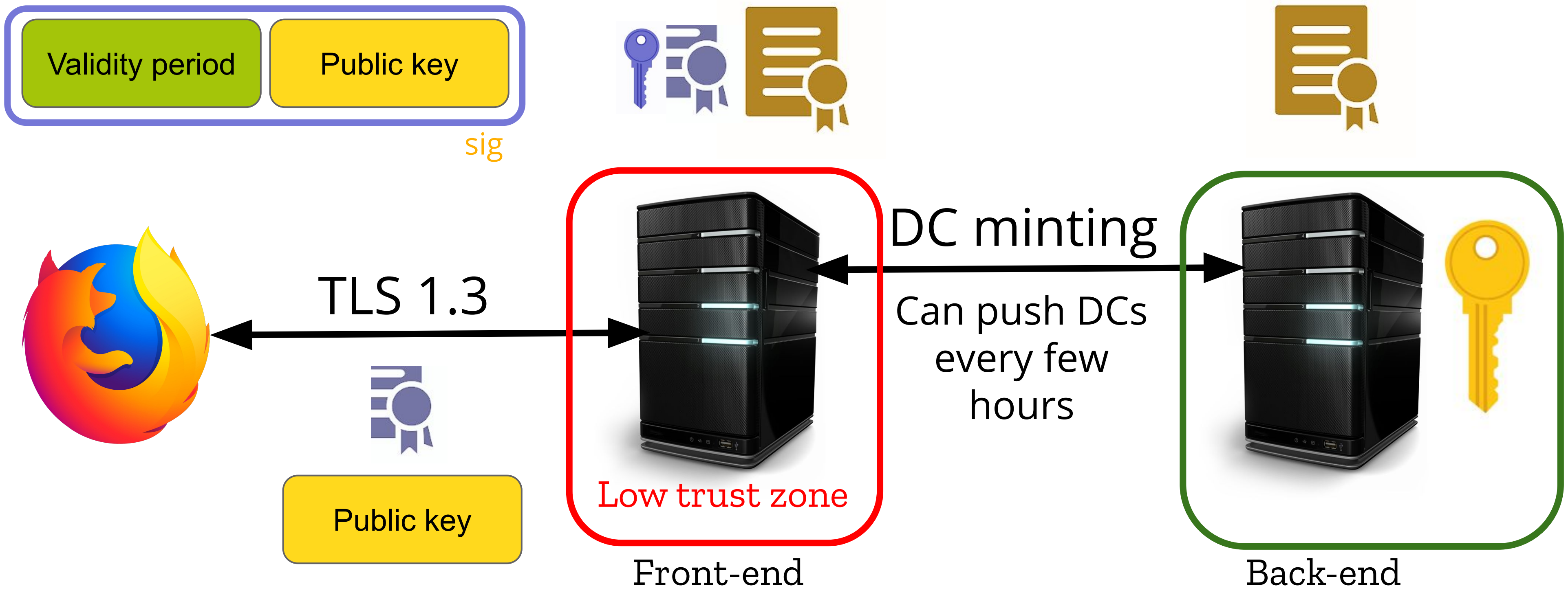



Geo Key
Manager
Server Operator
CLOUDFLARE
Keyless SSL
Key server
Low trust zones?


Server Operator
Geo Key
Manager
Keyless SSL
Key server
CLOUDFLARE
SLOW!
Low trust zones?

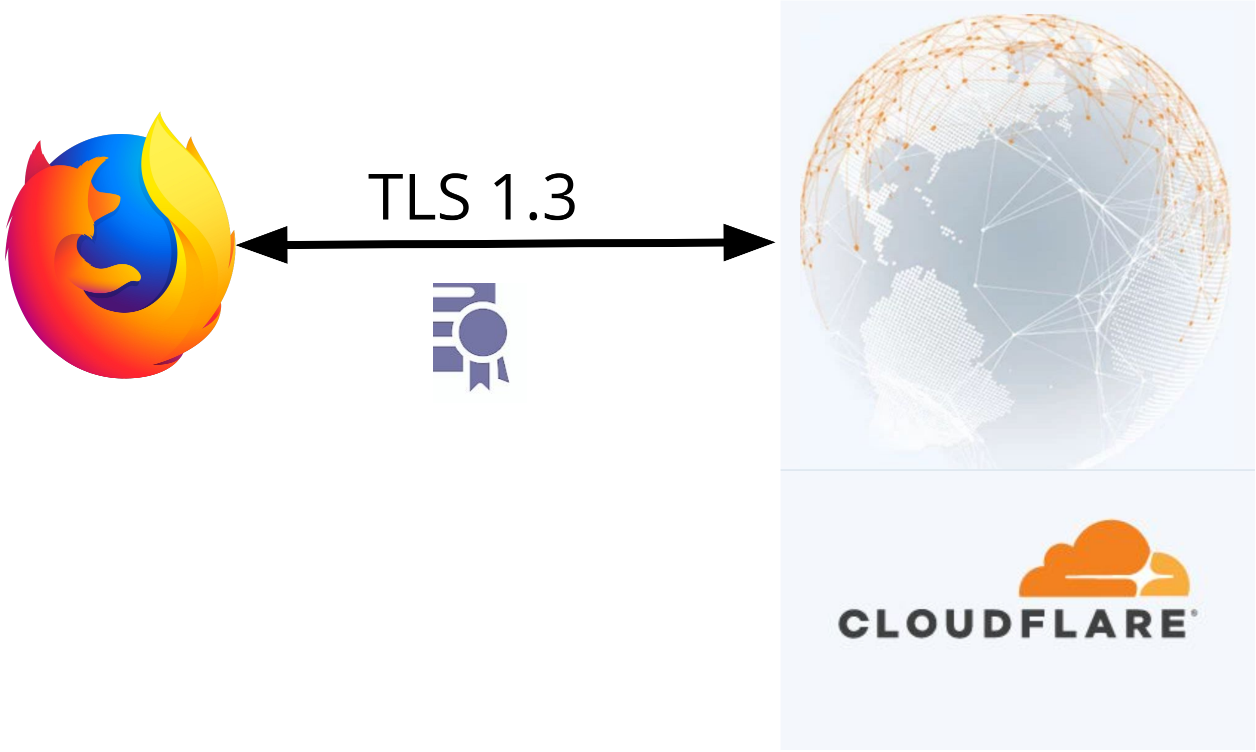
Delegated Credentials

- ❏ Server operator issues credentials within scope of certificate
- ❏ Delegated credential is bound to the delegation certificate
- ❏ Short-lived - **no longer than 7 days**



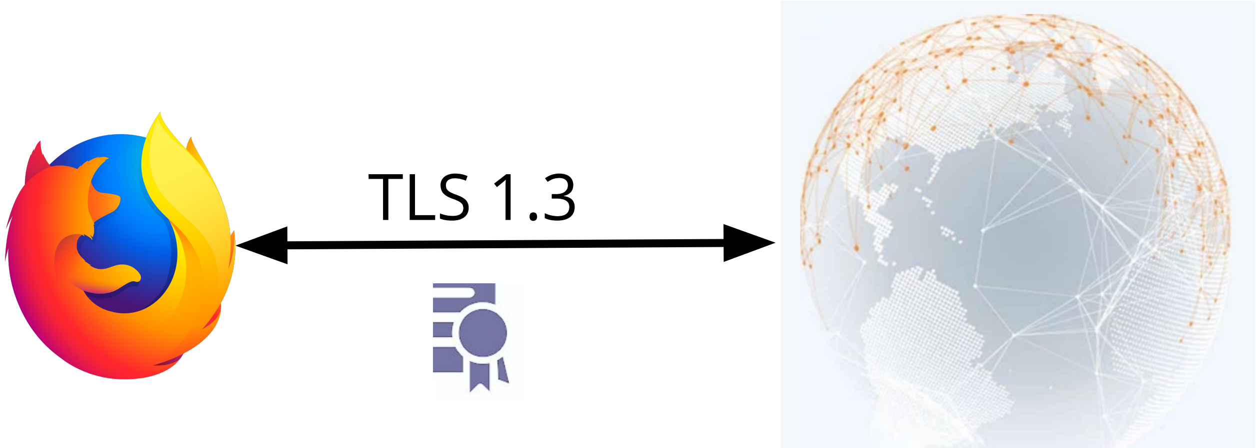
Why?

- ❑ Limited exposure
- ❑ Reduction in CA interaction overhead
- ❑ Reduction in latency



Why?

- ❑ Limited exposure
- ❑ Reduction in CA interaction overhead
- ❑ Reduction in latency



Bug 1540403

Delegated credentials (draft-ietf-tls-subcerts)

ASSIGNED Assigned to **chrispatton** ▾

Edit Bug | Copy Summary | Stop Following | ▾

Get help with this page

Mozilla Security Blog

NOV 1 2019

Validating Delegated Credentials for TLS in Firefox

Kevin Jacobs J.C. Jones Thyla van der Merwe

At Mozilla we are well aware of how fragile the Web Public Key Infrastructure (PKI) can be. From fraudulent Certification Authorities (CAs) to implementation errors that leak private keys, users, often unknowingly, are put in a position where their ability to establish trust on the Web is compromised. Therefore, in keeping with our mission to create a Web where individuals are empowered, independent and safe, we welcome ideas that are aimed at making the Web PKI more robust. With initiatives like our Common CA Database (CCADB), CRLite prototyping, and our involvement in the CA/Browser Forum, we're committed to this objective, and this is why we embraced the opportunity to partner with Cloudflare to test Delegated Credentials for TLS in Firefox, which is currently undergoing standardization at the IETF.

As CAs are responsible for the creation of digital certificates, they dictate the lifetime of an issued certificate, as well as its usage parameters. Traditionally, end-entity certificates are long-lived, exhibiting lifetimes of more than one year. For server operators making use of Content

Delegated Credentials for TLS

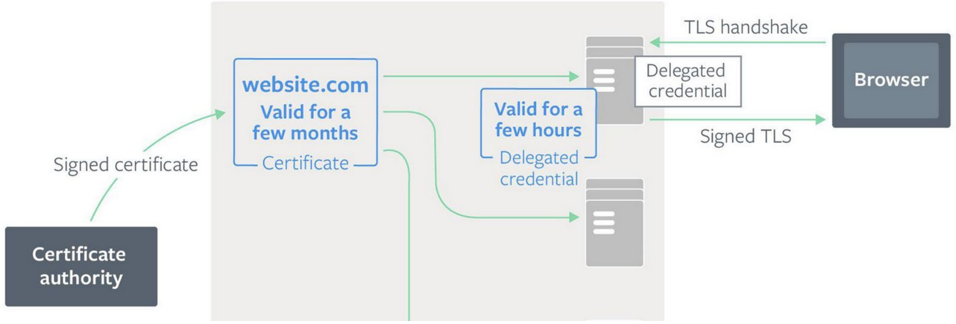
Share Like 25 Tweet

Nick Sullivan Watson Ladd

11/1/2019, 1:00:00 PM GMT

POSTED ON NOV 1, 2019 TO DATA INFRASTRUCTURE, NETWORKING & TRAFFIC, SECURITY

Delegated credentials: Improving the security of TLS certificates



5. Security Considerations

5.1. Security of delegated private key

Delegated credentials limit the exposure of the TLS private key by limiting its validity. An attacker who compromises the private key of a delegated credential can act as a man-in-the-middle until the delegate credential expires, however they cannot create new delegated credentials. Thus, delegated credentials should not be used to send a delegation to an untrusted party, but is meant to be used between parties that have some trust relationship with each other. The secrecy of the delegated private key is thus important and several access control mechanisms SHOULD be used to protect it, including file system controls, physical security, or hardware security modules.

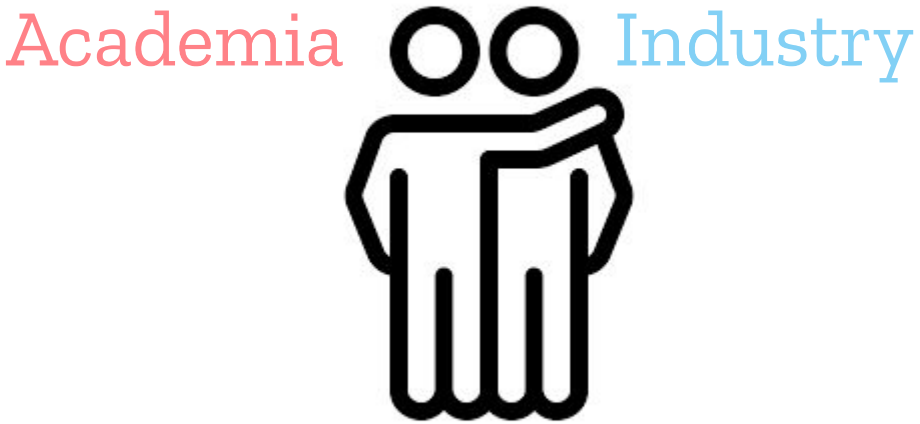
5.2. Re-use of delegated credentials in multiple contexts

It is possible to use the same delegated credential for both client and server authentication if the Certificate allows it. This is safe because the context string used for delegated credentials is distinct in both contexts.

5.3. Revocation of delegated credentials

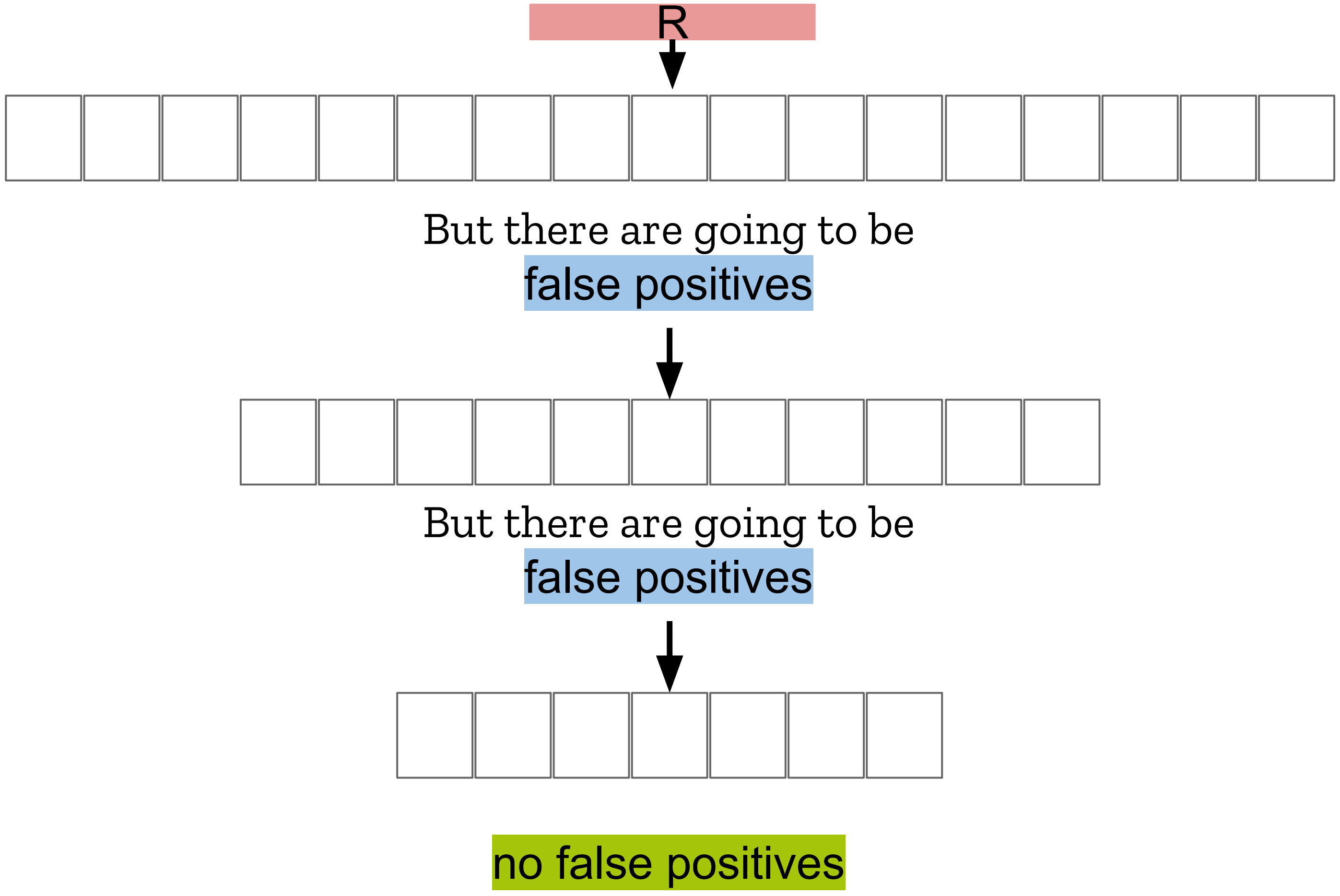
Delegated credentials do not provide any additional form of early revocation. Since it is short lived, the expiry of the delegated credential would revoke the credential. Revocation of the long term private key that signs the delegated credential also implicitly revokes the delegated credential.

Are these initiatives going to help us move towards a more robust Web PKI?



Bonus Slides

CRLite: Cascading Bloom Filters



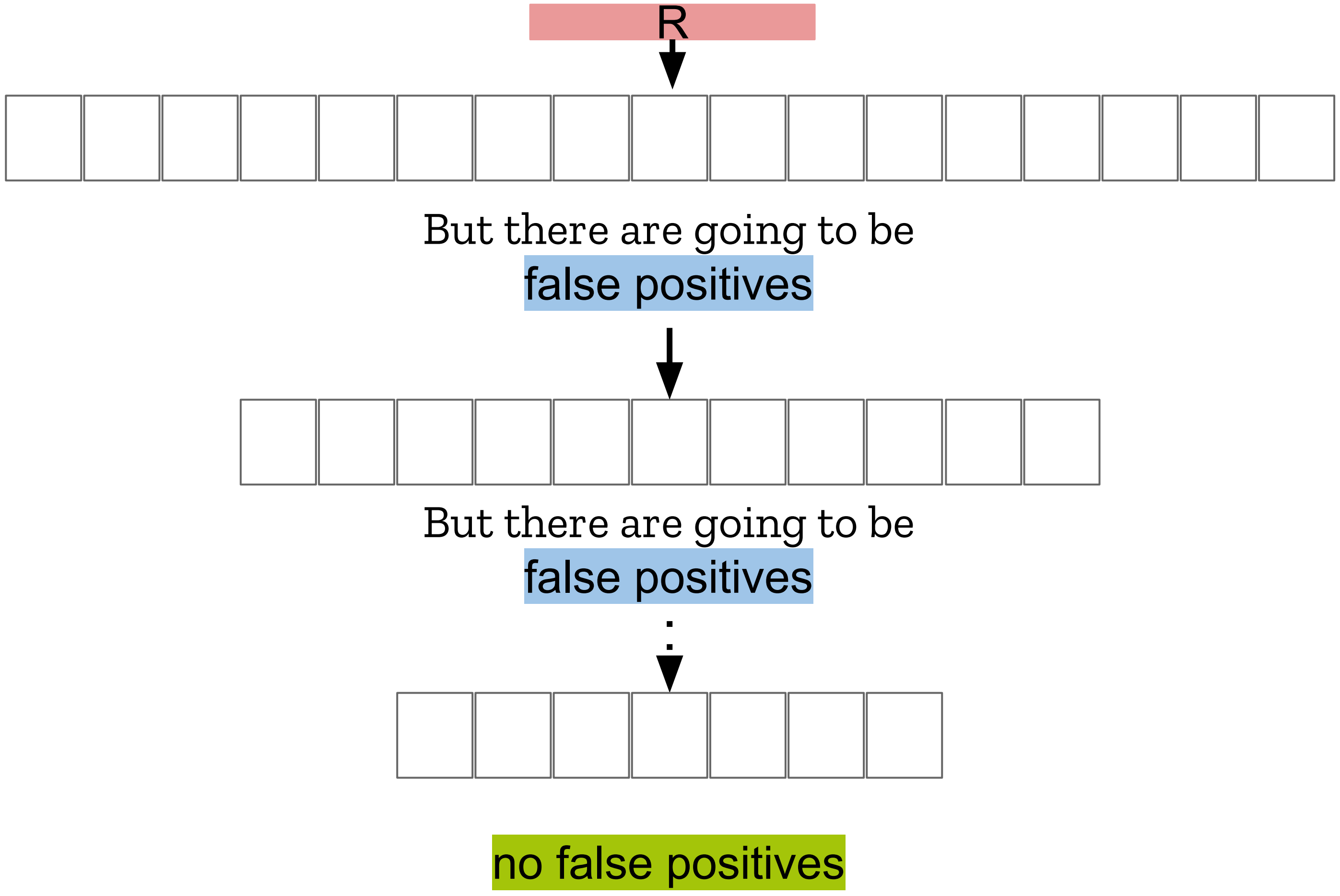
3 levels

If d^* in not in BF1, then definitely not in R. If d^* is in BF1, then we don't know.

If d^* in BF1 but not in BF2, then in R. If d^* is in BF1 and BF2, then we don't know.

If d^* in BF1 and BF2 but not in BF3, then definitely not in R. If d^* is all three, then in R.

CRLite: Cascading Bloom Filters



Is u in U in R?

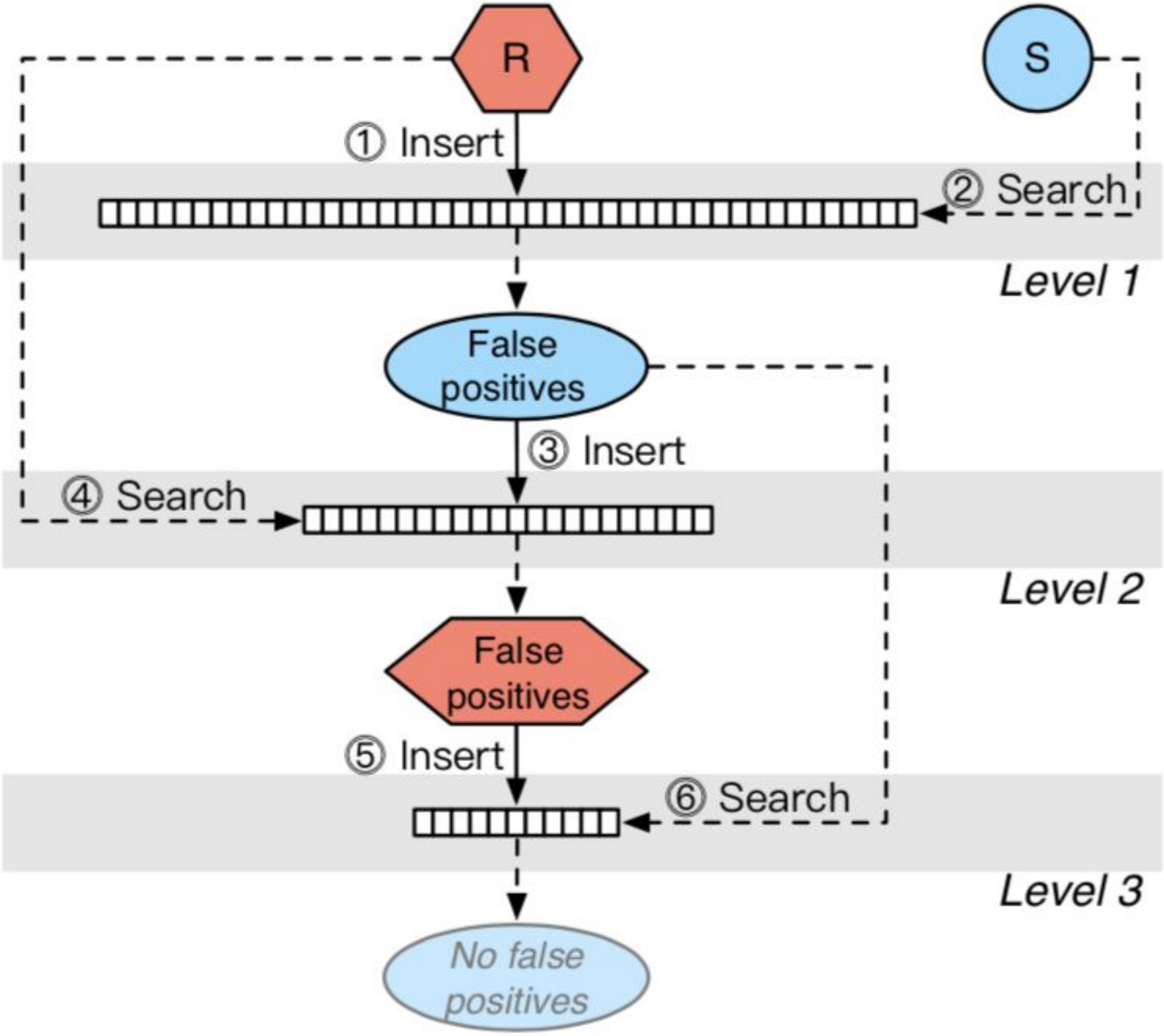
Starting at $i = 1$, keep going until ***u* not in *BF_i***.

- If i is **odd**, u **not in *R***.
- If i is **even**, u **in *R***.

If ***u* in all *BF_i*,** look at number of levels, l.

- If l is **odd**, u **in *R***.
- If l is **even**, u **not in *R***.

CRLite: Cascading Bloom Filters



If d^* in not in BF1, then definitely not in R, but not the other way round.

BF2 serves as a "blacklist" to BF1; contains items that should not be in BF1. If d^* in BF1 but not in BF2, then in R.

If d^* in BF1 and BF2 but not in BF3, then definitely not in R. Check for false positives again - only from FP1.

CRLite: A Scalable System for Pushing All TLS Revocations to All Browsers. Larisch et al, IEEE S&P 2017

CRLite: Cascading Bloom Filters

Want the minimum possible size...

Bloom filter minimized:
$k = log\_2(1/p)$ and $m \cong 1.44r\ log\_2(1/p)$

How do we set for p for filter cascades?

Analysis → p_1 for BF_1, p for other BFs

$r = |R|, s = |S|$
$p\_1 = r\sqrt{p}/s$

$p = 0.5$ → close to theoretical lower bound

Simulations confirm!

Size of R dominates, does not grow considerably with S!

CRLite: Security and Corner Cases

MITM - files are signed and timestamped by aggregator

Forcing fail-open? - CRLite allows for a fail-closed paradigm

Backdating - Signed Certificate Timestamps (SCTs) should help to guard against this

Created in the gap - `NotBefore` date should be checked and compared to filter timestamp - fall back to traditional methods