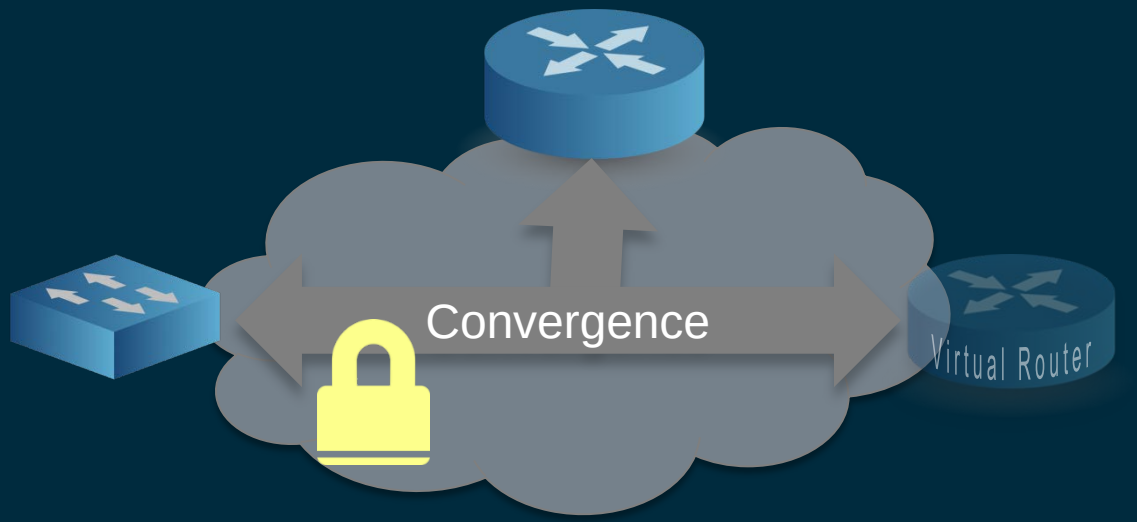
OpenC2 and Distributed Network Security Policy Convergence

Eric Voit
Principal Engineer
evoit@cisco.com

Jyoti Verma
Technical Leader
jyoverma@cisco.com

29-Sep-2016

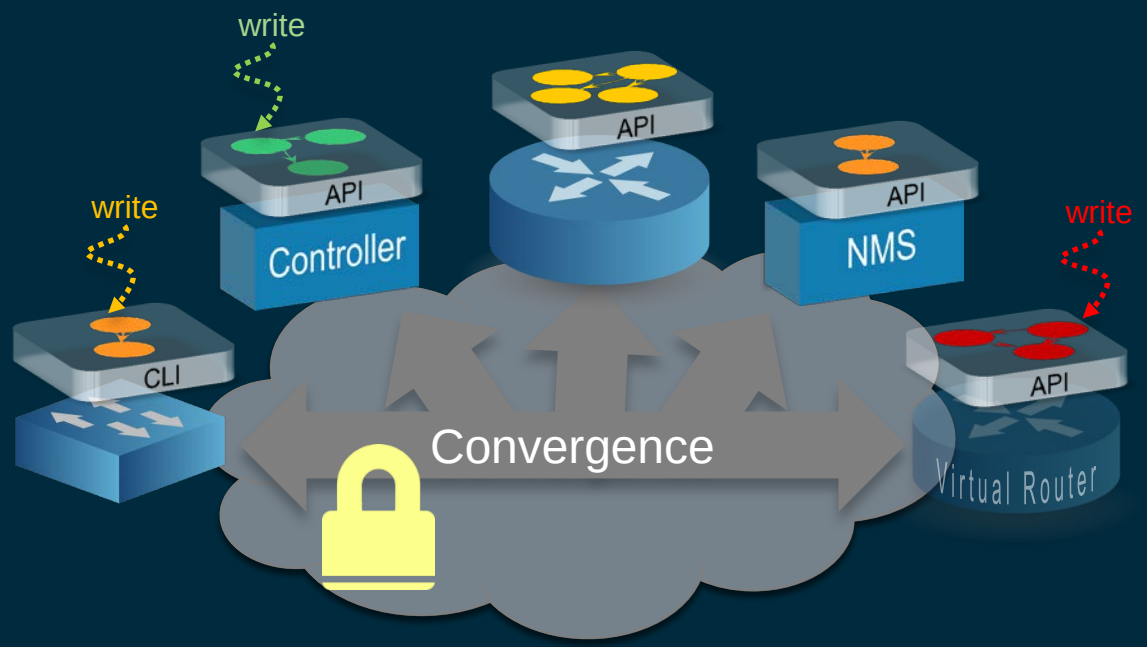
Routing Protocols and Network Convergence



- IP address forwarding table state
- Hundreds of trusted control plane devices
- Well known state machines
- Dozens of protocols

$10^{-6} \rightarrow 10^{2}$ second convergence times

Network Policy Convergence



- Distributed ownership and reconciliation
- Inter-dependent abstractions
- Custom & decoupled Mgmt systems
- Consistency enforcement a function of convergence speed

What it is

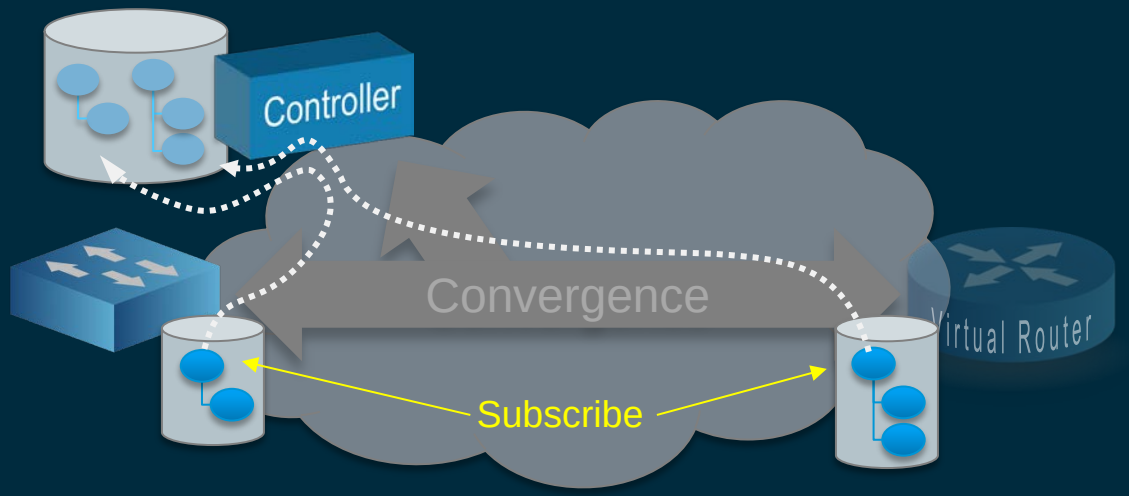
$10^{1} \rightarrow 10^{5}$

$10^{-2} \rightarrow 10^{1}$

What it must become

second convergence times ← 3 orders of magnitude improvement needed

Network Subscriptions



CRUDS (Create, Read, Update, Delete, Subscribe)

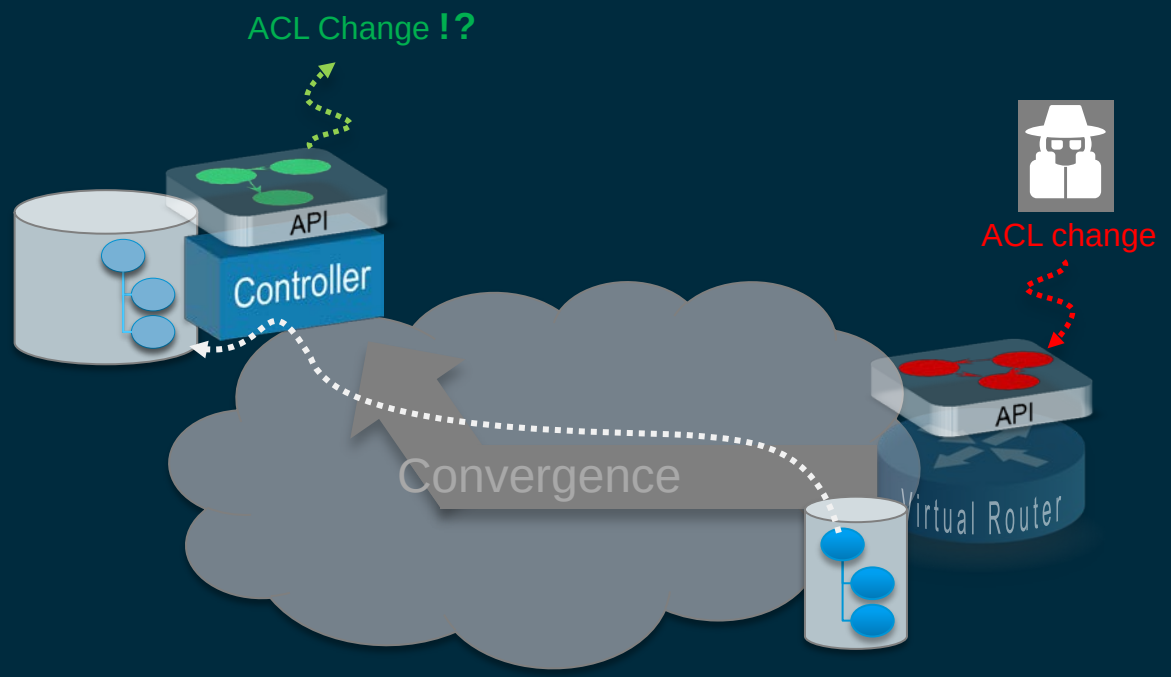
Solves known cost/scale limits of polling

- Propagation latency
- CPU, Bandwidth

Up-to-date objects delivered faster

- ✓ New use cases

Subscription Security Use Case: Integrity Verification

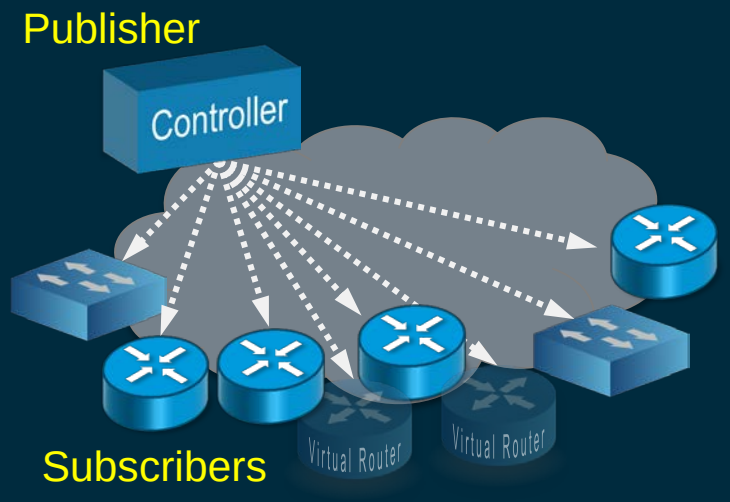


Immediate push of specific changes

- Unauthorized Hardware insertion
- Software Integrity Verification checksum
- Config change
- Current environmental fingerprint

5+ orders of magnitude improvement in recognition speed

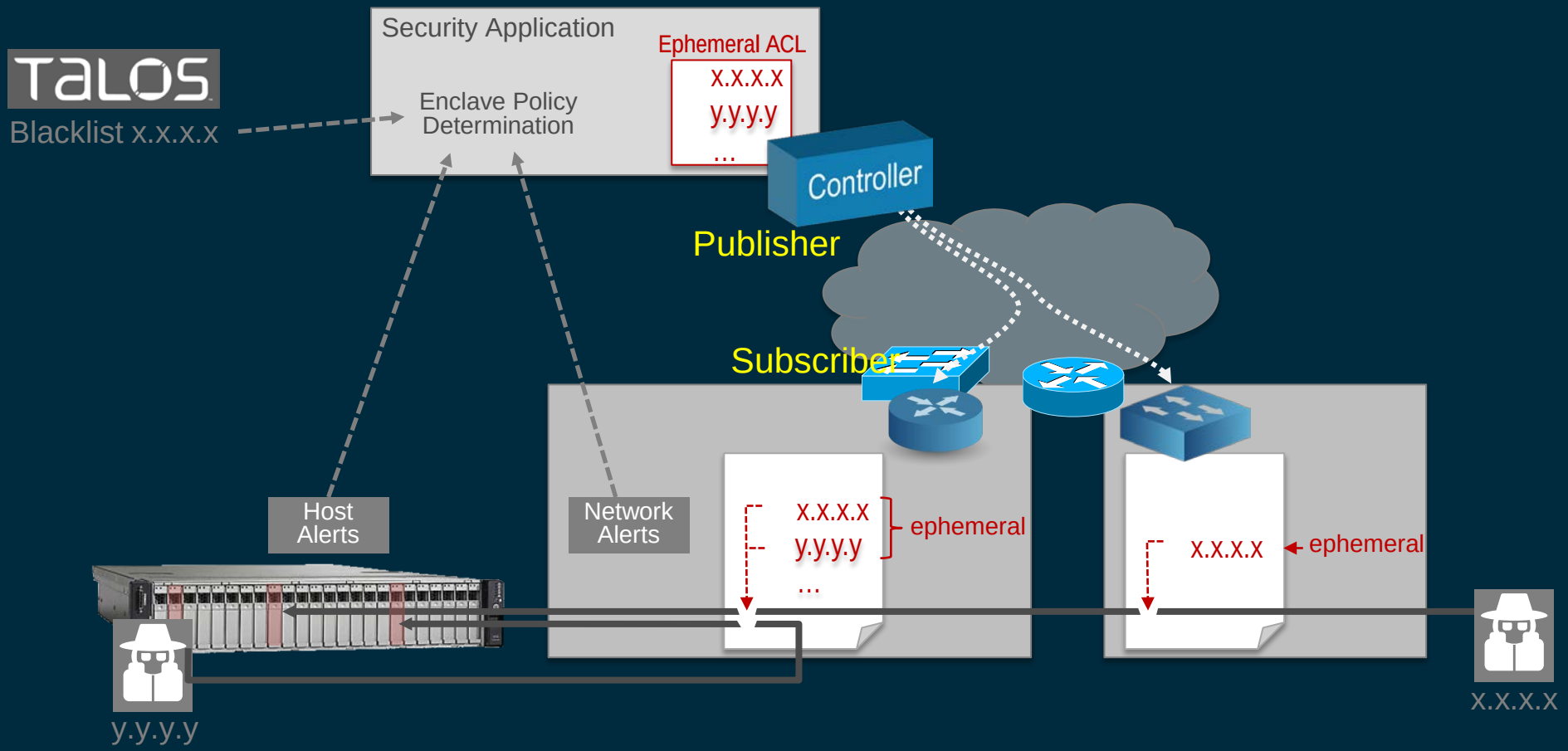
Network Element as Subscriber



Device doesn't have authoritative ownership.
Instead the primary copy is explicitly elsewhere.

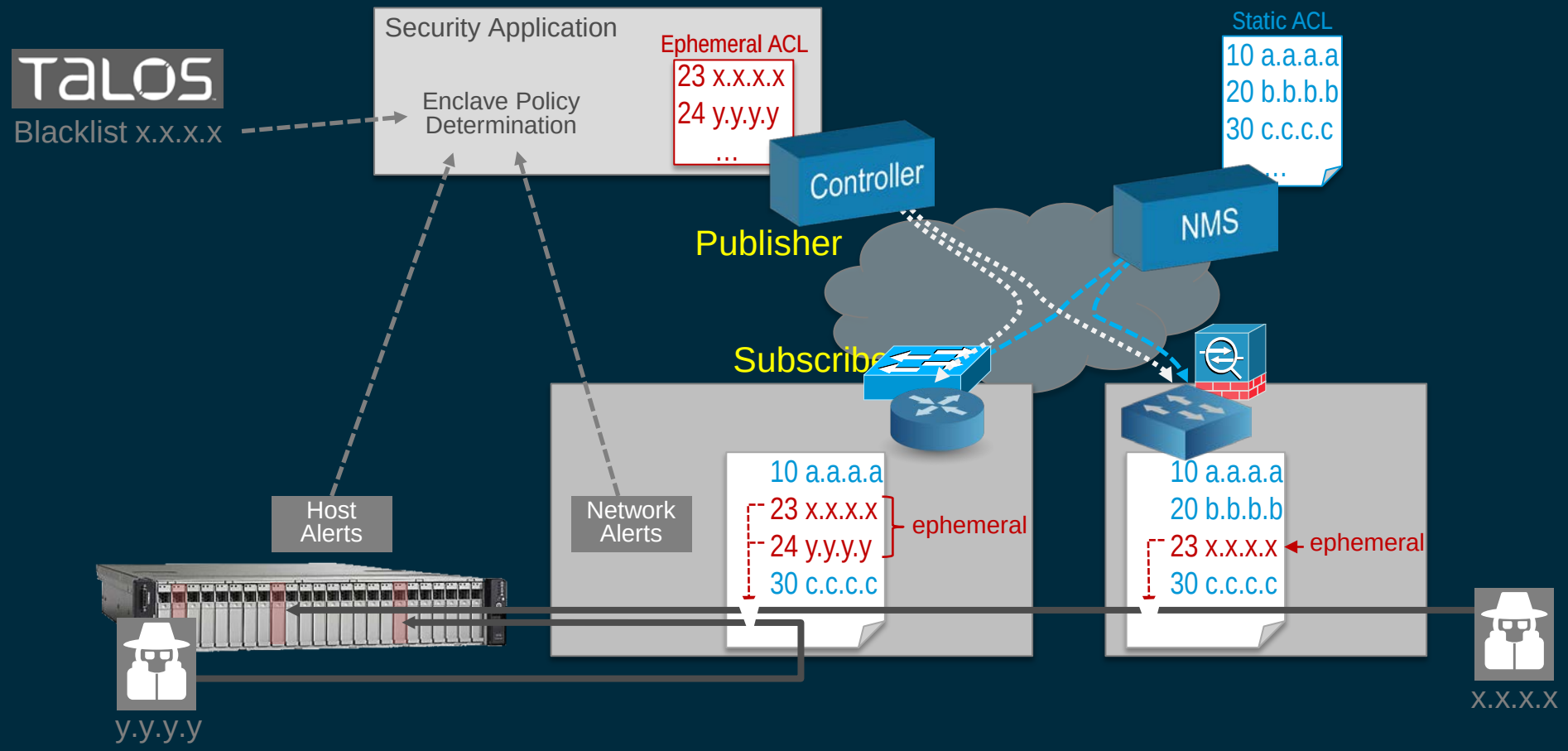
✓ Scalability	Frees up the authoritative source from continuously tracking config everywhere
✓ Troubleshooting	Single, central device config
✓ Self healing, Auto-config	Reduces logical copies of actively managed info
✓ Flexible	Can be for subset of config

Use Case: Perimeter & Internal Blocking



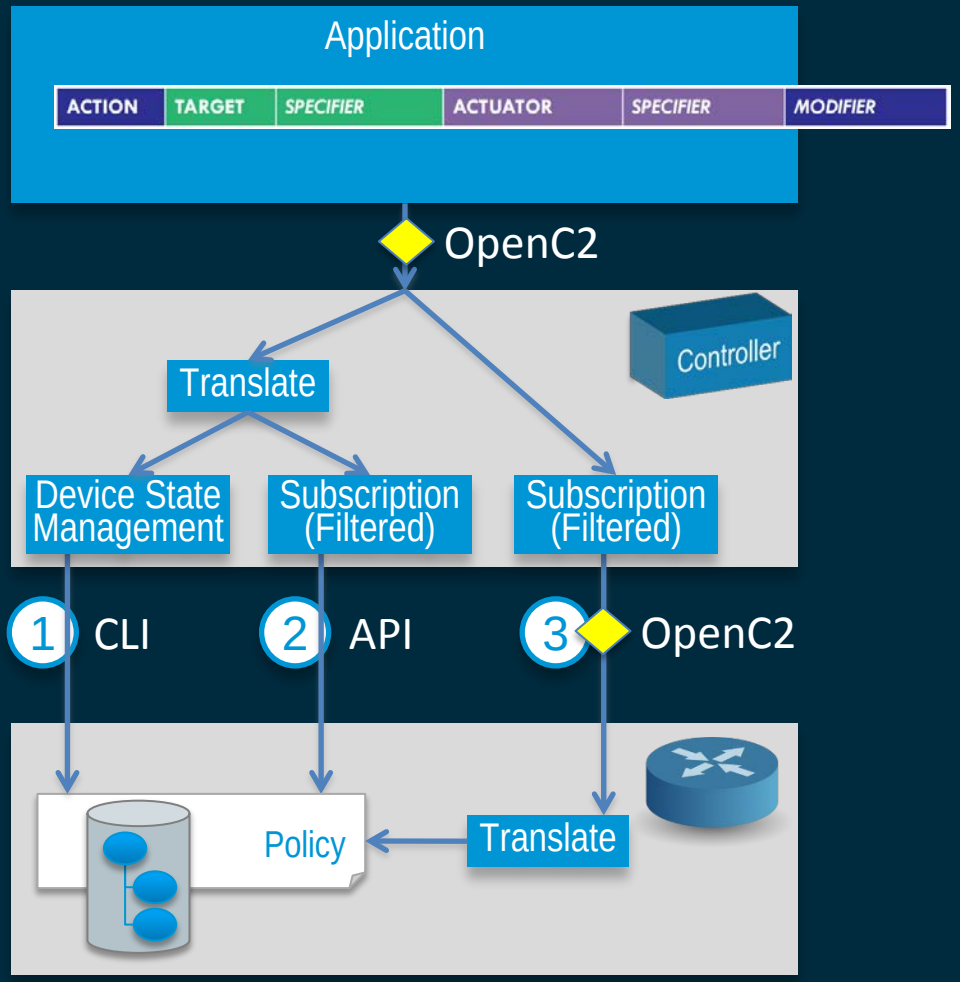
Changing Enclave Policy immediately reflected with Ephemeral config over a set of devices

Use Case: Perimeter & Internal Blocking



Separation from existing NMS Policy

OpenC2 Alternatives for Network Actuation



Alternatives for Network Element

1. Existing Network Element CLI/API
2. Subscribed Network OS API
3. Subscribed OpenC2 to Network Element

OpenC2 Alternative Selection Criteria

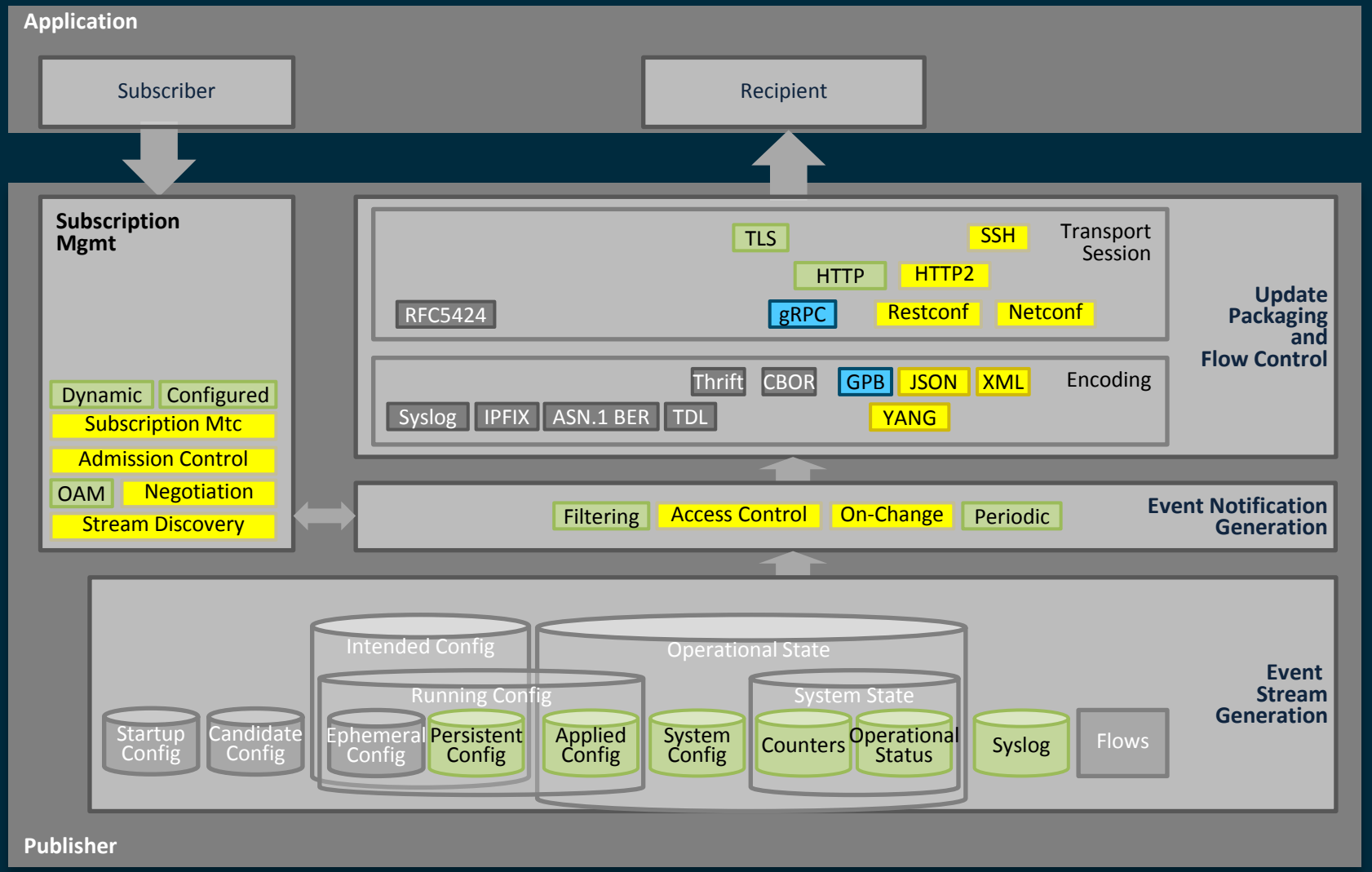
	Convergence Speed	Scale	Controller State?	Auto-config / Self-healing	End-to-end Encryption	Embedded base	Local NE Application
1 NOS CLI/API	Slow	Low	Yes	No	No	Yes	No
2 Subscribed NOS API	Fast	High	No	Yes	No	No	No
3 Subscribed OpenC2	Fast	High	No	Yes	Viable	No	Yes

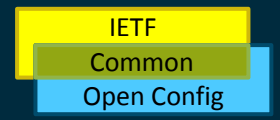
Takeaways

- Changes to Network Policy convergence will be relevant to end-to-end OpenC2 deployments, even if these changes are under-the-covers
- Edge/leaf based subscription to Policy (however it is expressed) improves scale and simplifies management

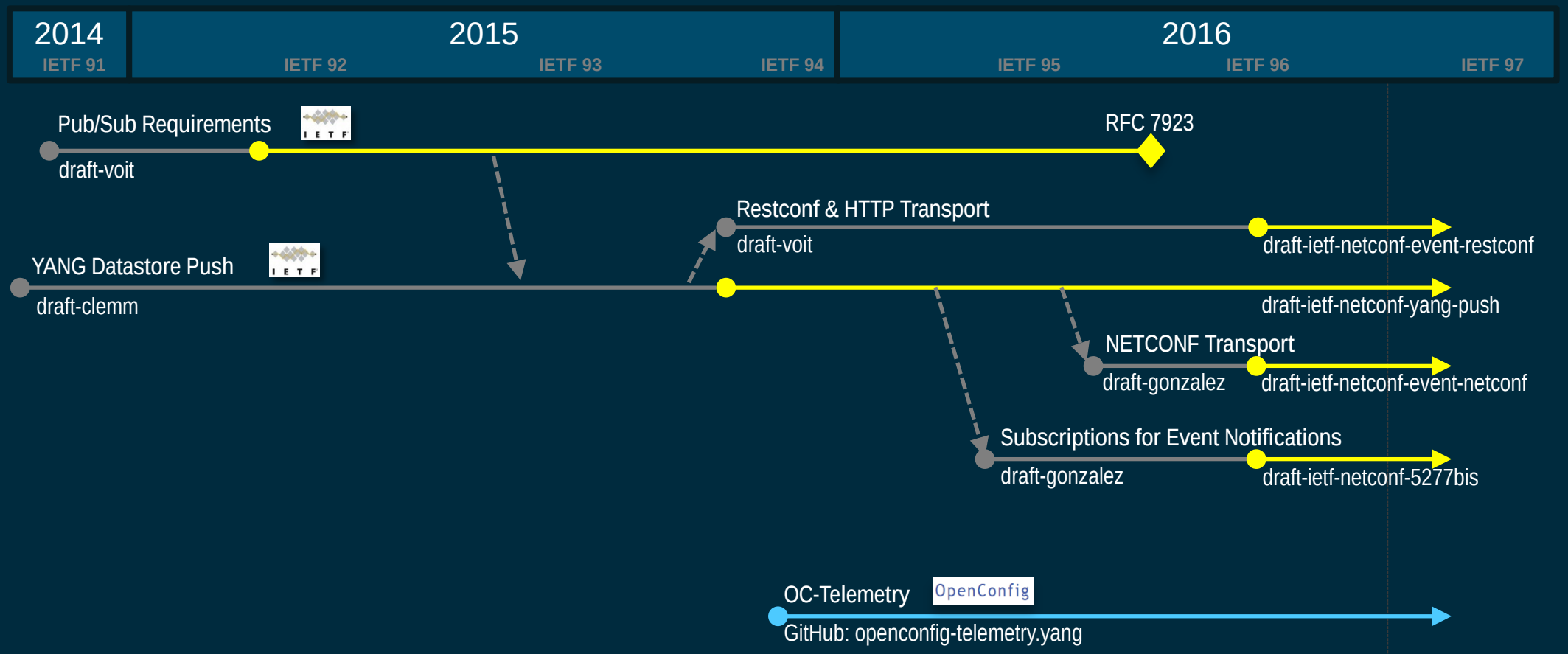

CISCO

Layered Subscription Framework



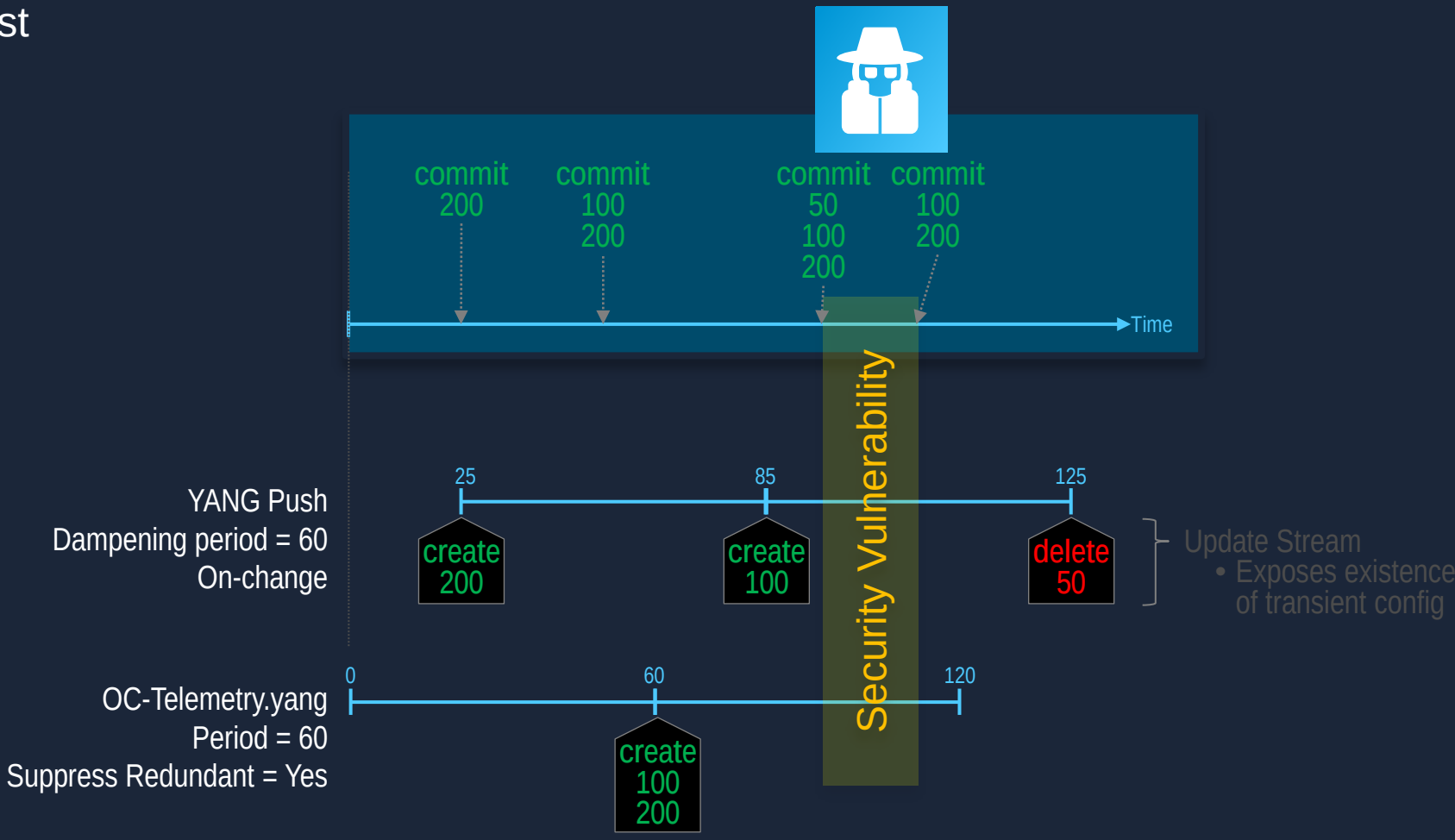


Network Subscription Specification Progression



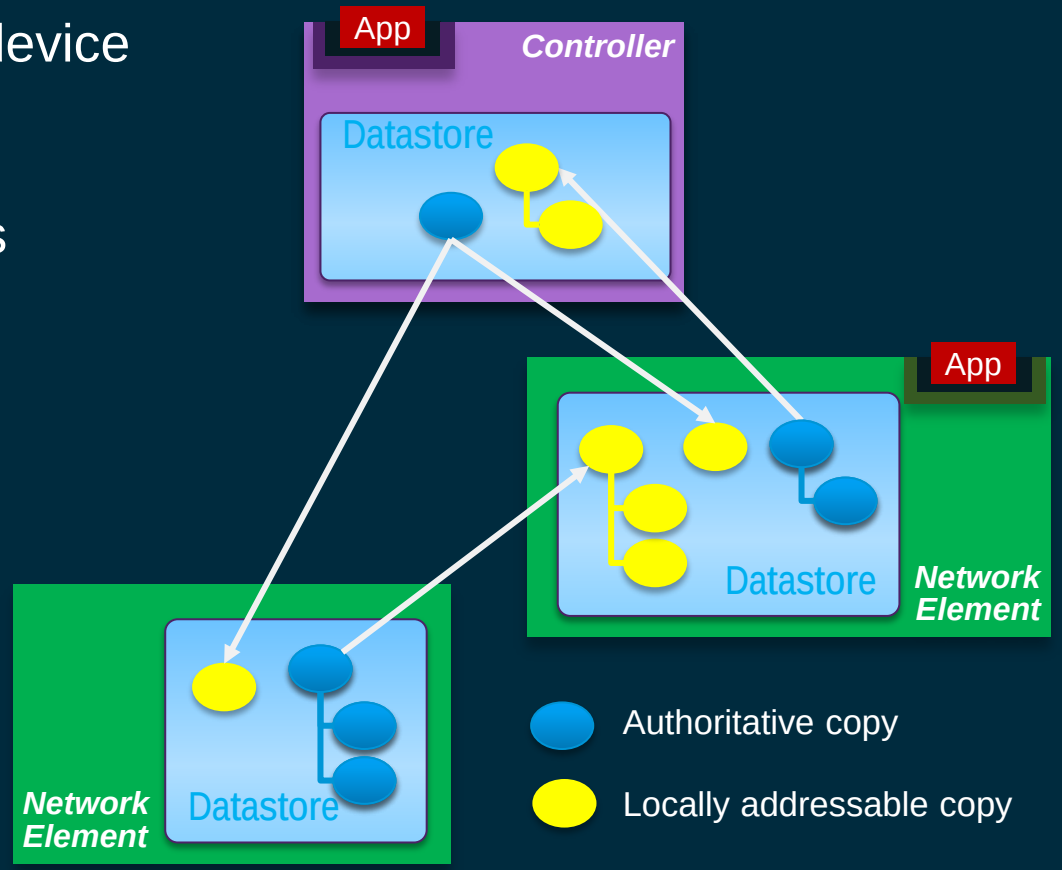
Dampening Period & Suppressed Periodic Behavior

Subscription to Access Control List



Mount
One Authoritative Copy

- Excerpt of Network-wide Datastore assembled on device
- Coding occurs without developer knowing protocols

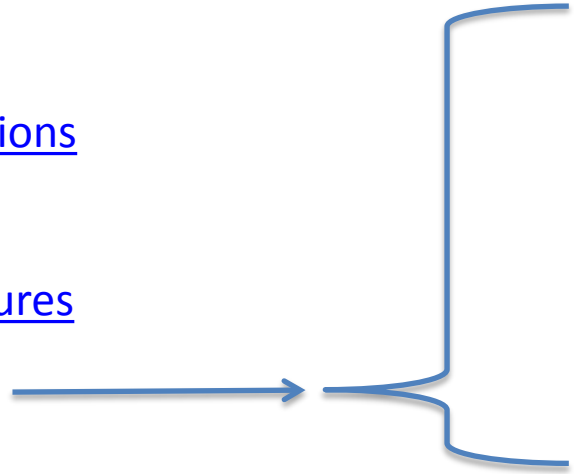


Questions as we try to figure what to prototype

https://github.com/OpenC2-org/docs-pub/blob/master/use-cases/mitigate-evil-domain.md

OpenC2 Use Cases

- Block on Indicators
- Email Phishing
- HBSS Signature
- Host Remediation Actions
- Host Remediation
- Update Sensor Signatures
- Mitigate Evil Domain



Mitigate Evil Domain actions DENY with Step 18 method = "sinkhole" or Step 20 method "ACL", plus applicable RESPONSE in Step 19/23.

Work through how the policy is withdrawn. I have been assuming that the applied policy would time-out of the network. But I would like to revisit the pros & cons.