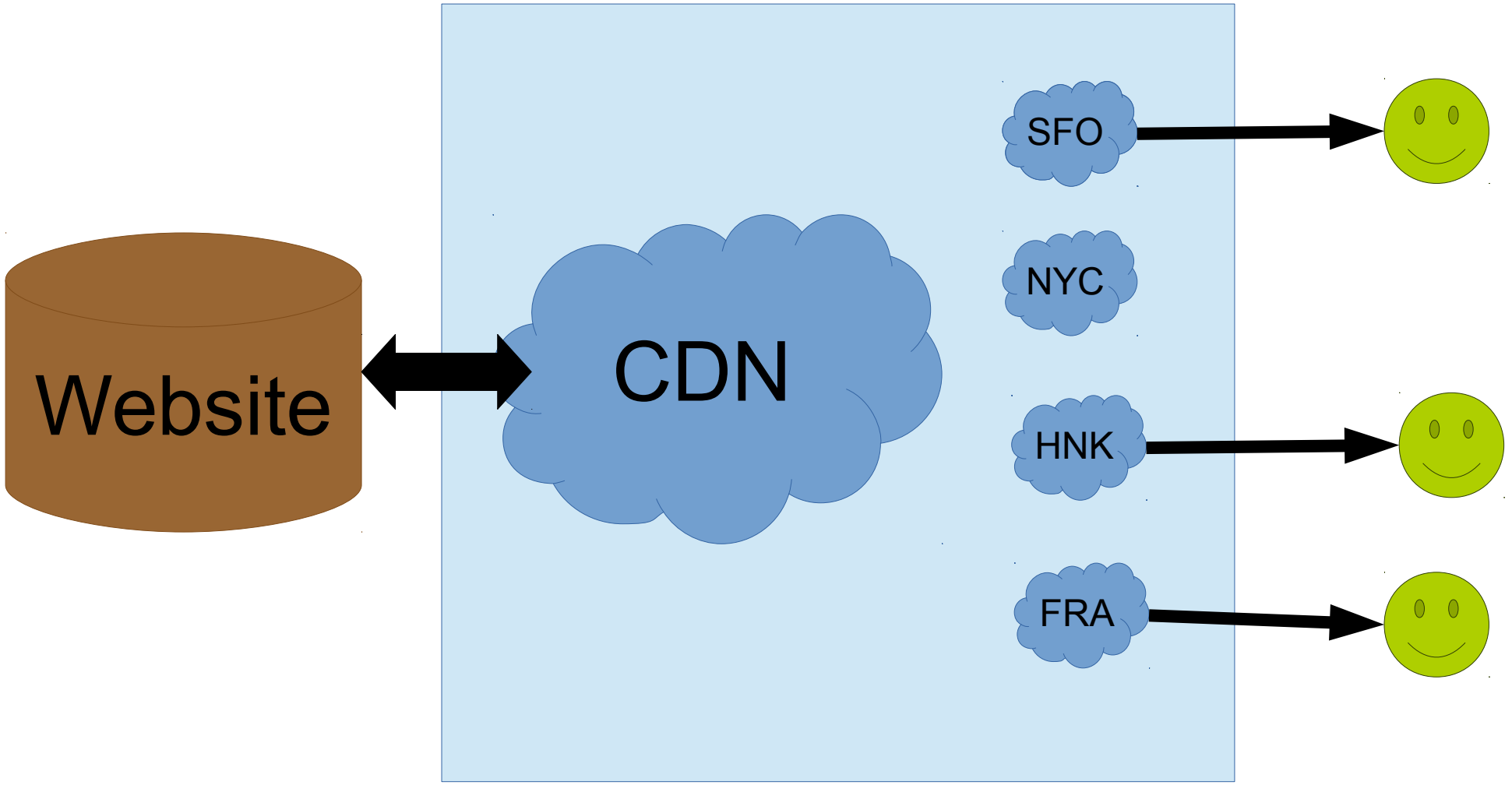
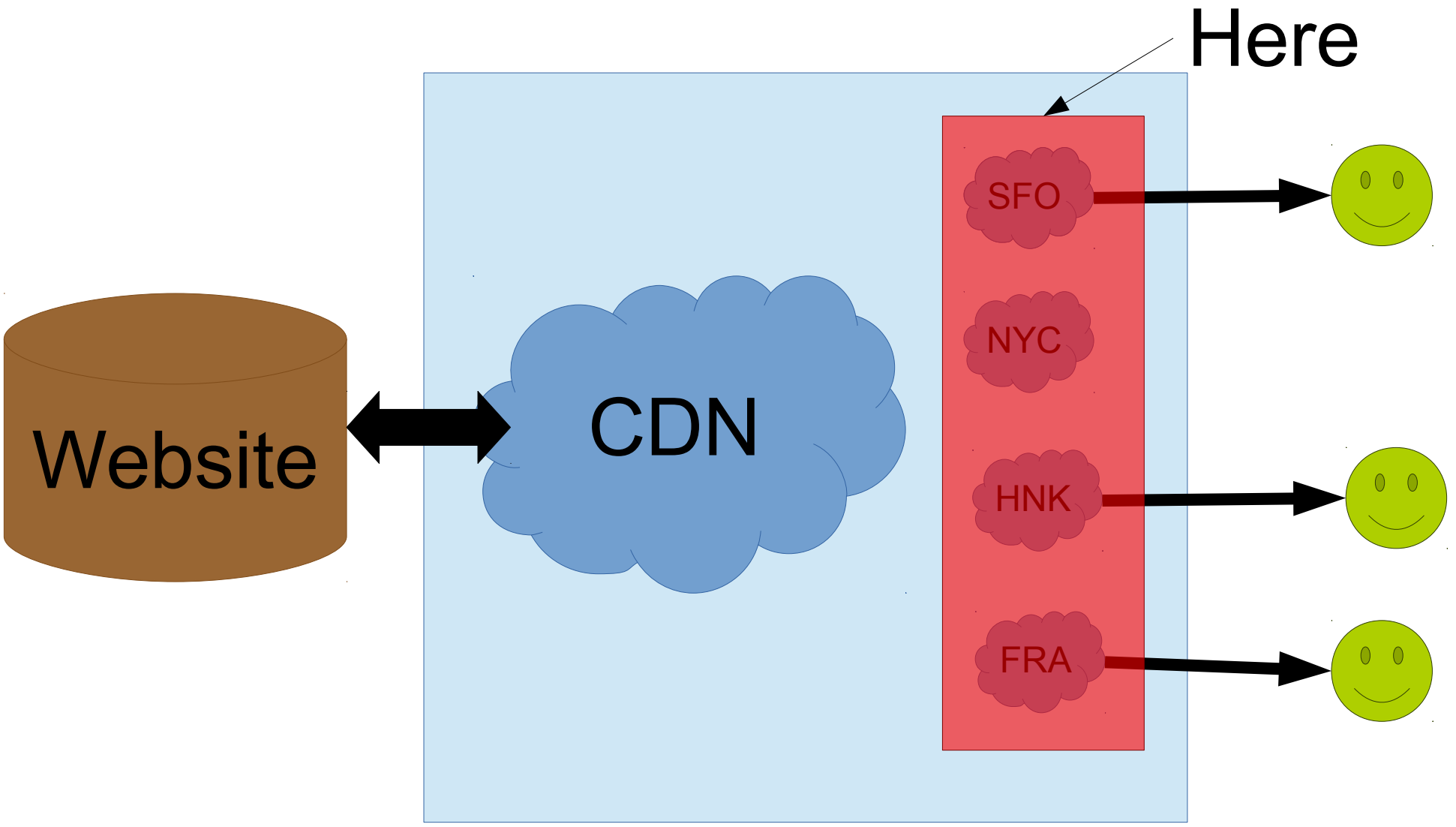
CDNs Considered Harmful

Content Distribution Network (CDN)



Where does SSL terminate?



Potential Threat

For Website

- Downsampling media
- Banner ads
- > attack surface

For End-User

- Snooping by CDN
- MITM attacks
- Disclosure to gov't

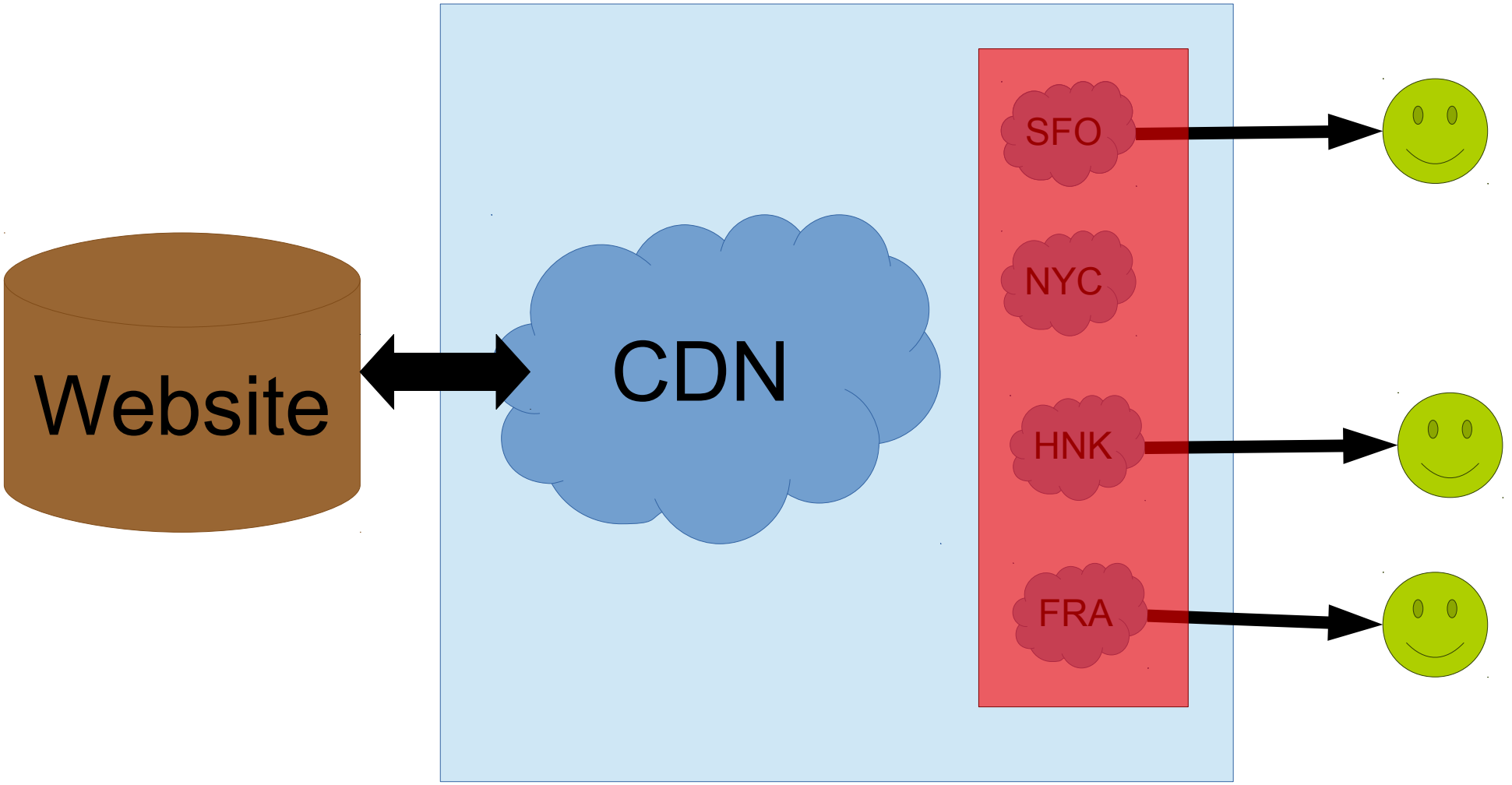
Traditional CDN Arrangements

- Large business contracts between large companies
- Lawyers involved
- Lots of $$ involved
- Result:
  - Trust
  - Disincentive for CDN to cheat

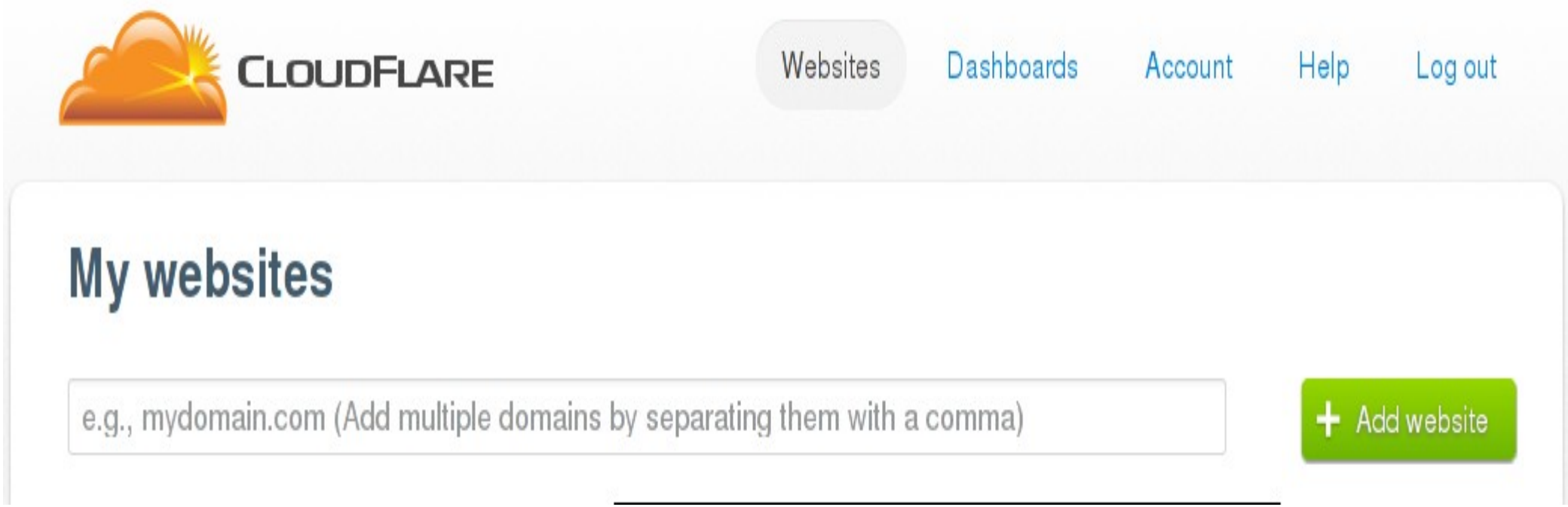
SSL Certs Generated By Owner

- CA e-mails contact from domain WHOIS
- Identity verfication for higher value certs:
  - Government identification
  - Phone call
  - Mail-in form
  - Proof of business registration
  - Etc
- So the domain owner is still in control

CDNs Today



CDNs Today


CLOUDFLARE
Websites
Dashboards
Account
Help
Log out
My websites
e.g., mydomain.com (Add multiple domains by separating them with a comma)
+ Add website

CDNs Today

Universal SSL – How?

- CloudFlare automatically generates SSL certs
- Using a “reputable” CA: GlobalSign...
- Without the domain owner's involvement

Universal SSL – How?

SSL Certificate
Alt. Names

Not Critical
DNS Name: ssl2370.cloudflare.com
DNS Name: *.jdmstyletuning.com
DNS Name: *.ywsinternational.com
DNS Name: *.iceboxintakes.com
DNS Name: *.shaleadvantage.com
DNS Name: *.topseos.in
DNS Name: *.zengarage.com.au
DNS Name: *.dnrcllc.com
DNS Name: *.sign2pay.com
DNS Name: *.tartech.net
DNS Name: *.adobegold.com
DNS Name: *.1001cocktails.com
DNS Name: *.virginiaseo.org
DNS Name: *.abacus-solutions.de
DNS Name: *.bestseos.com
DNS Name: *.immobilise.com
DNS Name: *.freshtools.ws
DNS Name: *.trinidadco.com
DNS Name: *.beamyourscreen.com
DNS Name: *.rapidform.com
DNS Name: *.bestseos-canada.com
DNS Name: *.passthepopcorn.me

Universal SSL – How?

Thought you stopped using CloudFlare?
Nup... sorry!

The Crux of the Problem

- SSL authenticates the **connection**
  - We know **who** we are connecting to
- It does not authenticate the **content**
  - We don't trust the CDN

Goals

- Thwart MITM attacks
- Retain performance/cost benefits of CDN
- Protect uncached content
- No changes to browser
- No changes to CDN

- Secondary-goal – PKI is broken anyway so let's avoid it

Proposed Solution – TPM.js

Key Idea: obtain a root of trust circumventing the CDN and leverage it to verify CDN-cached assets.

1) End-user gets /index.html directly from website
   - Contains embedded public key
   - Contains bootstrap javascript (TPM.js) for loading other assets
   - Long-term client-side caching
2) End-user loads signed assets from CDN
3) TPM.js verifies signature, extracts raw data and loads content

Open Questions

- Rotating public keys
- Encrypting sensitive content
- Performance
- Etc...
- Appropriate backronyms for “TPM”