KubeCon CloudNativeCon Europe 2022

NOTARY V2: DEEP DIVE AND OPEN ISSUES

Steve Lasker

Principal PM Architect
Azure
Steve.Lasker@Microsoft.com
@SteveLasker
.blog SteveLasker.blog
github.com/SteveLasker
github.com/SteveLasker/presentations`

Justin Cormack

CTO
Docker
justin@docker.com
@justincormack
.blog cloudatomiclab.com
github.com/justincormack

Today's Agenda

- Notary v2 Goals
- Promotion Workflows
- What are supply chain artifacts?
- Who Do You Trust?
- Open Questions

Notary v2 Goals

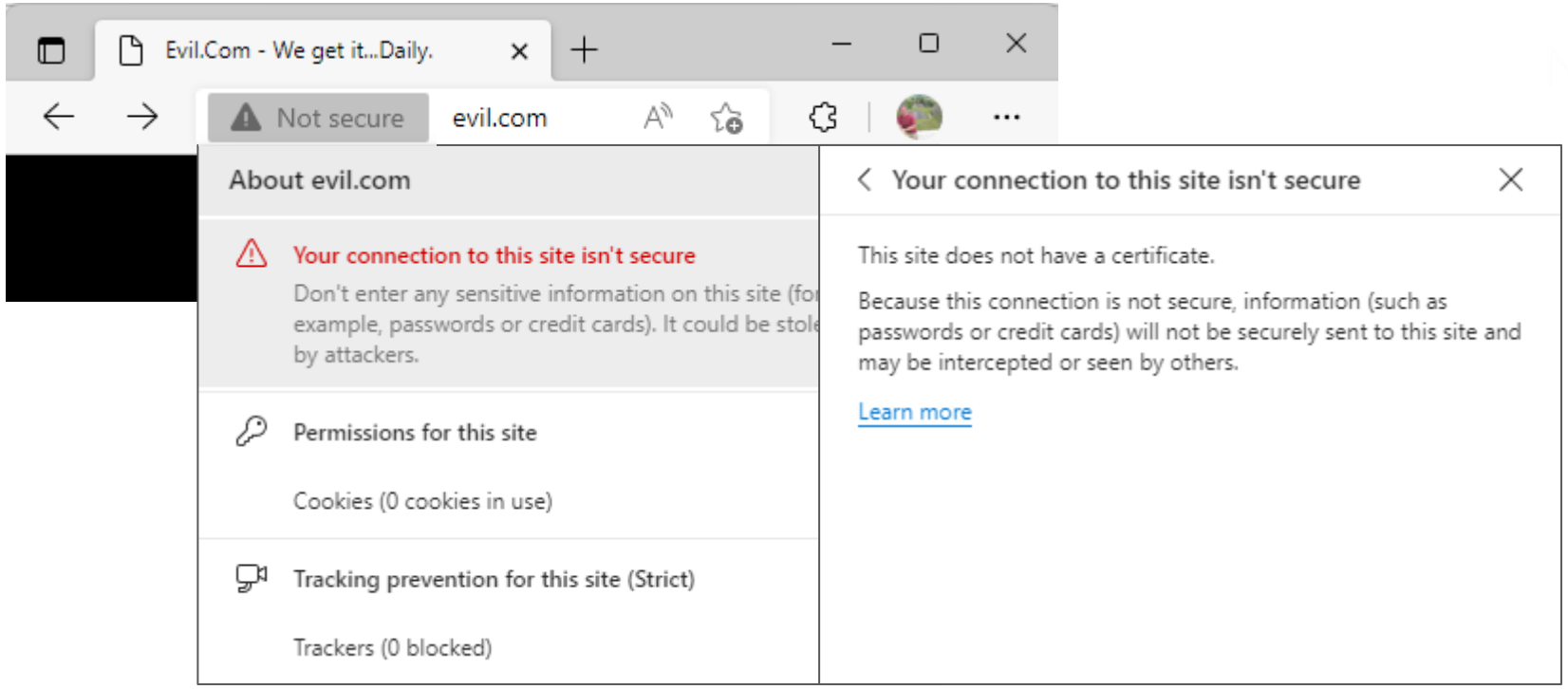
- Build on Existing Security Fundamentals
  - Building on existing specs, enhancing and adding as needed
  - Investing in security and supply chain libraries
    x.509, notation-go, oras-go, artifacts-spec
- Invest and Extend Existing Services
  - Registries are everywhere…
  - Investing in existing core infrastructure
- Best Practices for Secure Supply Chain Artifacts
  - Signatures, and all other supply chain artifacts flow with the images
  - Trust the integrity of the artifact made it from source to destination
  - Public → Private
  - Private → dev → staging → prod

https://github.com/notaryproject/notaryproject/blob/main/requirements.md#goals

Signing – what does it promise?

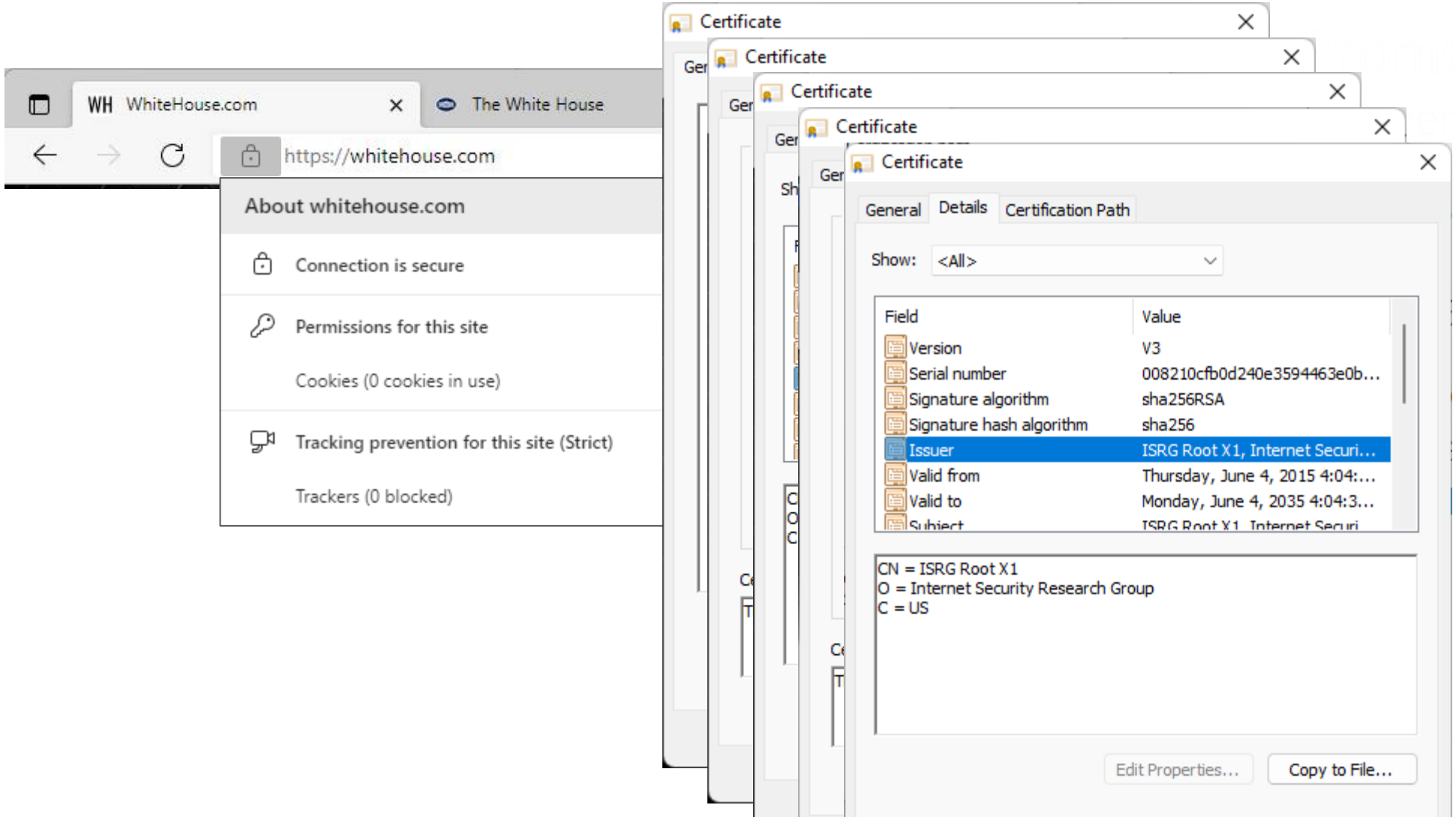
- “All deployed content must be signed”
  - By who?
  - Who **do you** trust?
  - Who **don’t you** trust?
  - Who do you/don’t you trust for a **given environment** or **application**?

Who Do You Trust

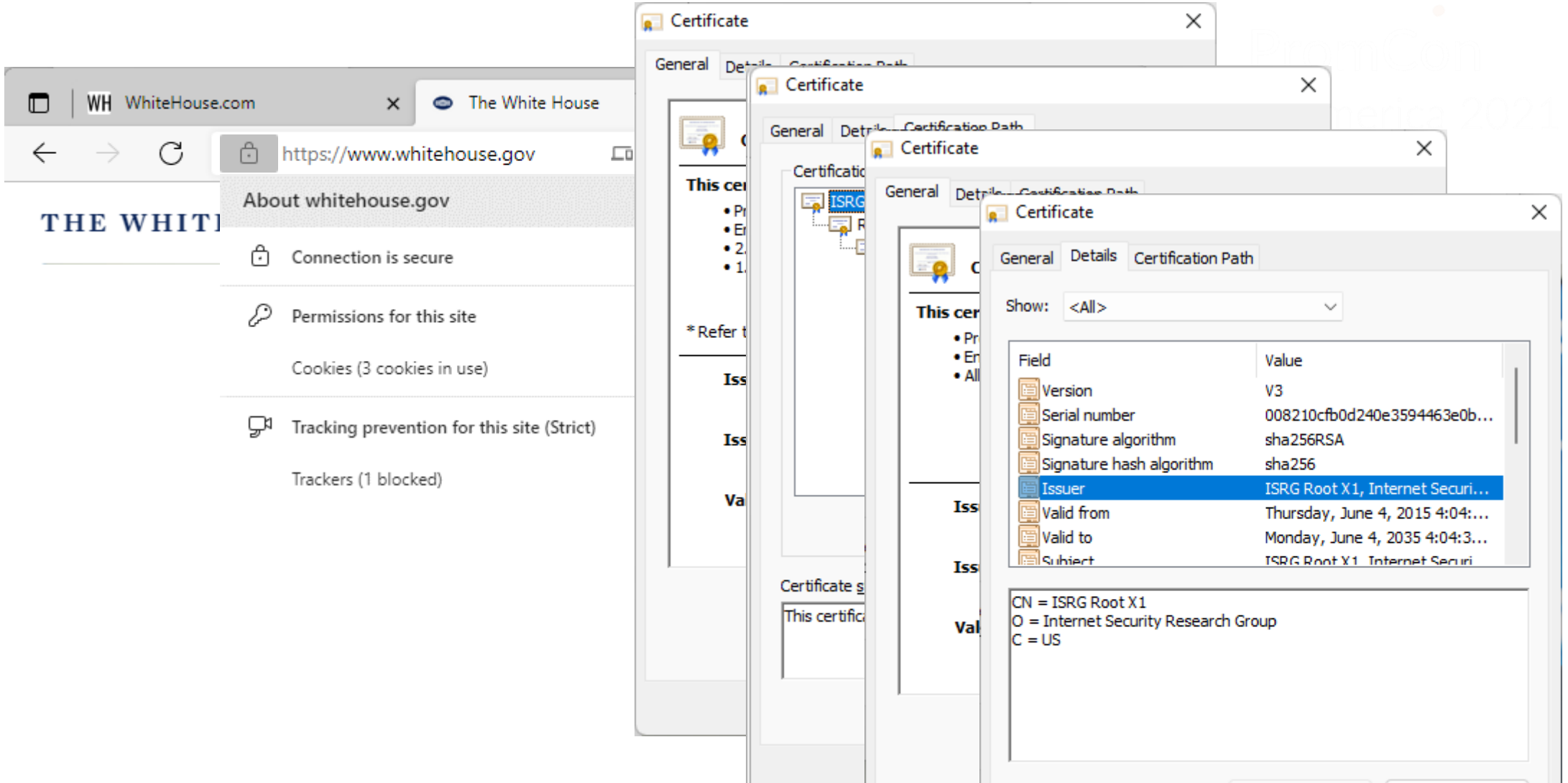


- No cert (not secure) = no trust

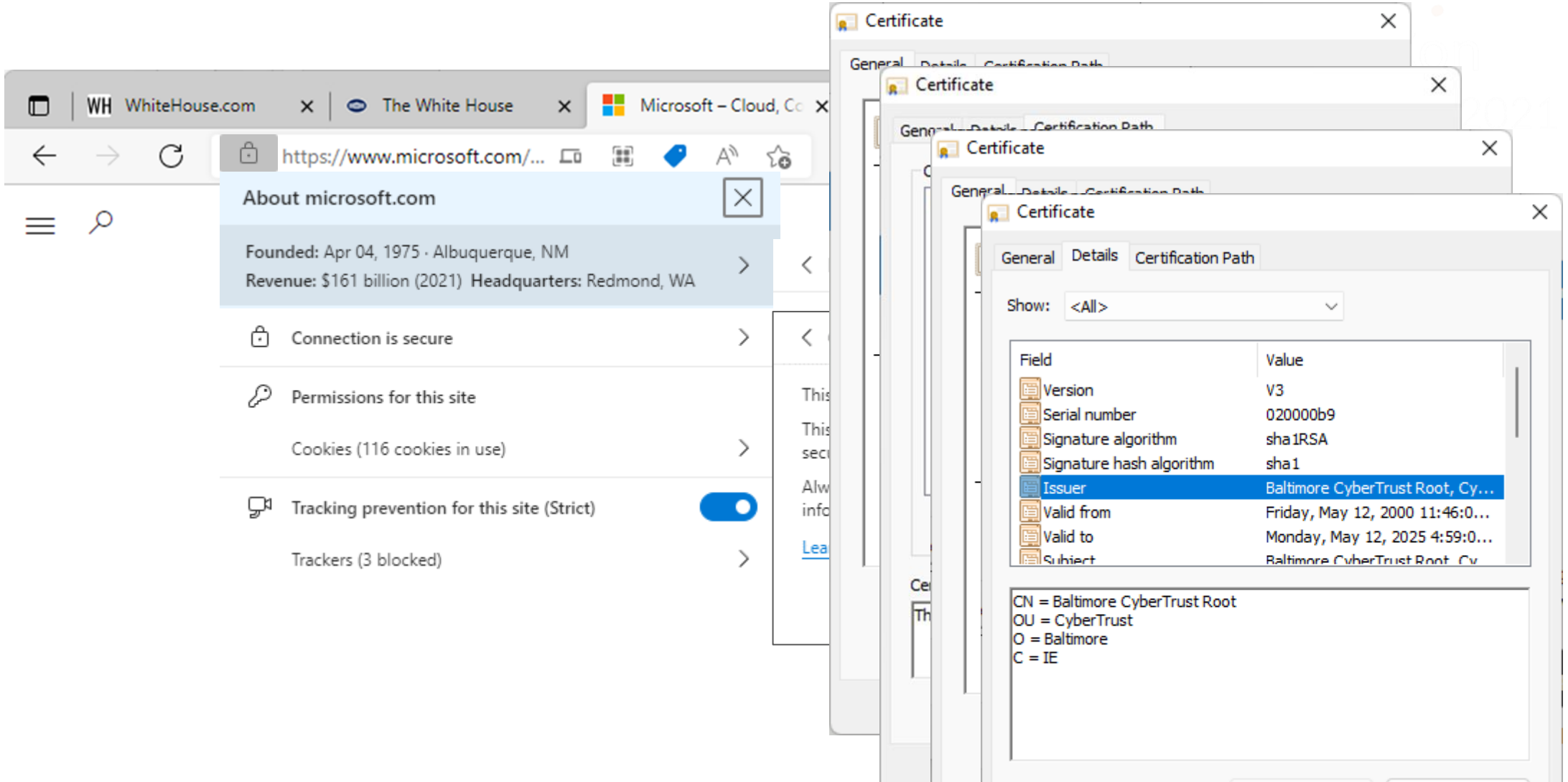
Who Do You Trust



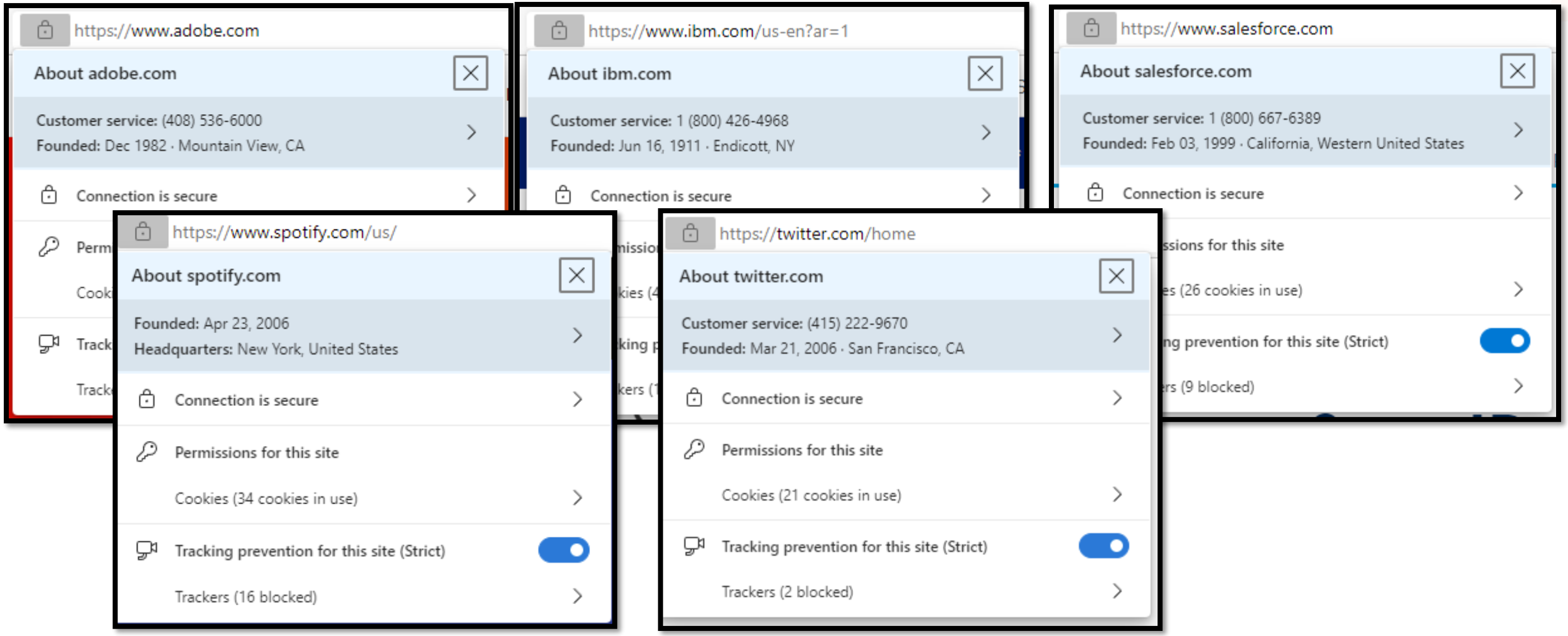
Who Do You Trust



Who Do You Trust



Who Do You Trust



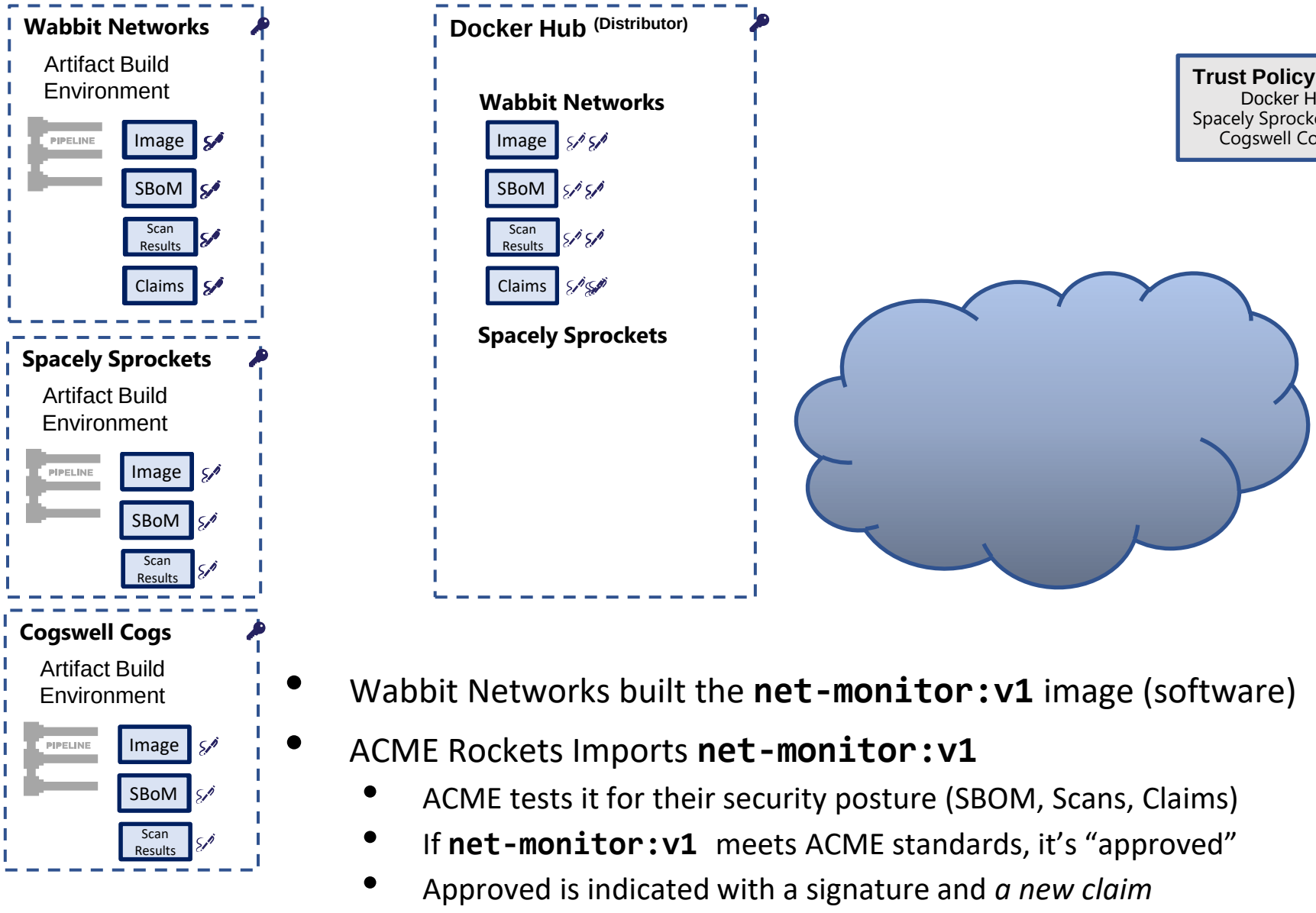
Different Levels of Trust

- Who/what do you trust?
- Do you defer everything you trust to ***central outside*** authorities?
- Browsers are different, *and similar*
  - There are many “interesting” and “bad” sites that have valid https: keys
  - Does that mean you want your children to view them?
- Do you have the same trust policies for:
  - Dev Box
  - Build Environment (SDKs and build tools)
  - Staging (secured from SDK and build tools)
  - Production (super secured)

Different roots and kinds of Trust

- I trust Microsoft to sign for Excel and Word
  - Trusted upstream vendors
- I trust my organization's signing infrastructure and certificate distribution
  - Spiffe
  - Traditional x509
- I have a trust root per application (TUF)
- Trust based on metadata (attestations) not just signatures

Artifact Promotion



- Wabbit Networks built the **`net-monitor:v1`** image (software)
- ACME Rockets Imports **`net-monitor:v1`**
  - ACME tests it for their security posture (SBOM, Scans, Claims)
  - If **`net-monitor:v1`** meets ACME standards, it's "approved"
  - Approved is indicated with a signature and *a new claim*

Notary v2 Policy Management

- Notary v2 enables policy-based management, per environment
- You configure which keys you trust, per environment
- Integrate with OPA/Gatekeeper and other policy managers

How Notary v2 Enables Secure Workflows

- Signatures are associated with a subject artifact
- Signatures are promoted with the artifact
- Multiple signatures may be associated with a given artifact
- Signatures are not embedded,
  enabling protection from trojan horse attacks
- Multiple Supply Chain Artifacts may be associated with a given artifact
  - SBOMs, Scan Results, Claims, Annotations
  - Anything in a registry may be signed with Notary v2

Demo

Local Signing and Verification

```
➜ docker push wabbitnetworks.azurecr.io/net-monitor:v1
The push refers to repository [wabbitnetworks.azurecr.io/net-monitor]
4fc242d58285: Pushed
v1: digest: sha256:81a768032a0dcf5fd0d571092d37f2ab31afcac481aa91bb8ea891b0cff8a6ec size: 527
```

```
➜ notation cert generate-test --default "wabbit-networks-test"
generating RSA Key with 2048 bits
generated certificates expiring on 2023-05-18T10:39:40Z
wrote key: /home/stevelas/.config/notation/key/wabbit-networks-test.key
wrote certificate: /home/stevelas/.config/notation/certificate/wabbit-networks-test.crt
wabbit-networks-test: added to the key list
wabbit-networks-test: marked as default
```

```
➜ notation sign --key "wabbit-networks-test" \
  wabbitnetworks.azurecr.io/net-monitor:v1
sha256:81a768032a0dcf5fd0d571092d37f2ab31afcac481aa91bb8ea891b0cff8a6ec
➜ oras discover -o tree wabbitnetworks.azurecr.io/net-monitor:v1
wabbitnetworks.azurecr.io/net-monitor:v1
└── application/vnd.cncf.notary.v2.signature
    └── sha256:119a5da2200b20a45df9c2cc82d36e313d7721db786b7f8f199945a1e8ba0b06
```

```
➜ notation verify wabbitnetworks.azurecr.io/net-monitor:v1
2022/05/18 12:41:08 trust certificate not specified
```



```
➜ notation cert add --name "wabbit-networks-test" \
  ~/.config/notation/certificate/wabbit-networks-test.crt
wabbit-networks-test
➜ notation verify wabbitnetworks.azurecr.io/net-monitor:v1
sha256:81a768032a0dcf5fd0d571092d37f2ab31afcac481aa91bb8ea891b0cff8a6ec
```

```
➜ notation cert list
NAME                   PATH
wabbit-networks-test   /home/stevelas/.config/notation/certificate/wabbit-networks-test.crt
```

Remote Signing

- Keep your private keys private
- Remote signing is enabled, but limited (eg: build machines)
- Leverages the existing x509 infrastructure most customers have

Notary Plug Ins

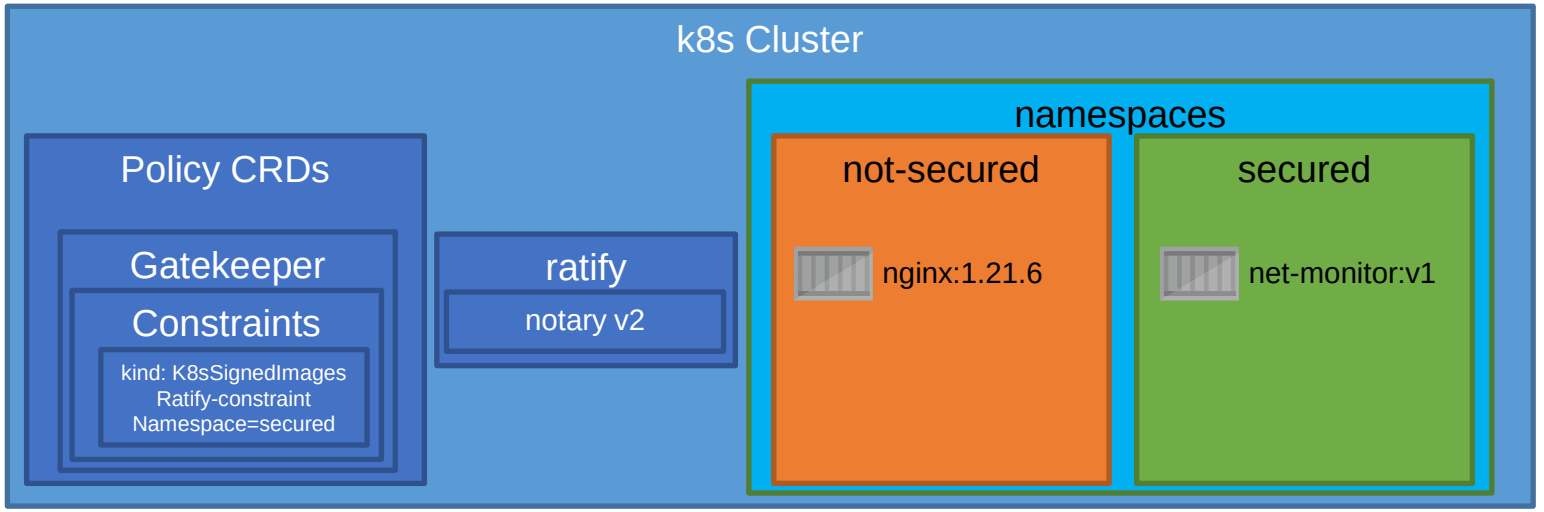
- Private keys locked away in secured key vaults
- Images and all other artifacts are remotely signed
- Plugins are external to the Notary Project
  - Plugin specification: github.com/notaryproject/notaryproject/blob/main/specs/plugin-extensibility.md
  - Plugins have autonomy for creation and updates

Demo

Remote Signing and Verification

Securing Kubernetes Namespaces

- Only allow the signers you trust
- Secured at scheduling with
  - Gatekeeper as the admission controller
  - Ratify for configured validators



Demo

Securing k8s Namespaces

```
➜ kubectl create ns secured
kubectl create ns not-secured
namespace/secured created
namespace/not-secured created
```

```
➜ kubectl run nginx \
  --image=nginx:1.21.6 \
  -n not-secured
pod/nginx created
```

```
➜ helm upgrade --install ratify ratify/ratify --atomic \
    --set registryCredsSecret=regcred \
    --set ratifyTestCert="$PUBLIC_KEY"
Release "ratify" has been upgraded. Happy Helming!
NAME: ratify
LAST DEPLOYED: Wed May 18 12:49:35 2022
NAMESPACE: default
STATUS: deployed
REVISION: 9
```

```
➜ cat <<EOF > ./constraint.yaml
apiVersion: constraints.gatekeeper.sh/v1beta1
kind: K8sSignedImages
metadata:
  name: ratify-constraint
spec:
  enforcementAction: deny
  match:
    kinds:
      - apiGroups: [""]
        kinds: ["Pod"]
    namespaces: ["secured"]
EOF
➜ kubectl apply -f ./constraint.yaml
k8ssignedimages.constraints.gatekeeper.sh/ratify-constraint
```

```
➜ kubectl run nginx \
  --image=nginx:1.21.6 \
  -n secured
Error from server (Forbidden): admission webhook "validation.gatekeeper.sh" denied the
request: [ratify-constraint] Image verification failed : {"errors": [["nginx:1.21.6",
"nginx:1.21.6_invalid"]], "responses": [], "status_code": 200, "system_error": ""}
```

```
➜ kubectl run net-monitor \
  --image=wabbitnetworks.azurecr.io/net-monitor:v1 \
  -n secured
pod/net-monitor created
```

Promoting Artifacts

- Several tools were used to create various artifacts
  - Container Build Tools, SBoM Creation, Image Scan Results, Signatures
- Once approved:
  promote from source to target, including the graph of artifacts

```
oras copy wabbit-networks.io/net-monitor:v1 \
      acme-rockets.io/net-monitor:v1 -r

registry.wabbit-networks.io/net-monitor:v1
├── application/vnd.cncf.notary.v2
│   └── sha256:8c0e82624475a4ad64ddca2d68c0f82e316ec758591cb916049fe59eaf27b5b4
├── application/vnd.org.snyk.results.v0
│   └── sha256:95b97532a5b2d0c36cfcefd403b02779fb5afca07479a66bd509a635d5681e7f
│       └── application/vnd.cncf.notary.v2
│           └── sha256:02f35a789d1ec2267727ac32c4f8ad643e88288528e648e4b71af42c2912699b
└── sbom/example
    └── sha256:82d89db16266d13ab7680badfea3dbd91fd8e311c3b4291d9c0d4f9cff86fa50
        └── application/vnd.cncf.notary.v2
            └── sha256:413e8b4de5f09c1b458d9d0ab8f1cc510d4276ffaa710073daff721e89b07aa2
```

Notary v2 Policy Management

- Notary v2 enables policy-based management, per environment
- You configure which keys you trust, per environment
- Integrate *with* Gatekeeper/Ratify and other policy managers

Notary v2 Signed Content

- Notary v2 finishing up RC1 this month (May)
- Microsoft shipping signed images and supply chain artifacts
  - For Azure Service validations
  - For US Executive Order Conformance (Claims & signed SBOMs)
  - Notation alpha 1 signed images coming online

```
notation verify --cert msft_supply_chain \
   mcr.microsoft.com/mcr/hello-world-oras-canary:demo
```

Other images:

- `mcr.microsoft.com/oss/kubernetes/kube-apiserver:v1.25.0-alpha.0`
- `mcr.microsoft.com/oss/kubernetes/ingress/nginx-ingress-controller:v1.2.0`

More info at: [aka.ms/mcr](aka.ms/mcr)

Notary v2 Status

- Specs finalizing
  - Notary Signature Specification
  - Signing and Verification
  - Trust Stores and Policy
- Releases
  - v0.7.1-alpha.1 released, with preview support in Azure
    - Supports remotely secured and signed x509 certs, verified and deployed to k8s
    - Azure Key Vault Provider
    - AWS in progress
    - Docker Official Image signatures coming soon
  - RC1 May
    - RC1 stable with feature/signature compatibility with 1.0

Open Questions

- Additional Identities?
  - SSH Keys for Build Systems & OSS projects/people
- Distributed Identity Support
  - Widening the types of identities, with validations
  - Policy, to decide the types of identities you wish to trust, for each environment

Thank You

- Notary v2: github.com/notaryproject/notaryproject
- OCI Artifacts: github.com/opencontainers/artifacts
- ORAS Artifact (Reference Types): github.com/oras-project/artifacts-spec
- ORAS CLI: github.com/oras-project/oras
- ORAS Library: github.com/oras-project/oras-go
- Ratify: github.com/deislabs/ratify
- CNCF Distribution Reference Types: github.com/oras-project/distribution
- OCI Reference Types Working Group: github.com/opencontainers/wg-reference-types

Steve Lasker

Principal PM Architect
Azure Container Registries
Steve.Lasker@Microsoft.com
@SteveLasker
.blog SteveLasker.blog
github.com/SteveLasker
github.com/SteveLasker/presentations

Justin Cormack

CTO
Docker
justin@docker.com
@justincormack
.blog cloudatomiclab.com
github.com/justincormack