TLP = CLEAR

Scaling CSAF

Building a Trusted Provider Network for ~~40~~ 45+ Vendors

Munich, December 13th, 2024

CERT@VDE
Web: https://certvde.com
Email: info@certvde.com

Christian Link
Information Security Manager

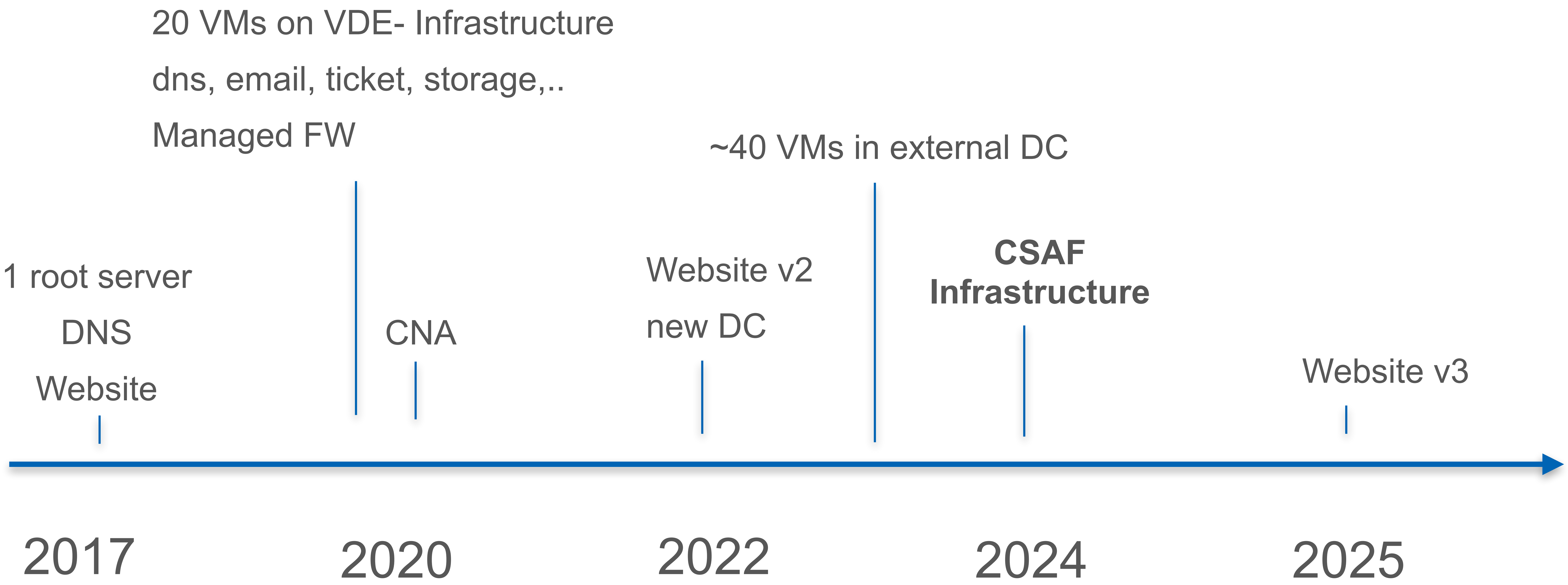
Presented by Jochen Becker

Scaling CSAF

Agenda

- About us
- CSAF Lifecycle
- CSAF Requirements for Distribution
- IT Requirements
- Organisation, Deployment, Configuration and Monitoring
- Status and Challenges
- Q & A

About us - infrastructure history



2017

1 root server
DNS
Website

2020

20 VMs on VDE- Infrastructure
dns, email, ticket, storage,..
Managed FW

CNA

2022

Website v2
new DC

~40 VMs in external DC

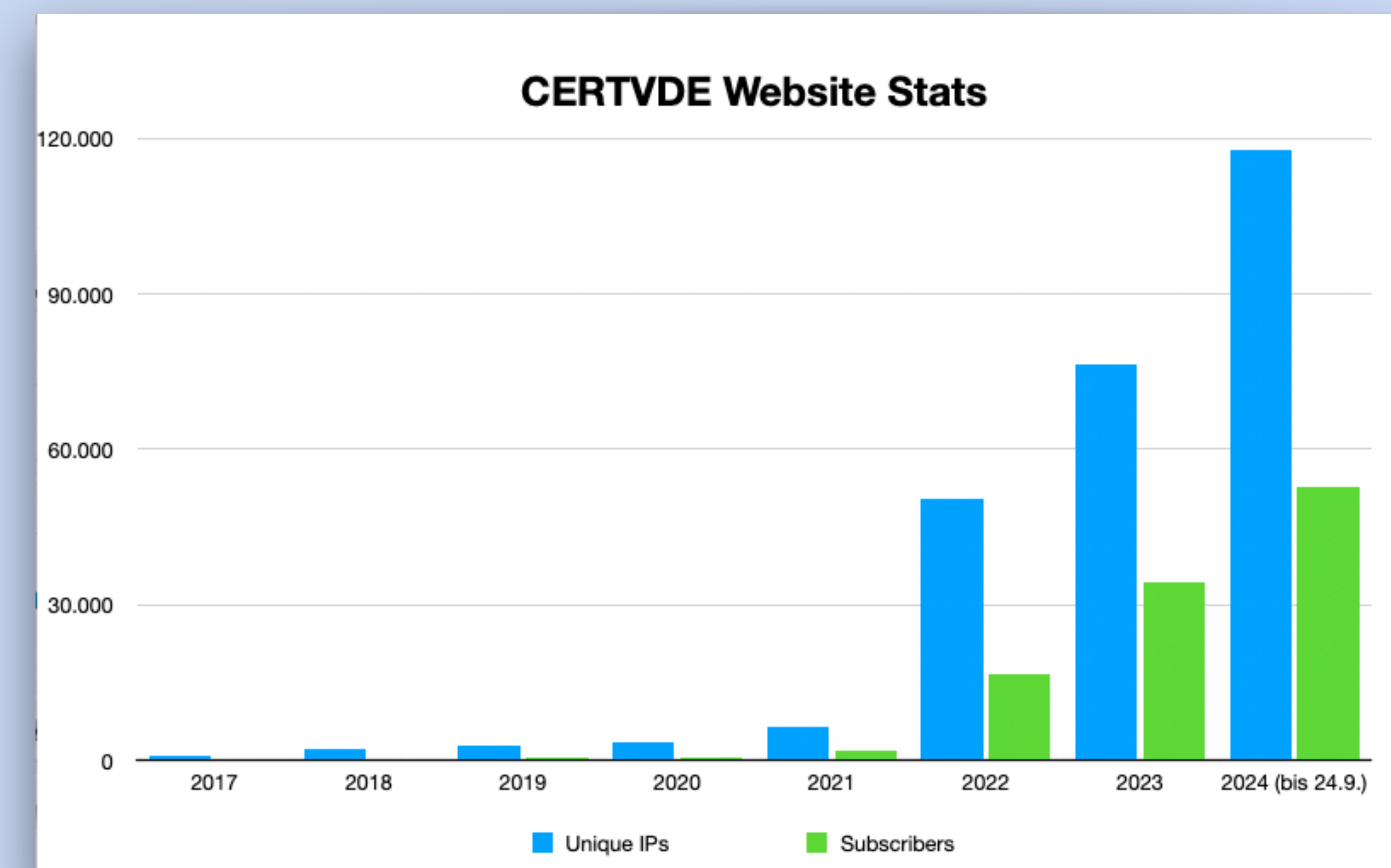
2024

CSAF Infrastructure

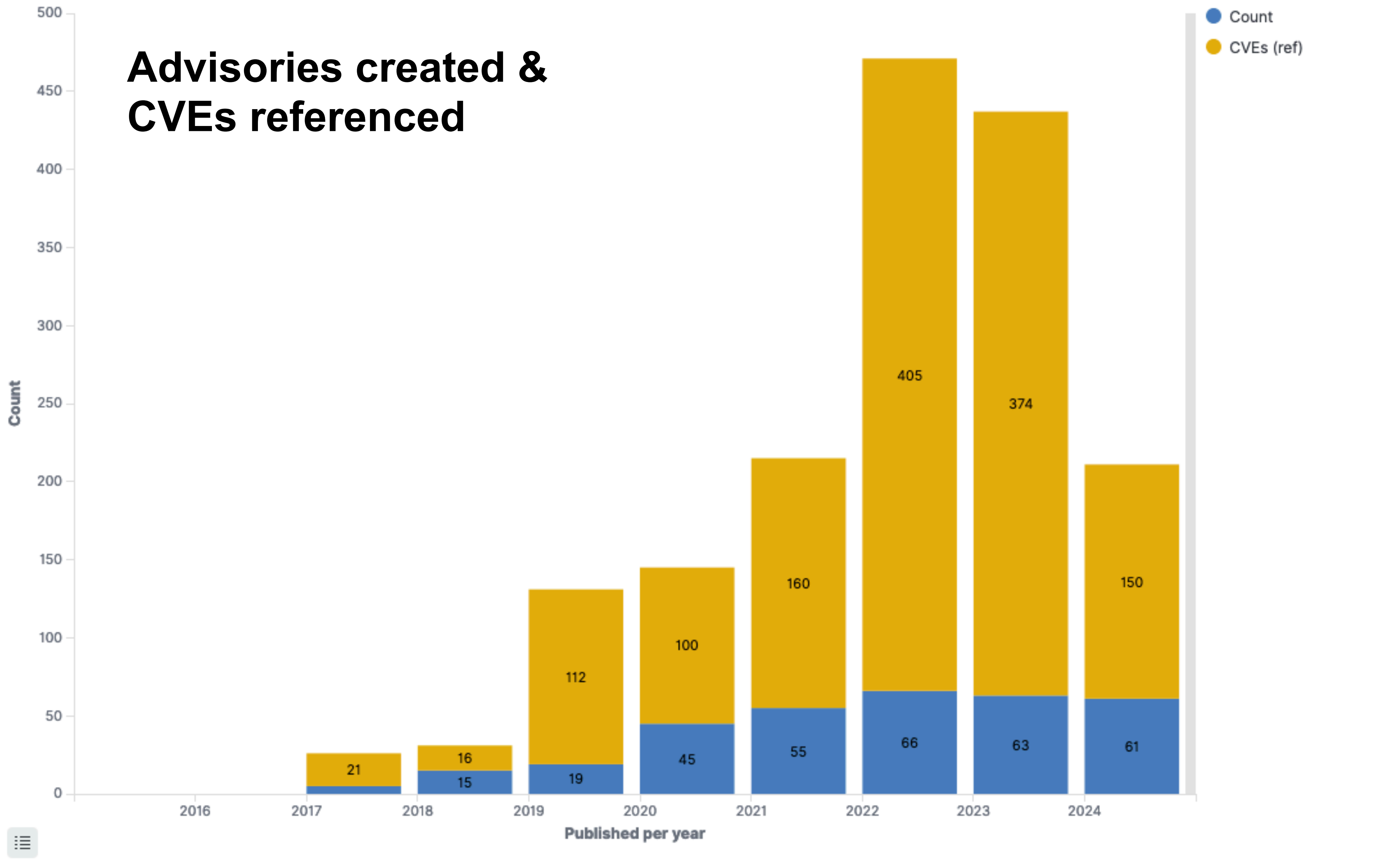
2025

Website v3

About us - Traffic



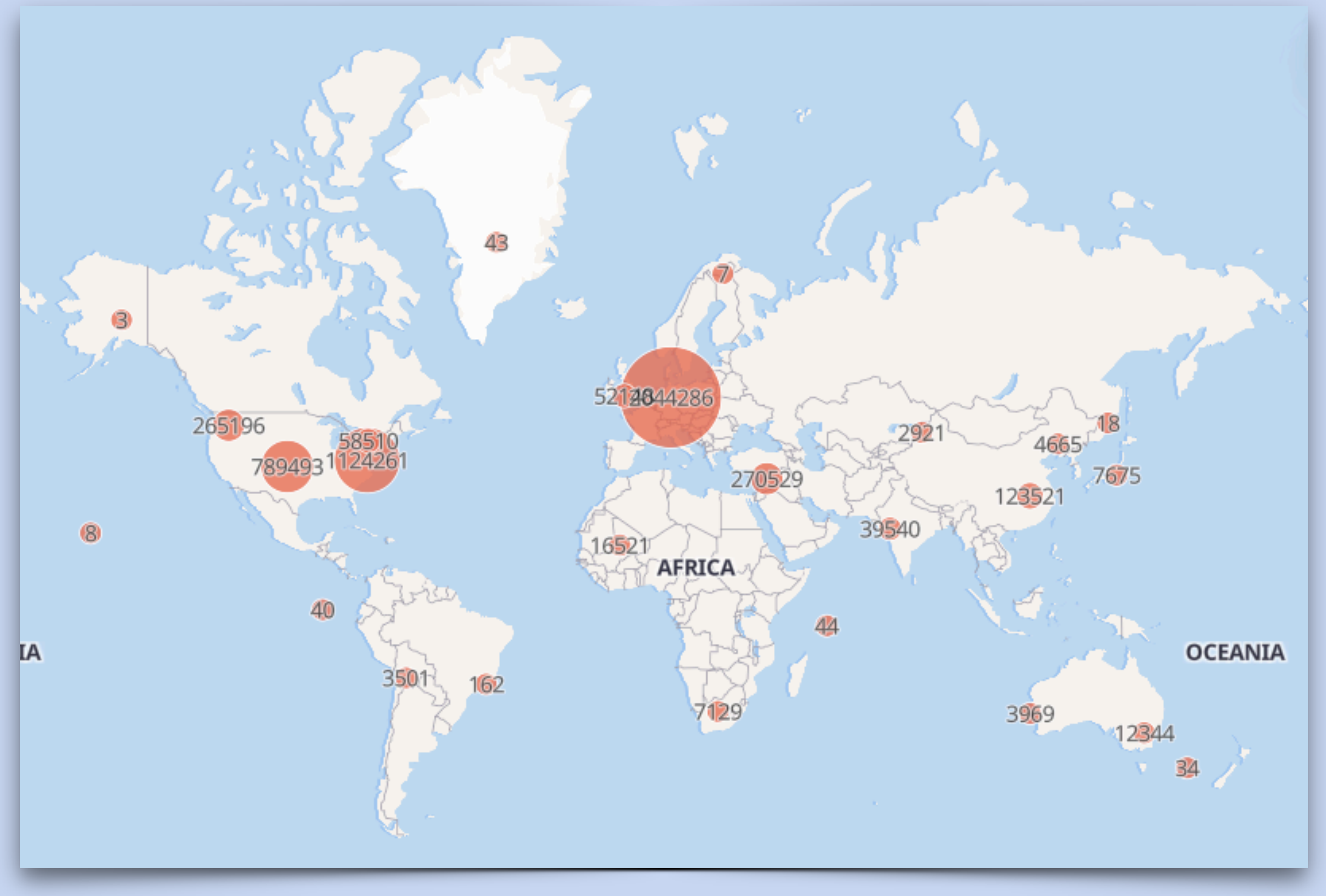
CERTVDE Website Stats

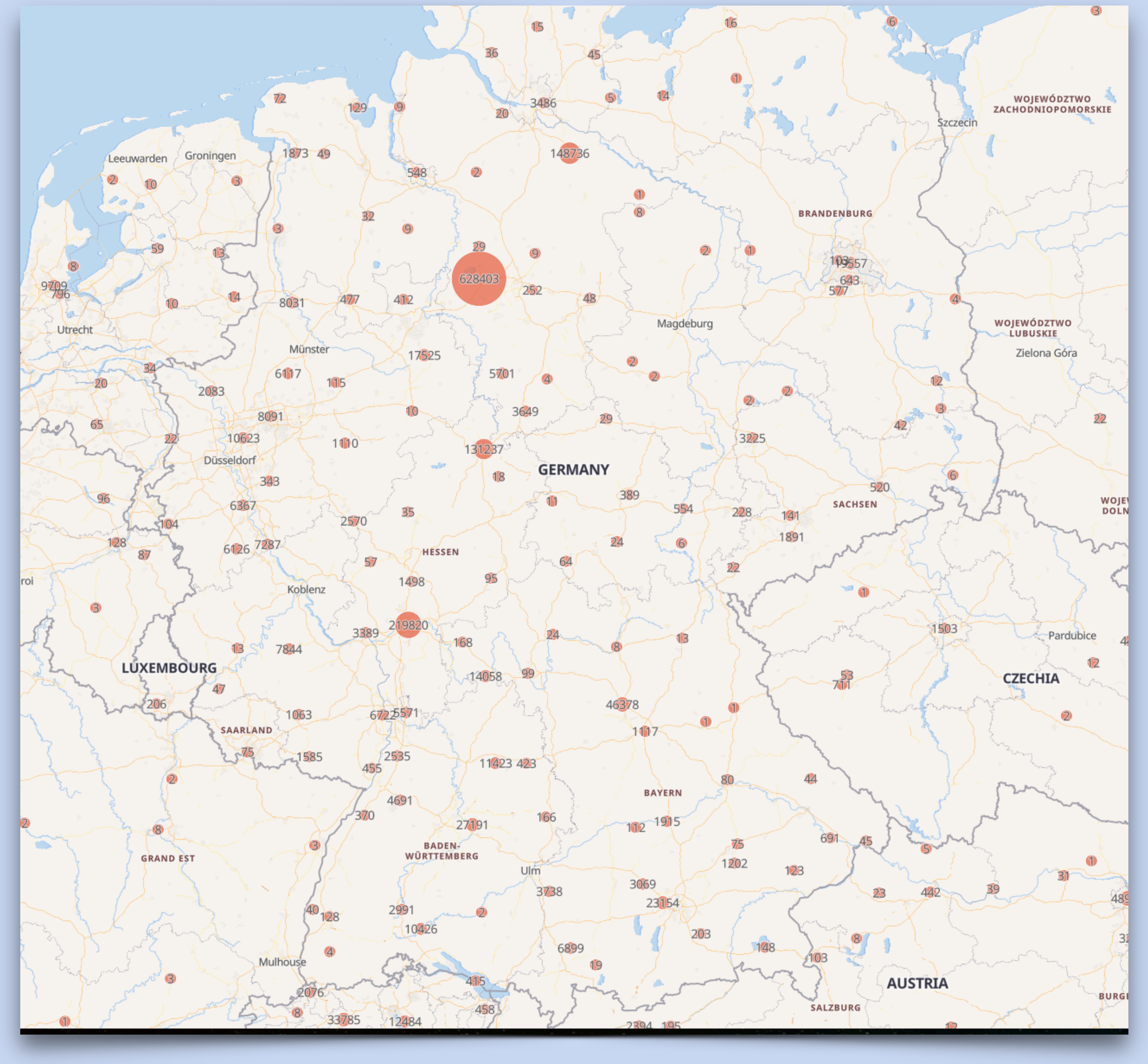


Advisories created & CVEs referenced

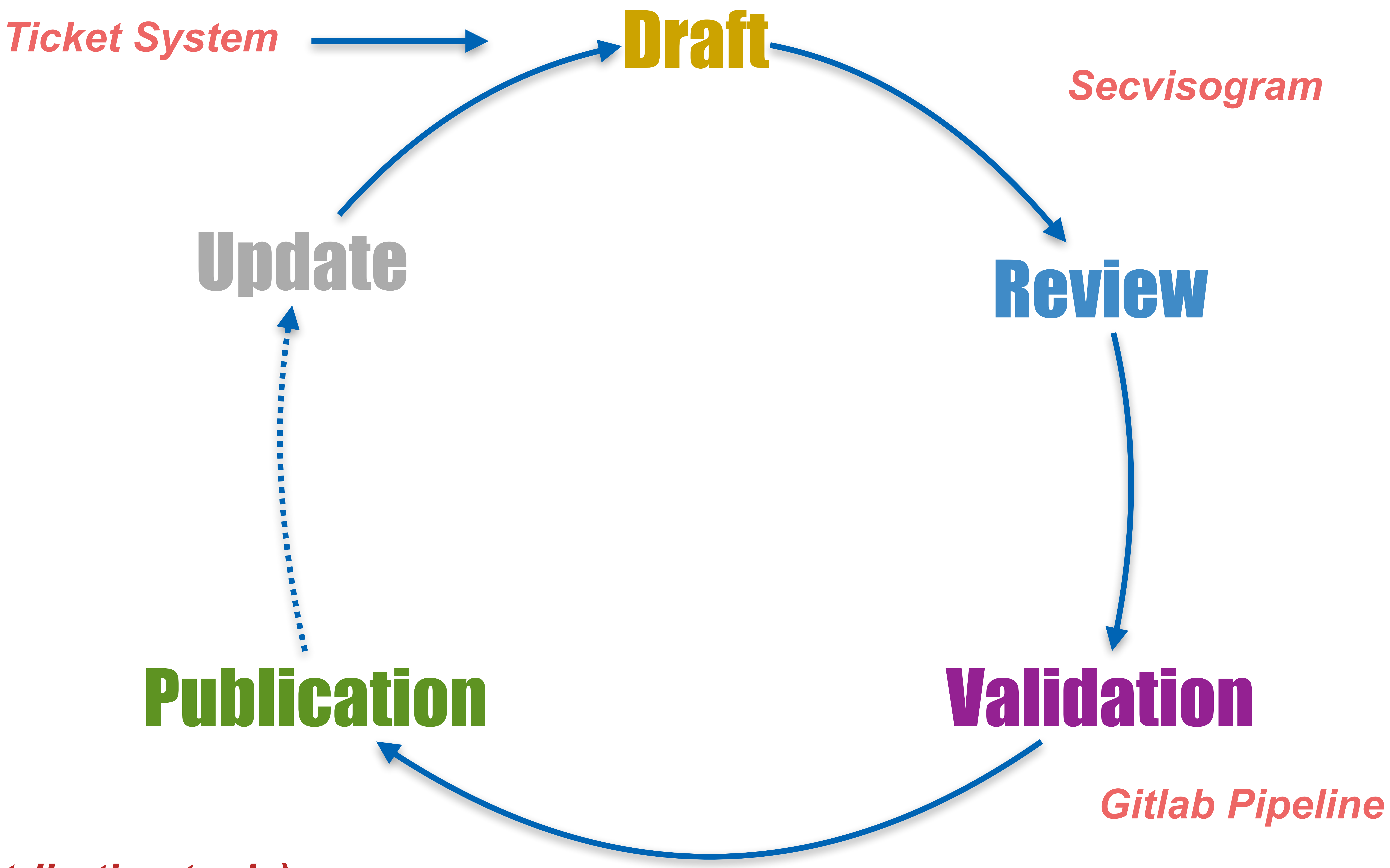
***CSAF**: ~5000 unique visitors in Q3 2024*

About us - Traffic Sources





(Our) CSAF Lifecycle



gocsaf/csaf
(ex. CSAF distribution tools)

What is gocsaf/csaf_provider?

"...is an implementation of the role CSAF Trusted Provider, ~~also offering a simple HTTPS based management service.~~"

https://github.com/gocsaf/csaf

- creates the file structure and necessary files and indices
- automatically signs
- handles rollie feed
- (together with nginx) handles client tls-auth
- ...

CSAF-Trusted Provider Requirements in detail

Draft

7.1.1: Valid CSAF document
7.1.2: Filename (see 5.1)

~~7.1.5: TLP:AMBER + TLP:RED access protection~~
7.1.21-7.1.23: Aggregator

Publication

7.1.3: TLS
7.1.4: TLP:WHITE
7.1.6: No Redirects
7.1.7: provide provider_metadata.json
7.1.8: provide security.txt
7.1.9: well-known URL for provider-metadata.json
7.1.10: DNS path (csaf.data.security.domain.tld => provider_metadata.json)
7.1.11: one folder per year
7.1.12: index.txt
7.1.13: changes.csv
7.1.14: directory listings
7.1.15-7.1.17: ROLIE feed, service and category documents
7.1.18: Integrity (HASH)
7.1.19: Signatures
7.1.20: Public OpenPGP-Key

csaf_provider
https://github.com/gocsaf/csaf

x45+

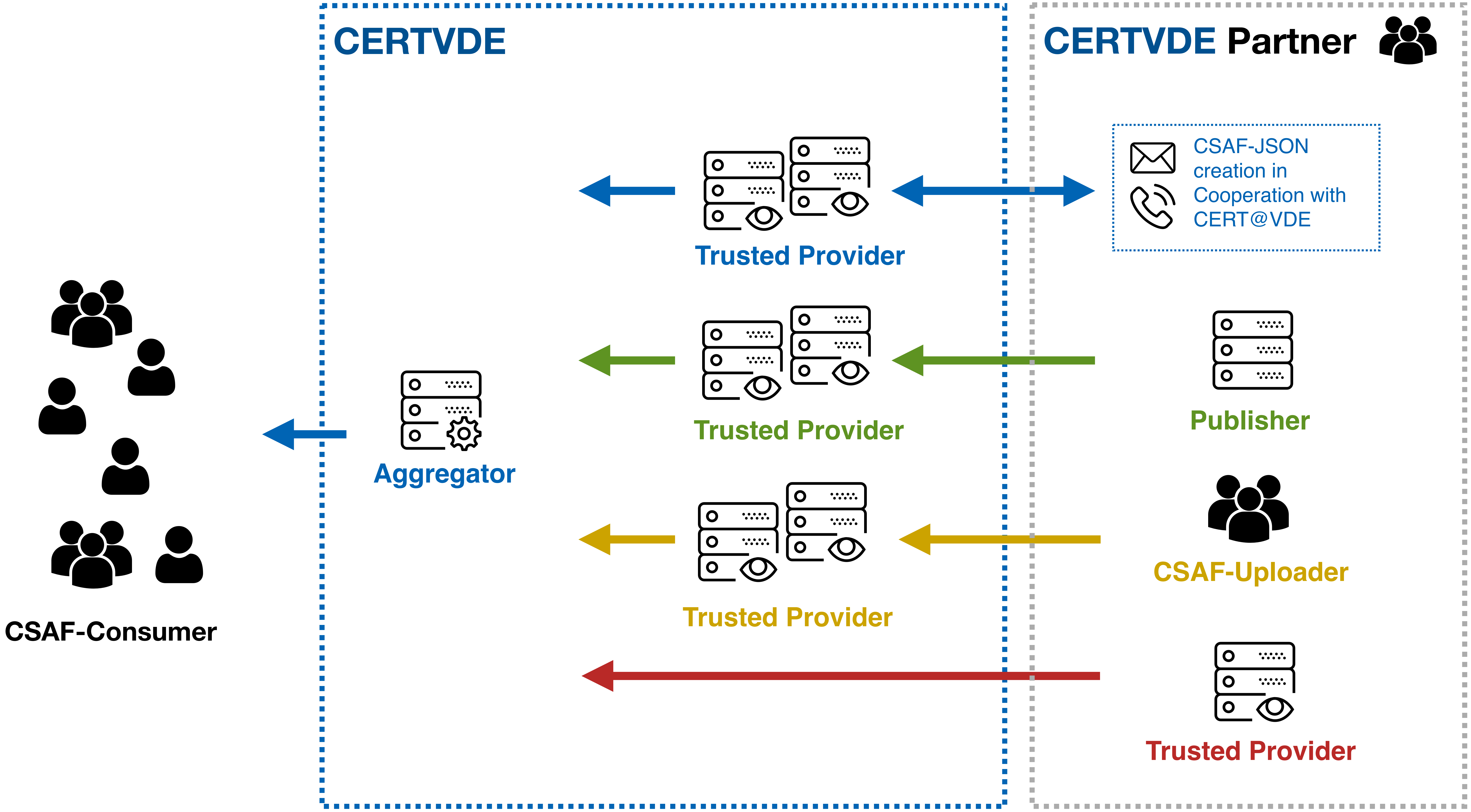
Requirements summed up

- 45+ Docker Container, one per partner
- 45+ PGP keys
- 45+ TLS client certificates
- 45+ TLS certificates for outside-facing services
  (or a wildcard cert, might require DNS validation)
- 45+ DNS entries (or some form of request routing)

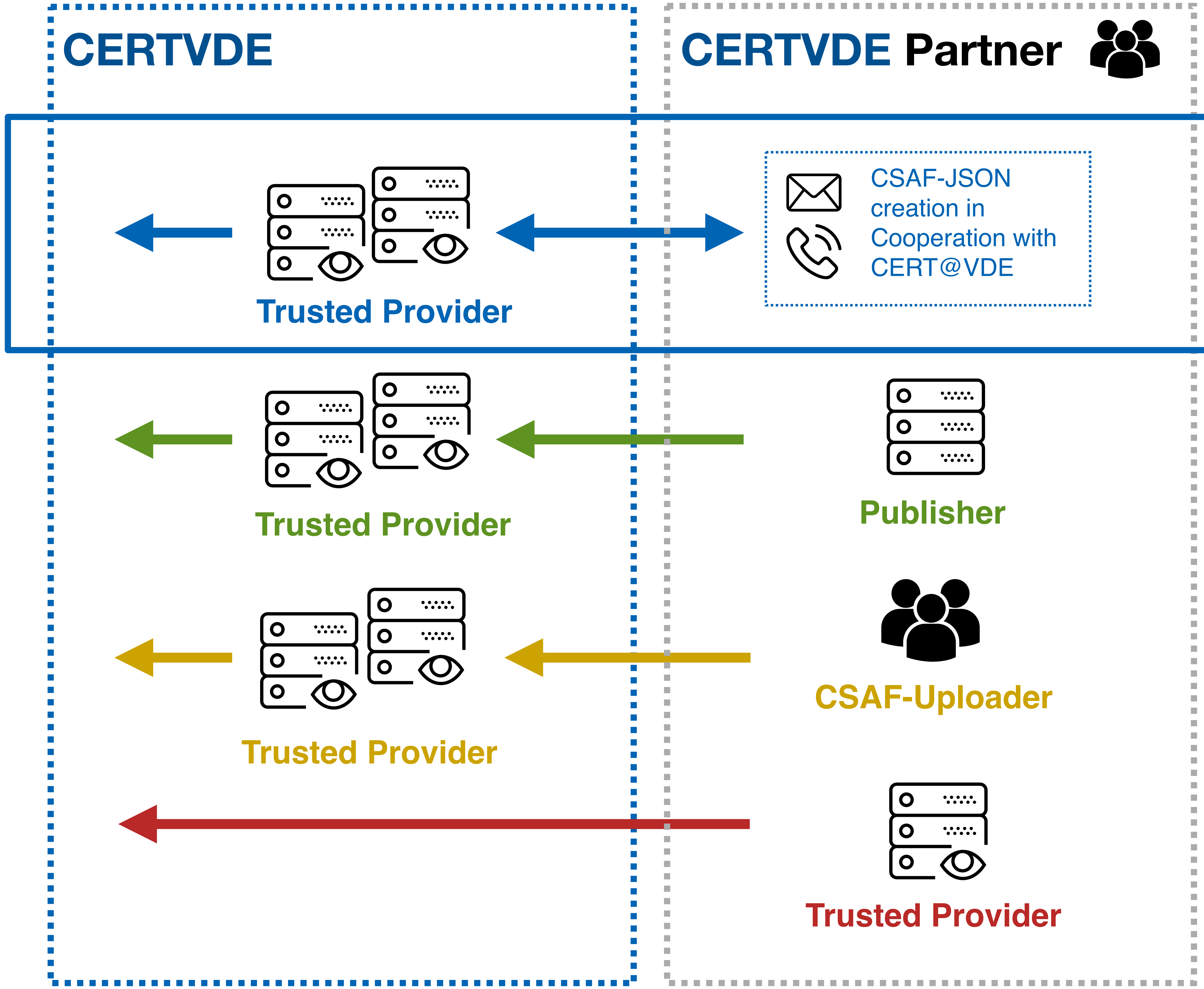
- A system for csaf_aggregator
  - every provider needs to be added as publisher (no sec.txt) or provider
- DNS, TLS Certificates
- A uploader UI

- **Different maturity level amongst our partners**

CSAF at CERT@VDE



CSAF at CERT@VDE



Details

CERTVDE handles CSAF publication

Partner provides csaf document on their infrastructure. We aggregate.

Partner uploads CSAF to Trusted Provider hosted by CERT@VDE

Partner hosts and handles everything. We aggregate

CSAF Consumer : wants provider-metadata.json !

CSAF-Consumer

csaf_downloader

csaf_checker

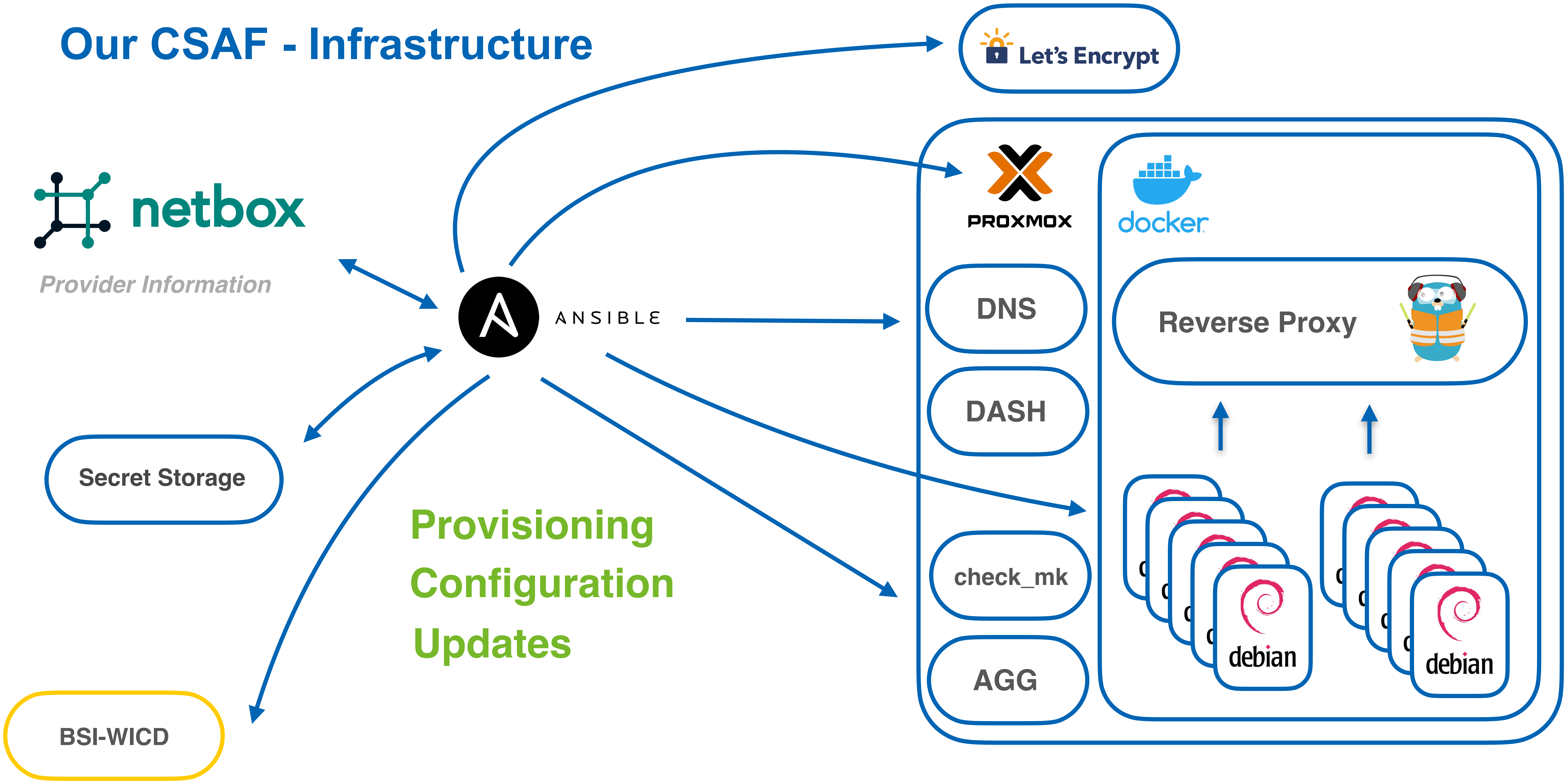
festo.com

DNS Query: **csaf.data.security**.festo.com → provider-metadata.json ?

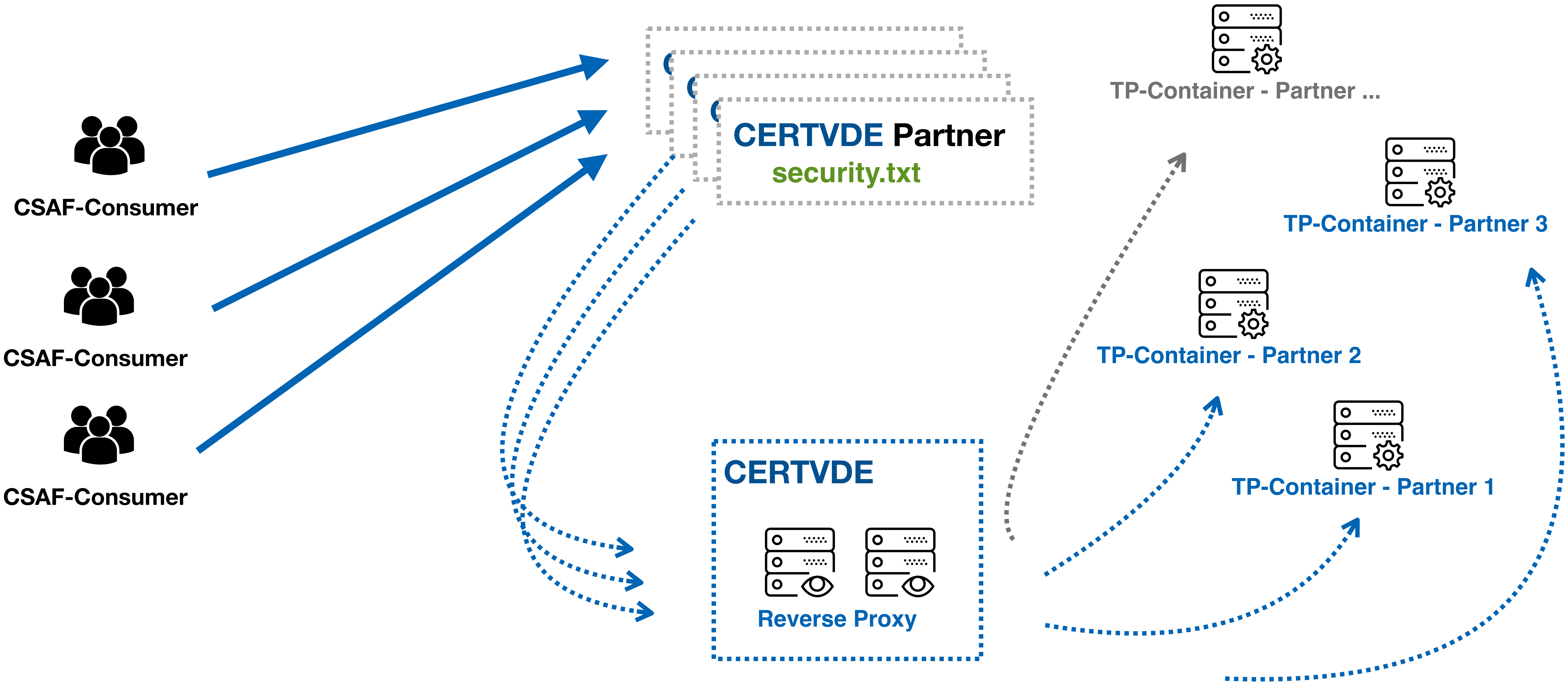
GET https://www.festo.com/**.well-known/security.txt**

→ https://festo.csaf-tp.certvde.com/**.well-known/csaf/provider-metadata.json**

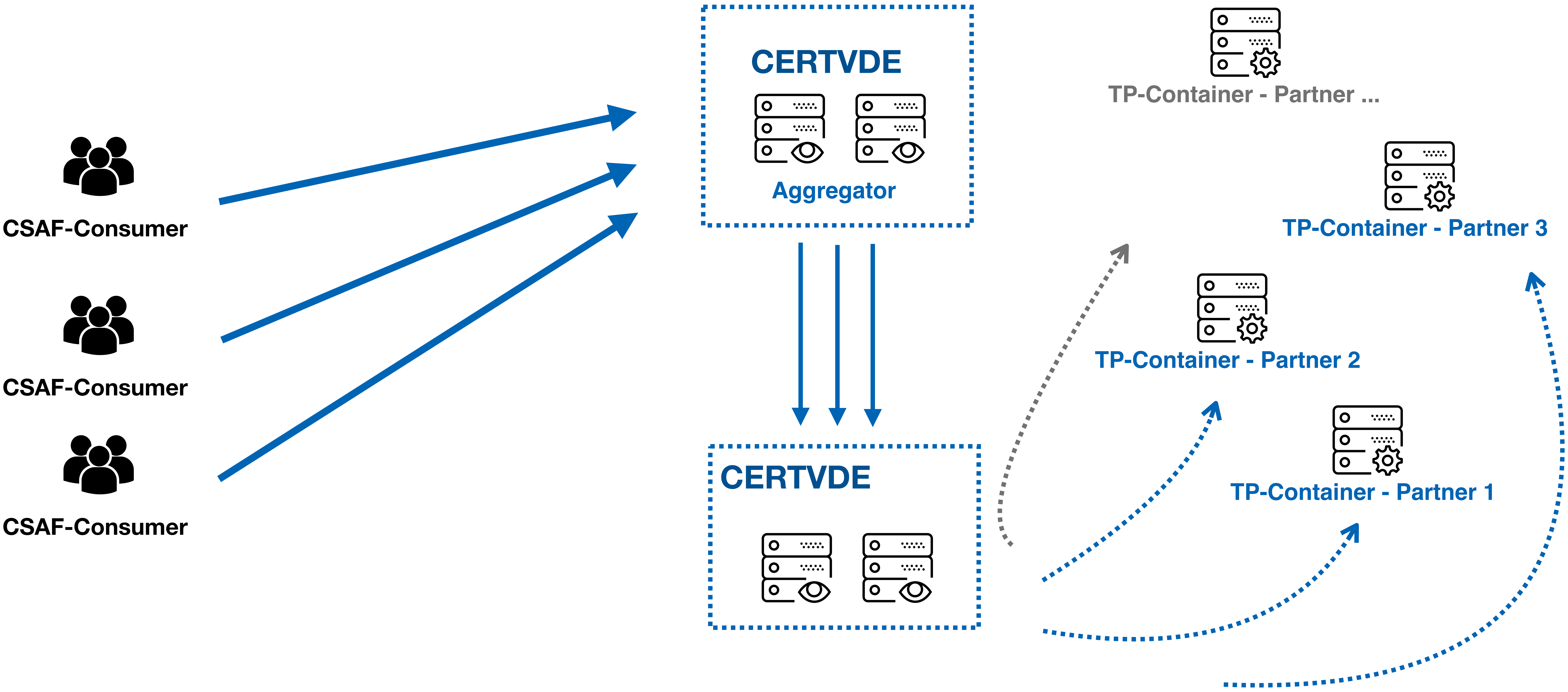
Our CSAF - Infrastructure



Direct access to the providers

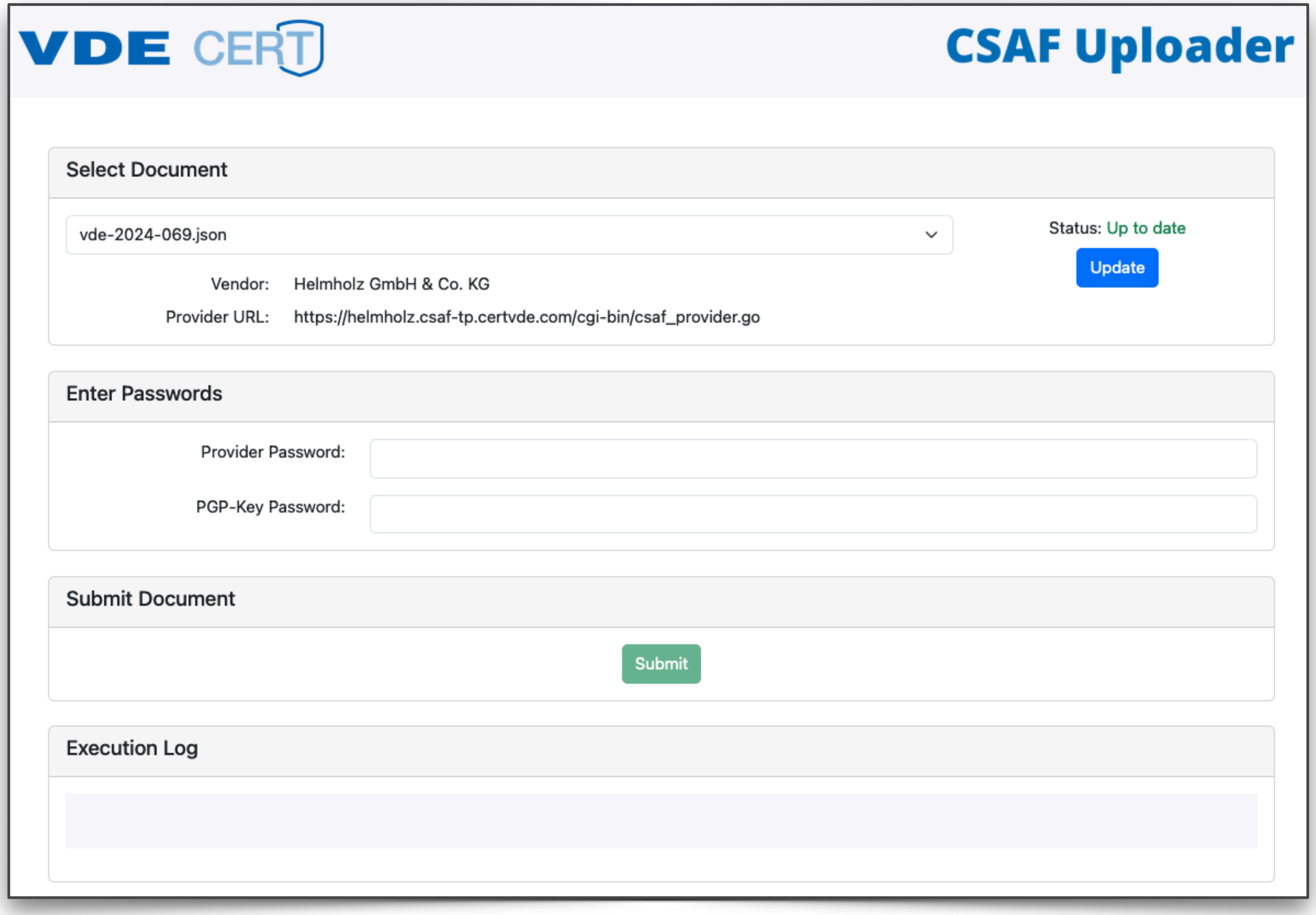


Access through aggregator



How to upload ?

- via ssh ?
- via csaf_provider webui ?
- DIY!



Select Document

vde-2024-069.json

Status: Up to date

Vendor: Helmholz GmbH & Co. KG

Provider URL: https://helmholz.csaf-tp.certvde.com/cgi-bin/csaf_provider.go

Enter Passwords

Provider Password:

PGP-Key Password:

Submit Document

Execution Log

- automatically selects the correct provider
- easy to use, no shell-fu needed
- extendable & integratable

Status

34 TrustedProvider up-n-running
1 Aggregator

~80 CSAF documents published

~250 "old" advisories to convert

~10.000 LoC for helper and scripts, n8n pipeline

Challenges I

gocsaf

- OK for a single-deployment, does not scale well
- documentation still needs improvement
- architecture could be simpler
- ~~no binaries~~

- not sure why it does what it does sometimes
- not sure why it doesn't do what it's supposed to
- debug information quality is miserable
  (if something doesn't work, you start at -1
  and almost always end up in the sourcecode)

Challenges II

Provisioning

- Source of static information about partners ?
- new, complex ansible playbooks needed.
- new provider-urls must be configured in DNS, FW and RevProxy Routing
- key/secret material storage and distribution, external TLS certificates
- Docker images, Docker registry, Deployment pipelines

Publication

- ease-of-use not given with csaf_uploader (for us)
- risk of uploading to the wrong TP

Operational aspects

- config changes, de-comissioning, monitoring
- tools and helper needed
- long request chain, hard to debug

Challenges III

one more thing ..

Questions ?