Sign your SBOM…
With WHAT?

Trusting the software supply chain
2024-12-16 / 2026-01-20
@oej@infosec.exchange

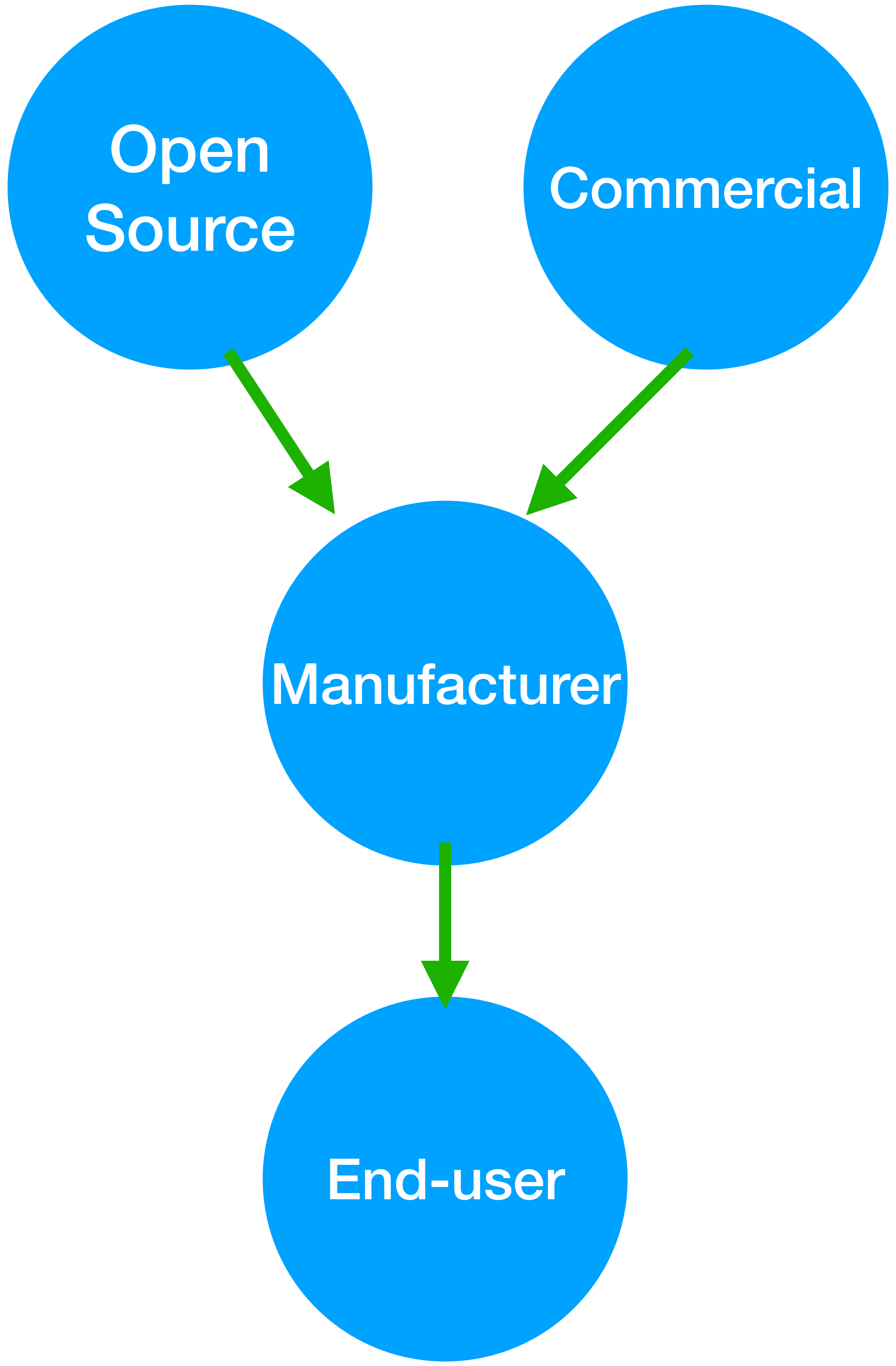
SBOM

SIGNED
VERIFIED
TRUSTED

OWASP®

The problem as I see it

- Many customers need to trust many vendors, much like web trust
- But web trust is broken
- Commercial software signing certs cost too much
- **As the SBOM is exchanged across the software supply chain, we need to implement trust and transparency.**



Aaaaaaaaargh

TEA

OWASP TRANSPARENCY EXCHANGE API

OWASP SCVS talks about signing SBOMs, validating

…but not about how and with what…

ENISA SBOM landscape (review)
talks about digital signatures for SBOMs and trust…

But not how and with what.

OWASP TEA
has some ideas published by OEJ

…but not a lot of feedback….

https://scvs.owasp.org/

https://github.com/cyclonedx/transparency-exchange-api

HELP ME!

Looking for a solution

SIGN

VALIDATE

TRUST

- We need to be able to sign, validate and trust digital signatures on documents from Open Source projects as well as commercial vendors

- How do we build such a trust system without a single point of failure, without a central actor?

Transparency is important

- *"How do I trust that the vendor did not change the SBOM without telling me?"*
- Someone needs to monitor the actors, much like certificate transparency
- It's not clear who that actor is

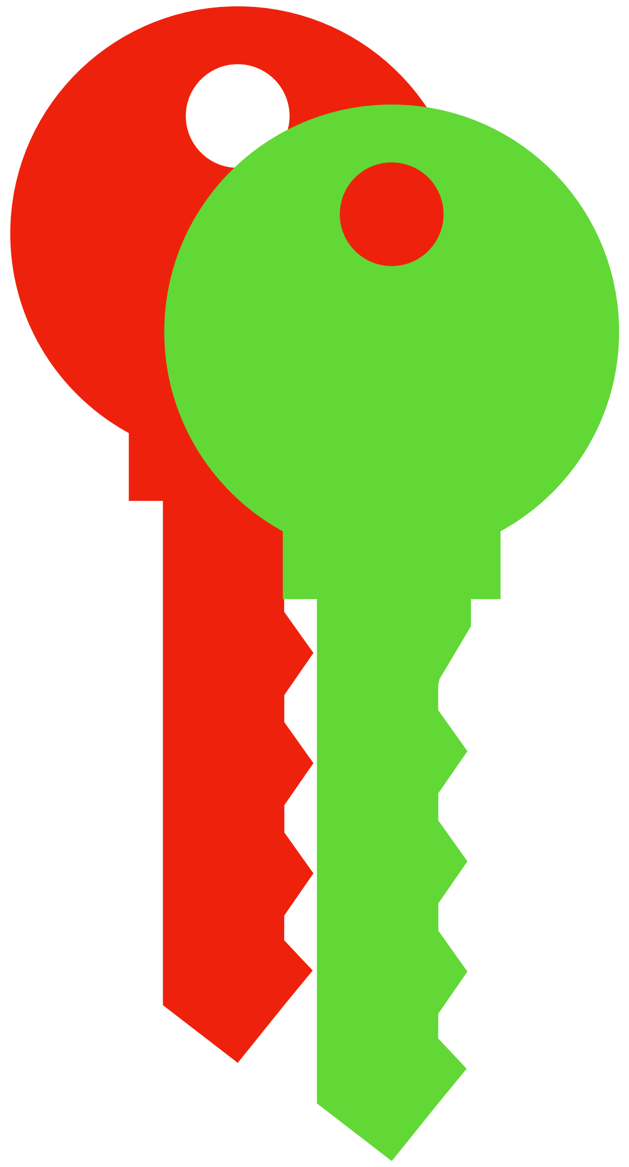
A global problem

Business manufacturers

Open Source projects and stewards

Sigstore is cool

- Based on another ID, like OpenID connect auth, creates a **temporary key pair and a certificate**
- This key pair is used for signing - a commit, container or something else
- Proof is stored in a signed log for transparency and verification
- The certificate expires within minutes and the key pair is no longer valid, can be deleted and forgotten
- *Signing with key and cert, but almost no key management needed*



…but Sigstore is not a solution for everyone

- Sigstore (the service) works well for some Open Source projects
    - *Integration to Github and more*
- But it’s not a solution for everyone
    - *Privacy - stores email addresses in US cloud*
    - *Won’t be accepted by commercial actors*
- The Open Source software is hard to install and put in production

Letsencrypt turned off signing

- A free PKI for Web certificates - enabling TLS everywhere at no cost
- Killing business for a lot of commercial Certificate Authorities
- Funded by large companies

- Only US based system
- Turned off signing in key usage bits

Let's Encrypt

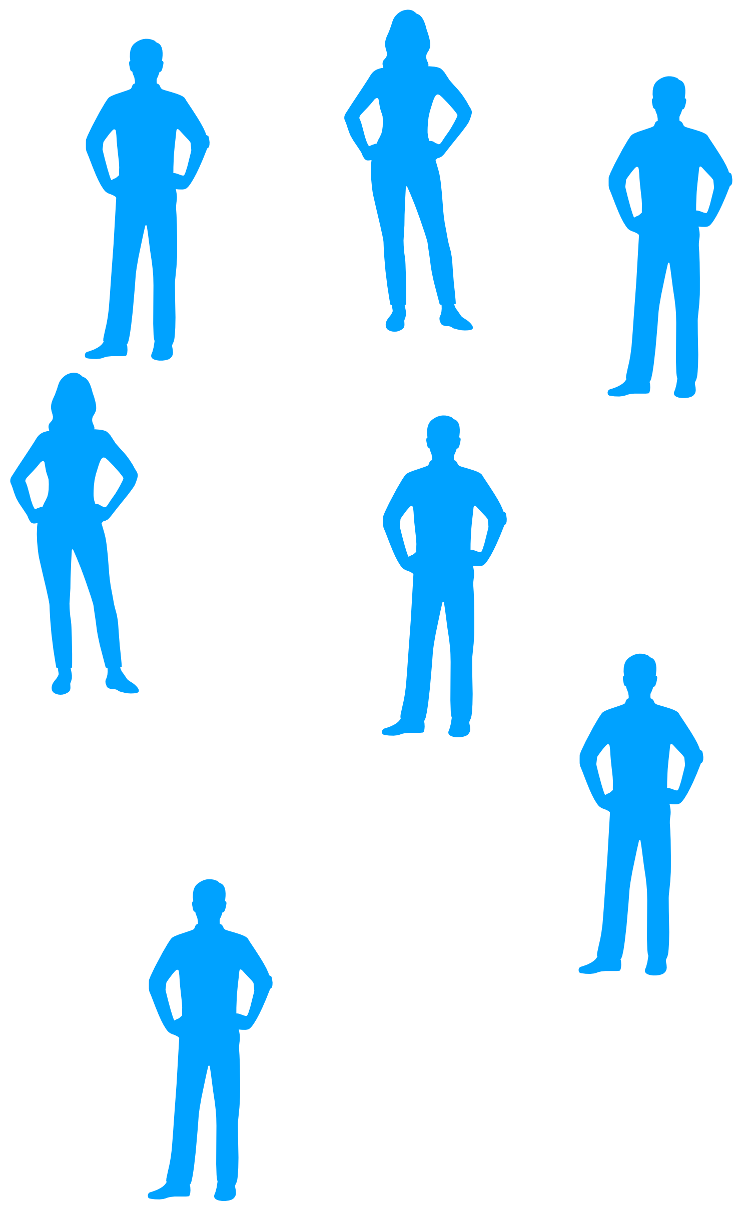
Documentation Get Help

Encryption for Everybody

A nonprofit providing free TLS certificates to more than 700M websites.

Open Source use PGP/GPG

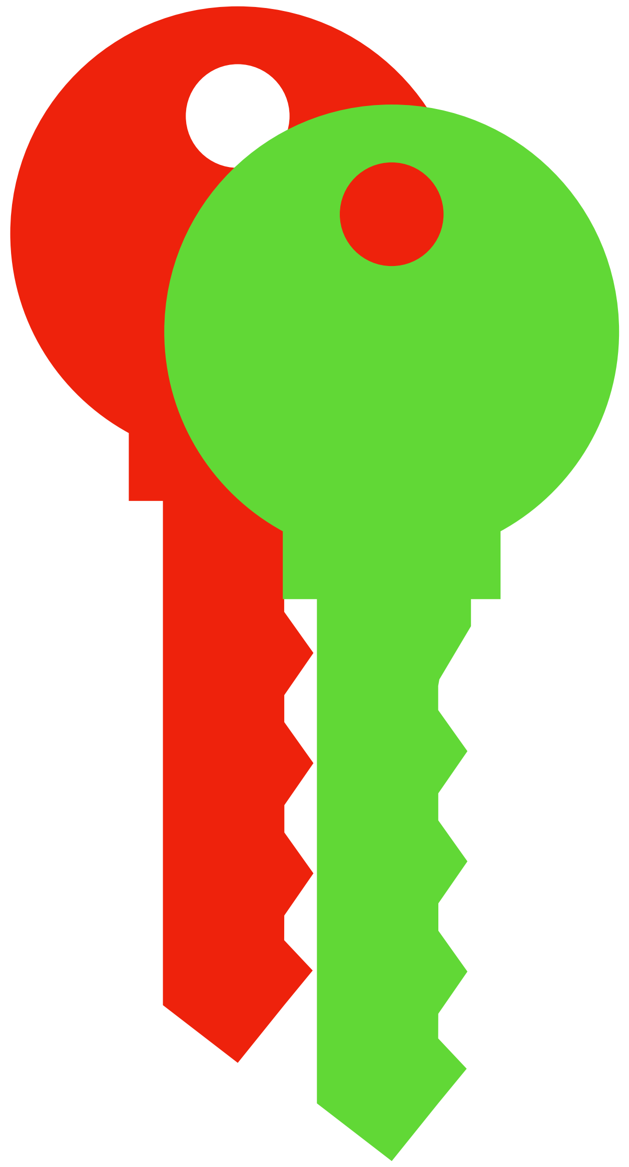
- A web of trust which is not really used
  - *Very few keys are signed by trusted parties*
- A single key pair with limited trust anchors
- Best current practise is not well documented
  - *"Download this file and add blindly to your key store as a trusted key"*
  - *NO FINGERPRINT GIVEN*
- Technology is not the primary problem



FOSDEM used to have huge key signing events

Managing a private PKI is hard

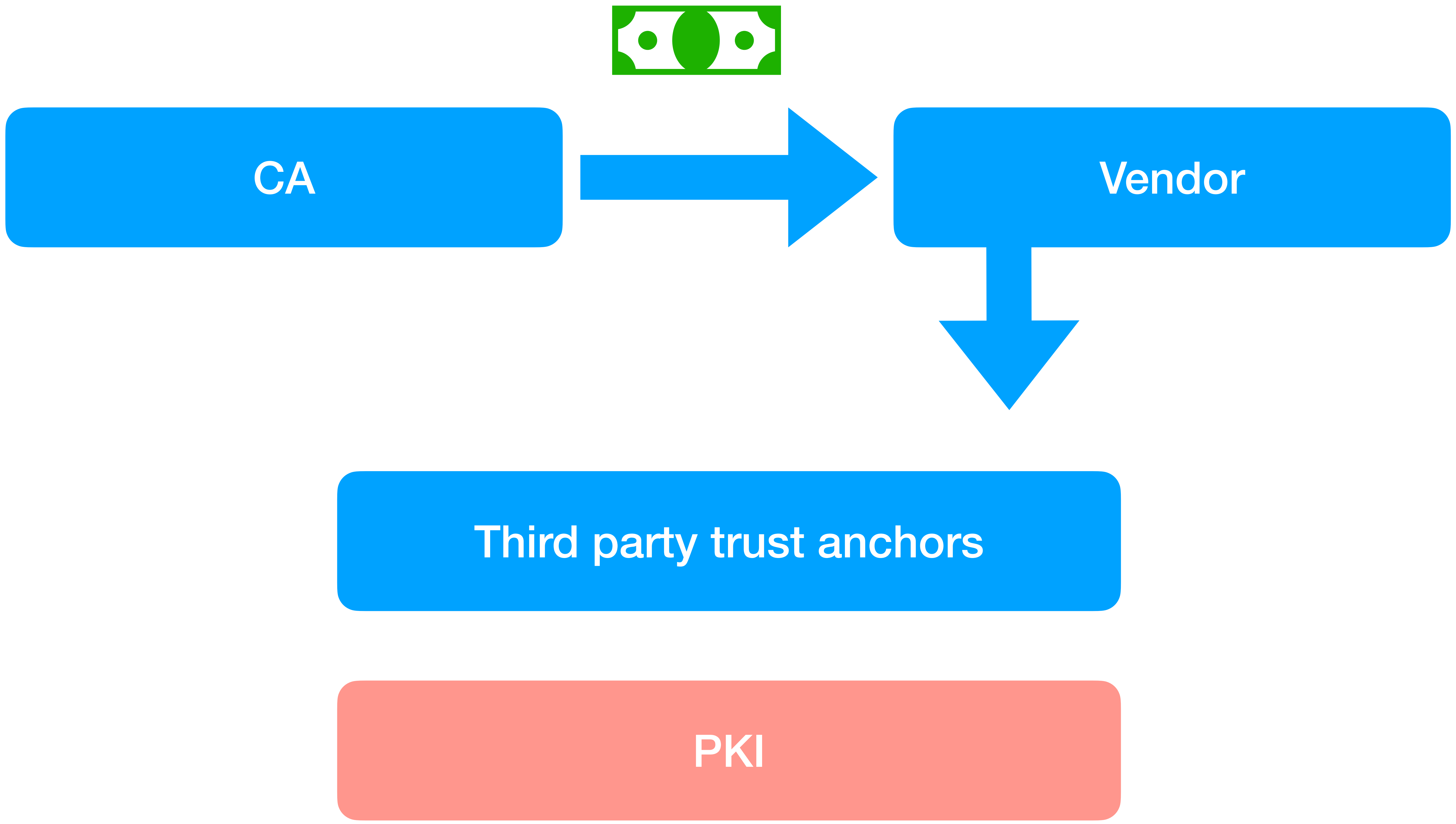
- Protecting a private key from abuse is hard
- If it’s stored on a HSM - it’s a physical device to protect
- Backup is needed, but needs protection
- Needs some kind of policy in an organisation



Trust is earned

Trust on first use	Third party trust anchor	Web of trust
SSH	PKI	GPG

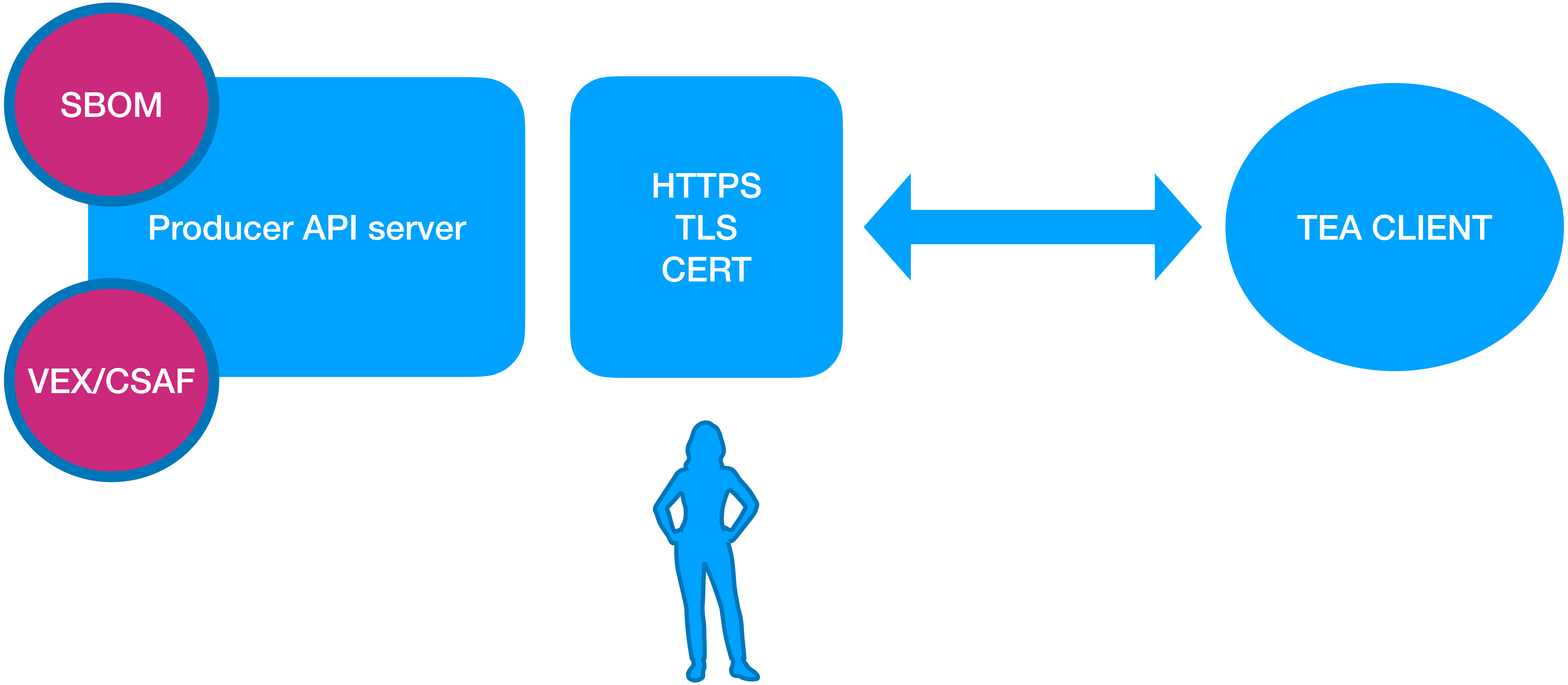
Trust is earned - but in the web it is forced upon us



Firefox Certificate Validations by CA
Google Trust Services LLC
DigiCert
Internet Security Research Group
Amazon Trust Services
Entrust
Sectigo
GlobalSign nv-sa
GoDaddy
Asseco Data Systems S.A.
SECOM Trust Systems CO.
Deutsche Telekom Security GmbH
Government of Turkey
SSL.com
Certigna
Buypass
IdenTrust Services
SwissSign AG
QuoVadis
Other
0
20B
40B
60B
80B
100B
120B
140B
160B

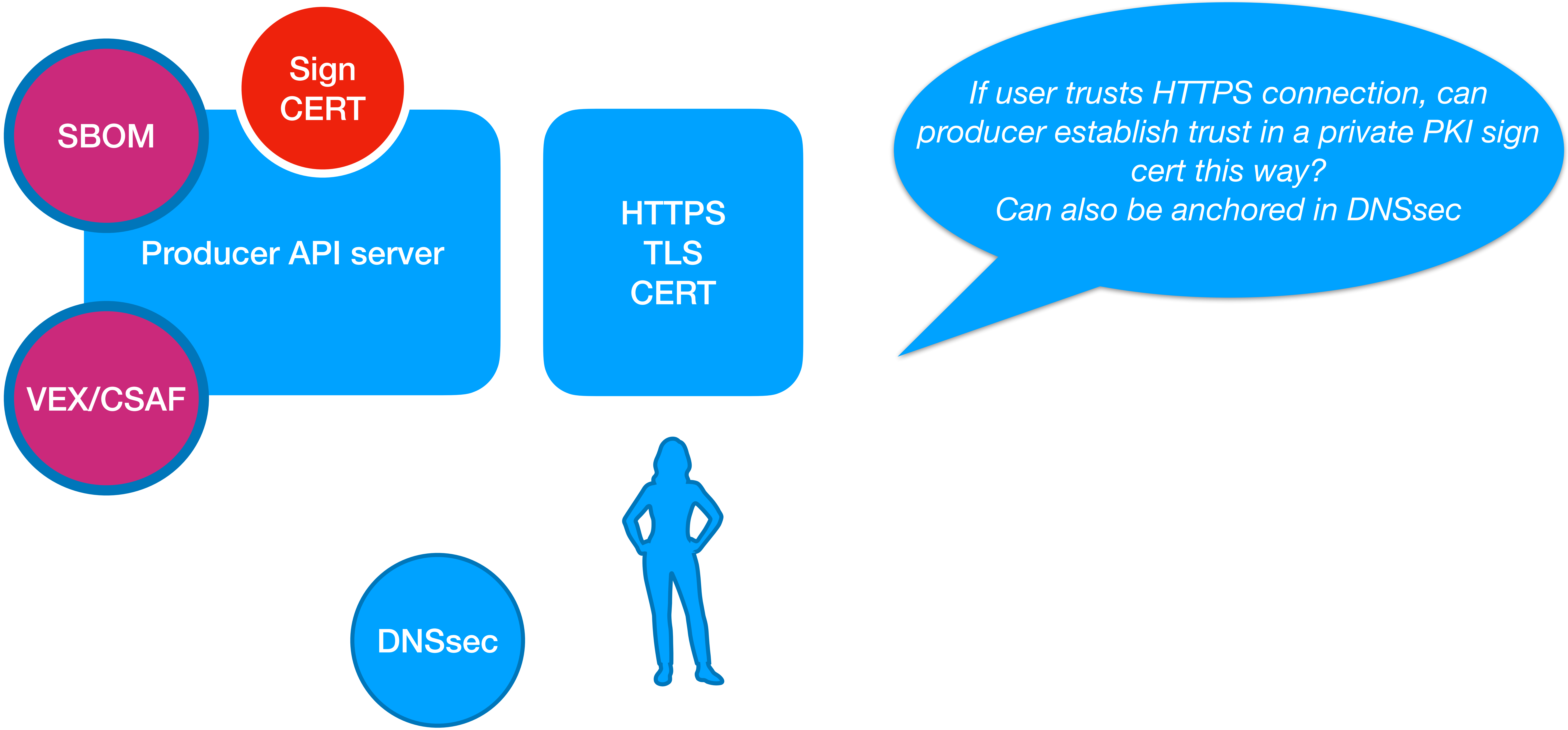
https://github.com/mcpherrinm/cert-count/blob/main/CA.svg

Example: Transparency Exchange API



The user must trust the HTTPS connection

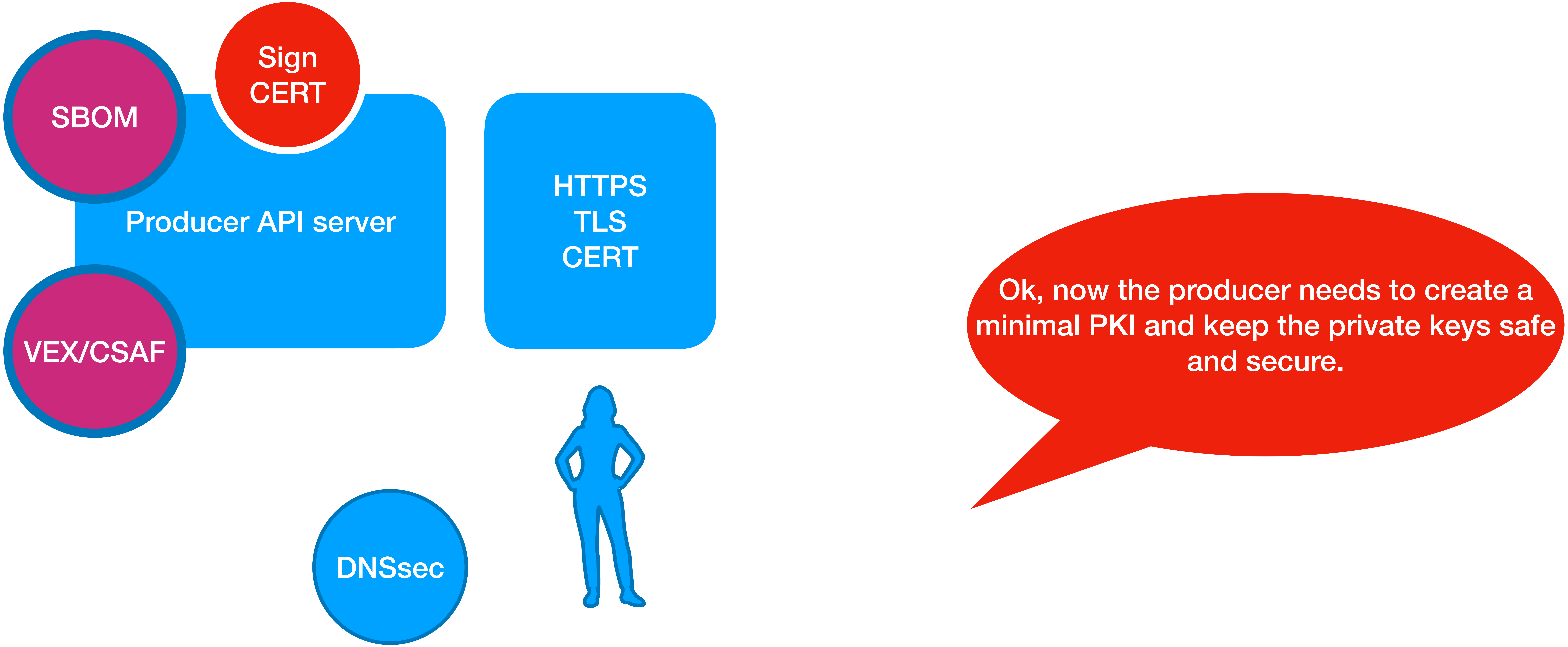
From HTTPS cert to sign cert



If user trusts HTTPS connection, can producer establish trust in a private PKI sign cert this way?
Can also be anchored in DNSsec

The user must trust the HTTPS connection.
Producer may add anchor in DNSsec

And a new PKI is born!


SBOM
Sign CERT
Producer API server
VEX/CSAF
HTTPS TLS CERT
DNSsec
Ok, now the producer needs to create a minimal PKI and keep the private keys safe and secure.

The user must trust the HTTPS connection.
Producer may add anchor in DNSsec

Can this be simplified?

My wishlist

- Simple key management
- Simple trust anchor establishment (PKI CA root)
- Distributed, not centralised
- With respect of privacy

Some cool stuff

- We need more dancers: https://datatracker.ietf.org/group/dance/about/
  DANE Authentication for Network **Clients** Everywhere (dance)

- Certs and trust anchor selection in DNSsec
  https://datatracker.ietf.org/wg/dane/documents/

- Public transparency log of signatures
  https://www.sigsum.org/

- Sigstore: https://www.sigstore.dev/
  OpenSSF project

- TEA, OWASP Transparency Exchange API
  https://github.com/CycloneDX/transparency-exchange-api

Let’s work on this!

- Mastodon @oej@infosec.exchange
- Matrix oej@matrix.org
- LinkedIn (QR)
- Email oej@edvina.net
- Github oej