A Case for Remote Attestation in Programmable Dataplanes

(from HotNets'22)

Nik Sultana, Illinois Tech

Deborah Shands, SRI Intl.

Vinod Yegneswaran, SRI Intl.

HotSoS'23

“Athens Affair”

- Provided inspiration for this work.
- Programmable network equipment was silently patched to activate Lawful Intercept features.
- Enabled eavesdropping on the private communications of the prime minister of Greece and least 100 other high-ranking officials.
- The attack came to light by accident, when an upgrade resulted in a noticeable malfunction.

Programmability

Opportunities

+ Features/Extensions

+ In-field updates

+ Low time-to-market

Programmability **in Networking**

Opportunities

+ Features/Extensions

+ In-field updates

+ Low time-to-market

Risks

– **Control plane:** larger scope of misconfiguration of dataplane. (software version, state, and hardware features).

– **Data plane:** misbehaving firewall and ACL logic leading to data leaks, DoS, liability.

Programmability **in Networking**

- Internal: installing modified dataplane program.
- Internal: misconfiguring (dropping/redirecting wrong packets). Installing contradictory rules across switches.
- External: cache poisoning, spoofing, evasion, “bypass attacks”.
- Internal+External: exploiting device-specific “undefined” behaviours.

Risks

Examples

– **Control plane:** larger scope of misconfiguration of dataplane. (software version, state, and hardware features).

– **Data plane:** misbehaving firewall and ACL logic leading to data leaks, DoS, liability.

Programmability in Networking

Opportunities

+ Features/Extensions

+ In-field updates

+ Time-to-market

Risks

– **Control plane:** larger scope of misconfiguration of dataplane. (software version, state, and hardware features).

– **Data plane:** misbehaving firewall and ACL logic leading to data leaks, DoS, liability.

Scope of programmability is widening:
Successive generations of hardware have more programmable resources.

] Can amplify **+** And exacerbate **-**

Programmability in Networking

Opportunities

+ Features/Extensions

+ In-field updates

+ Time-to-market

Risks

– **Control plane:** larger scope of misconfiguration of dataplane. (software version, state, and hardware features).

– **Data plane:** misbehaving firewall and ACL logic leading to data leaks, DoS, liability.

This talk

Mitigations:

- **Semantic analysis & Verification**
- **Remote Attestation**
- **…**

Remote Attestation

- Attestation ≠ Correctness
  Both are important.

- Answers question:
  “What hardware/software processed my data?”

- Helps us determine integrity.

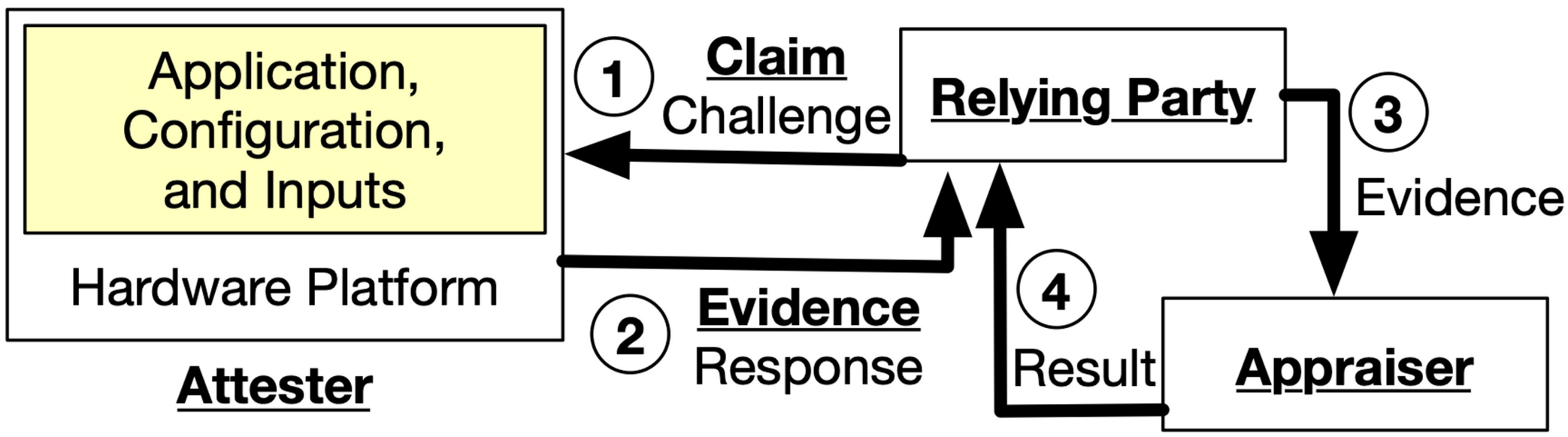
- Use cases:

  - Remote computing resources (e.g., commercial cloud).

  - IoT (e.g., trustworthiness of 50+ devices at home)

Remote Attestation

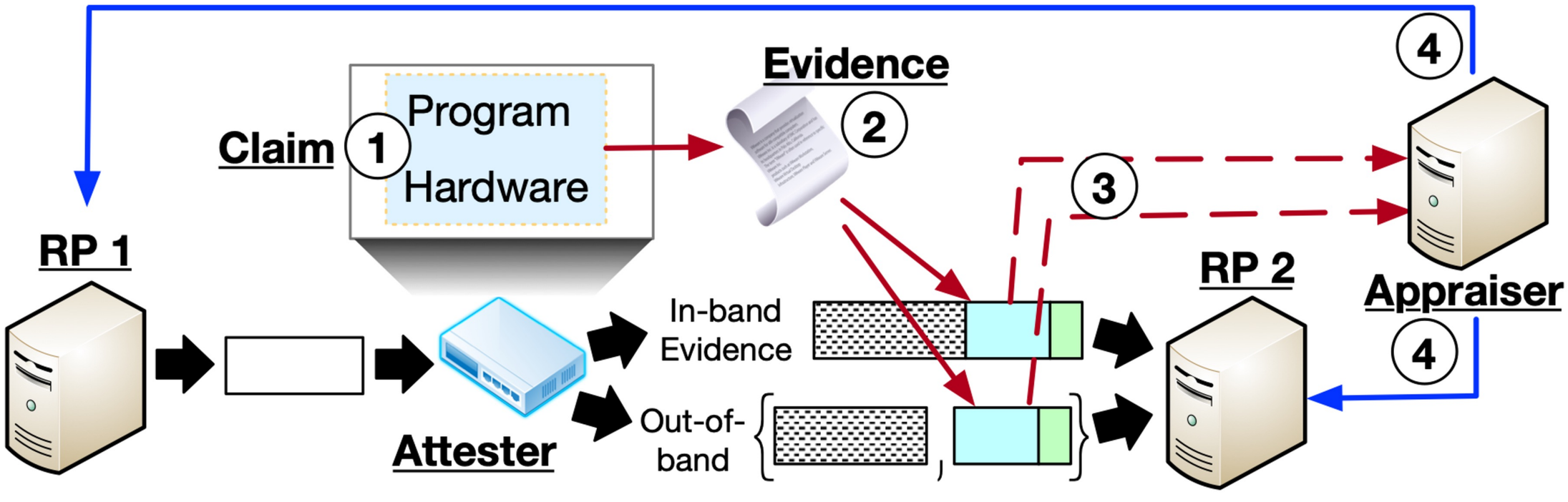
- Protocols for building + using evidence.
- Bootstrapping trust among mutually distrustful actors.



- **Our paper** = How to leverage Remote Attestation in Programmable Networking:
  - Specification
  - Mechanism
- *Scratches the surface*: much more work needed across different technical domains.

Remote Attestation
in Programmable Networking

- Adapting ideas + actor roles from RA.
- Defining requirements & assurance & performance trade-off wrt RA in PN.
- Architecture sketch:

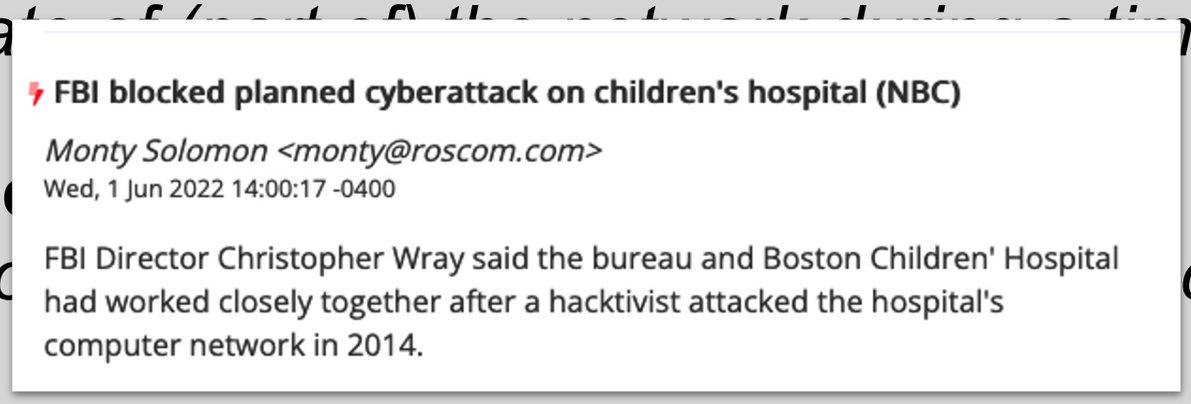


Use Cases

- Paper has several use-cases:
  - **Configuration assurance**:
    What is processing your traffic, and under what configuration?
  - **Authentication**: Path Evidence as Security Factor
    Authorization: Path Evidence as Tag
    What can be inferred from your (network) location, and what can that enable?
  - **Auditing**: Evidence as Documentation
    What was the state of (part of) the network during a time interval?
  - **Cross-Referenced** Attestation
    How does end-host evidence affect path-based evidence generators?
- Here focusing on **Configuration assurance**; please see the paper for the others.

Use Cases

- Paper has several use-cases:
  - **Configuration assurance**:
    What is processing your traffic, and under what configuration?
  - **Authentication**: P
    Authorization: Pa
    What can be infe
    that enable?
  - **Auditing**: Evidence a
    What was the sta *ne interval?*
  - **Cross-Referenc**
    How does end-ho *ce*
    generators?



- Here focusing on **Configuration assurance**; please see the paper for the others.

Remote Attestation
in Programmable Networking

- Language for reasoning about RA.
  - Multiple topologies **/ Topology-agnostic**
  - Verified toolchain **/ Preserving trust**
- Network-related abstractions for:
  - **Path** — might be unknown, and paths might change over time.
  - **Place** — might be unknown in advance, and can depend on path.
  - **Reachability** — need to predicate policy on specific nodes being reachable.

Remote Attestation
in Programmable Networking

- Language for reasoning about RA.
  - Multiple topologies.
  - Verified toolchain.

  ] **Copland**

- Network-related abstractions for:
  - **Path** — might be unknown, and paths might change over time.
  - **Place** — might be unknown in advance, and can depend on path.
  - **Reachability** — need to predicate policy on specific nodes being reachable.

Copland Example

$*\text{bank} : @_{\text{ks}}[\text{av us bmon}] \overset{++}{\sim} @_{\text{us}}[\text{bmon us exts}]$

Adversary confinement

$*\text{bank} : @_{\text{ks}}[\text{av us bmon} \rightarrow !] \overset{--}{<} @_{\text{us}}[\text{bmon us exts} \rightarrow !]$

- Bank is RP. Requests 2 measurements:
  - **First:** AV @KernelSpace verifies bmon that is in UserSpace
  - **Second:** bmon @UserSpace verifies browser extensions
- Evidence is not sent forward. (“- -” above “<”)
- Evidence is signed. (“!” symbol)

Remote Attestation
in Programmable Networking

- Language for reasoning about RA.
  - Multiple topologies.
  - Verified toolchain.

Copland

- Network-related abstractions for:
  - **Path** — might be unknown, and paths might change over time.
  - **Place** — might be unknown in advance, and can depend on path.
  - **Reachability** — need to predicate policy on specific nodes being reachable.

How to bridge?

NetKAT

- "Kleene Algebra with Tests"
  Combining Kleene Algebra and Boolean Algebra.

- NetKAT proposed as basic language to reason about SDN.

- Example: (Kozen, APLAS'14):
  t = sum of all link expressions
  p = sum of all switch policies
  then "packets from A can reach B" is encoded as:

  (switch=A; t(pt)*; switch=B) != 0

Example: Copland+NetKAT

$$*\mathsf{bank}\langle n, X\rangle : \forall hop, client : (@_{hop}[\mathsf{K}_{\mathsf{hop}} \blacktriangleright \mathsf{attest}(n)X \to !] \stackrel{-}{>}^{+} @_{\mathsf{Appraiser}}[\mathsf{appraise} \to \mathsf{store}(n)]]) \stackrel{*}{\Rightarrow}$$
$$@_{client}[\mathsf{K}_{\mathsf{client}} \blacktriangleright @_{\mathrm{ks}}[\mathsf{av\ us\ bmon} \to !] \stackrel{-}{<}^{-} @_{\mathrm{us}}[\mathsf{bmon\ us\ exts} \to !]]$$

- Quantifying over places
  (NetKAT: weak “switch policy”)

- Abstracting over paths:
  (NetKAT: Kleene star over product of places and links)

- Boolean tests as predicates

Ongoing work

- Software prototype.
  Ack: **Alexander Wolosewicz**

- Preparing physical testbed experiments.
  Ack: **Sean Cummings**

Conclusion

- Programmability: double edged sword
- This paper: How to leverage Remote Attestation in Programmable Networking:
  - Focus on Specification and Mechanism.
- *Scratches the surface*: much more work needed across different technical domains.
- Future work: stress-testing use cases, scoping-out implementation, and formalizing Copland+NetKAT hybrid.

ILLINOIS TECH

Computer Science Department

Nik Sultana
http://www.cs.iit.edu/~nsultana1

Extra slides

Copland Example

$$* \text{ bank} \;:\; @_{\text{ks}}[\text{av us bmon}] \overset{+}{\sim}\!\overset{+}{} \; @_{\text{us}}[\text{bmon us exts}]$$

Copland Example

$$* \text{ bank} : @_{ks}[\text{av us bmon}] \overset{+}{\sim}\!\!\overset{+}{} @_{us}[\text{bmon us exts}]$$

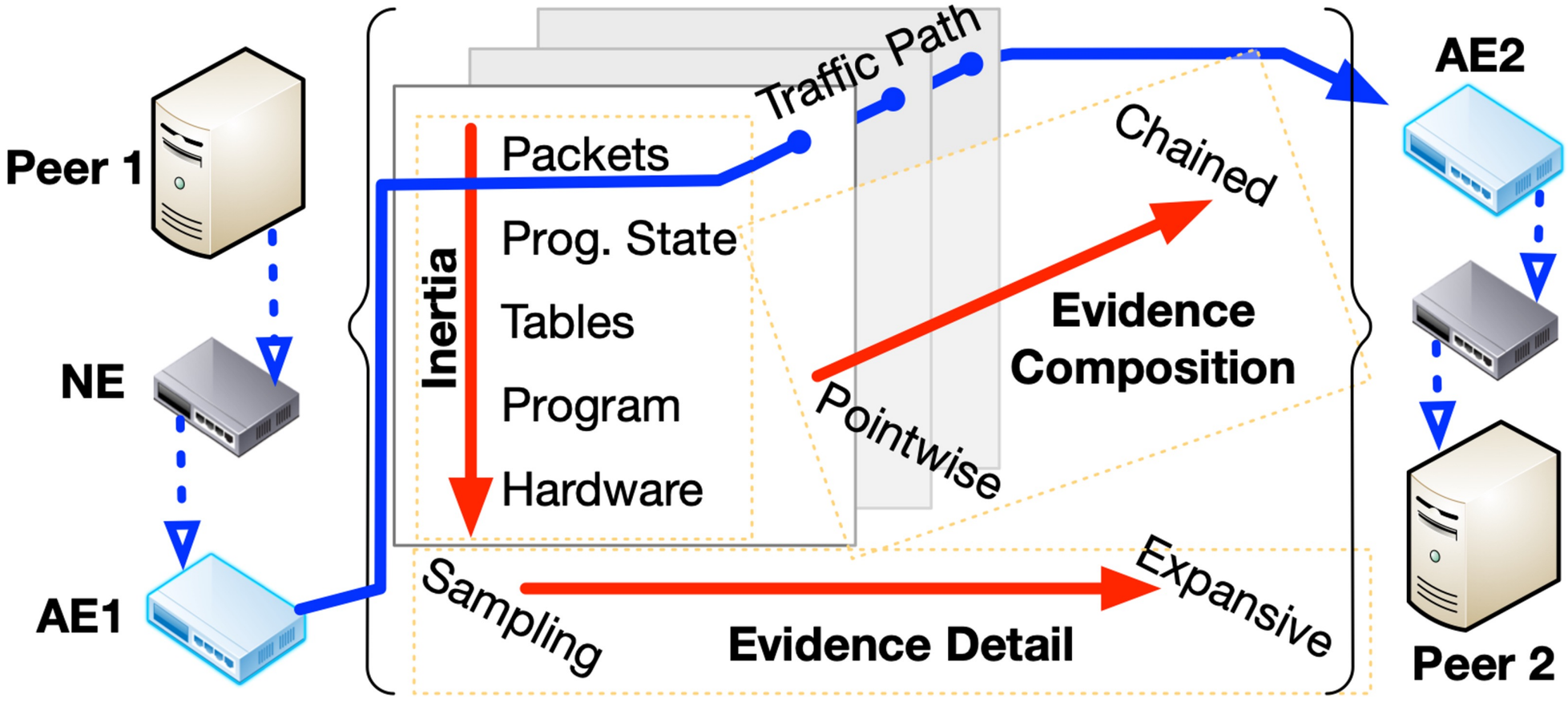
- Bank is RP. Requests 2 measurements:
  - AV @KernelSpace verifies bmon that is in UserSpace
  - bmon @UserSpace verifies browser extensions
- But this can be tricked:

Copland Example

$$* \text{ bank} : @_{ks}[\text{av us bmon}] \overset{+}{\sim}\overset{+}{} @_{us}[\text{bmon us exts}]$$

- Bank is RP. Requests 2 measurements:
  - AV @KernelSpace verifies bmon that is in UserSpace
  - bmon @UserSpace verifies browser extensions
- But this can be tricked:
  - Use a modified bmon.
  - Have “bmon us exts” return “OK”, and then swap bmon to have AV certify unmodified bmon.

PISA + Remote Attestation



Remote Attestation

IETF WG on
Remote ATtestation ProcedureS

https://datatracker.ietf.org/doc/draft-ietf-rats-architecture/

3. Architectural Overview

Figure 1 depicts the data that flows between different roles, independent of protocol or use case.

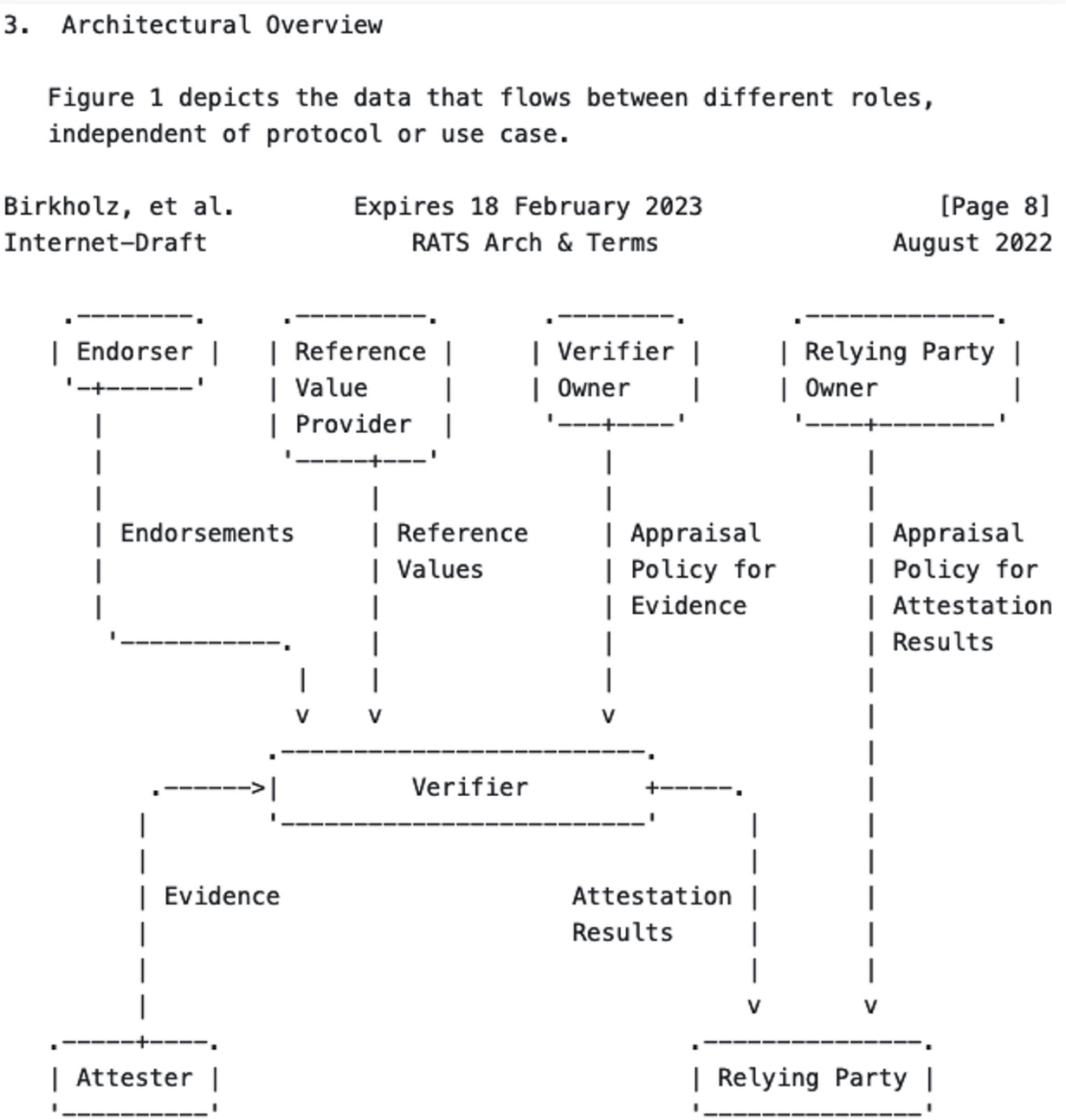


Figure 1: Conceptual Data Flow

Motivation for this paper:
How to adapt RA to *programmable* networks?