KU INSTITUTE FOR INFORMATION SCIENCES
The University of Kansas

ripple | University Blockchain Research Initiative

Science of Security LABLETS

DARPA

NSF

KANSAS CITY NSC National Security Campus

Attestation and Time

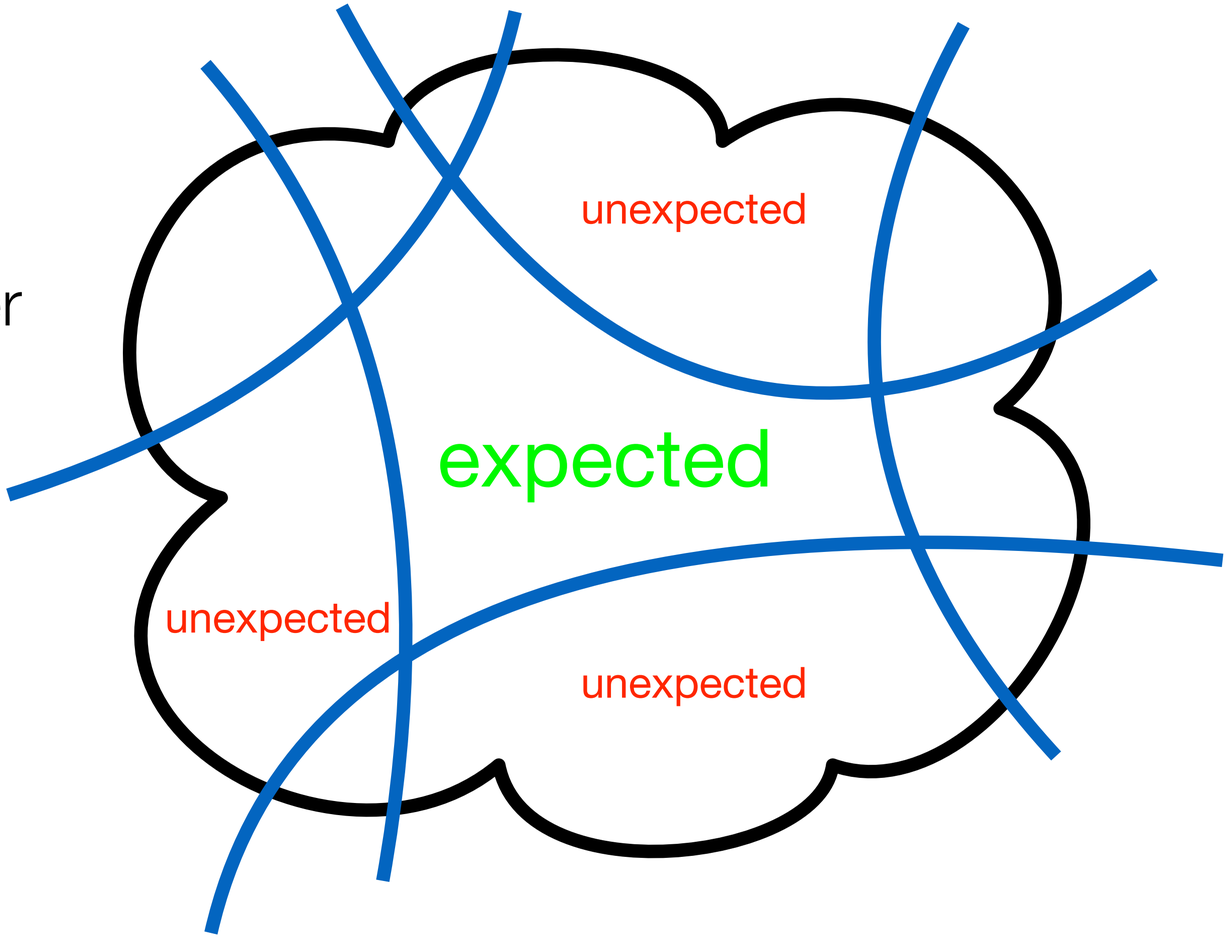
Dr. Perry Alexander
Director, Institute for Information Sciences
AT&T Foundation Distinguished Professor, Electrical Engineering & Computer Science
The University of Kansas
palexand@ku.edu

“Time is relentless and undefeated.”

–Unknown

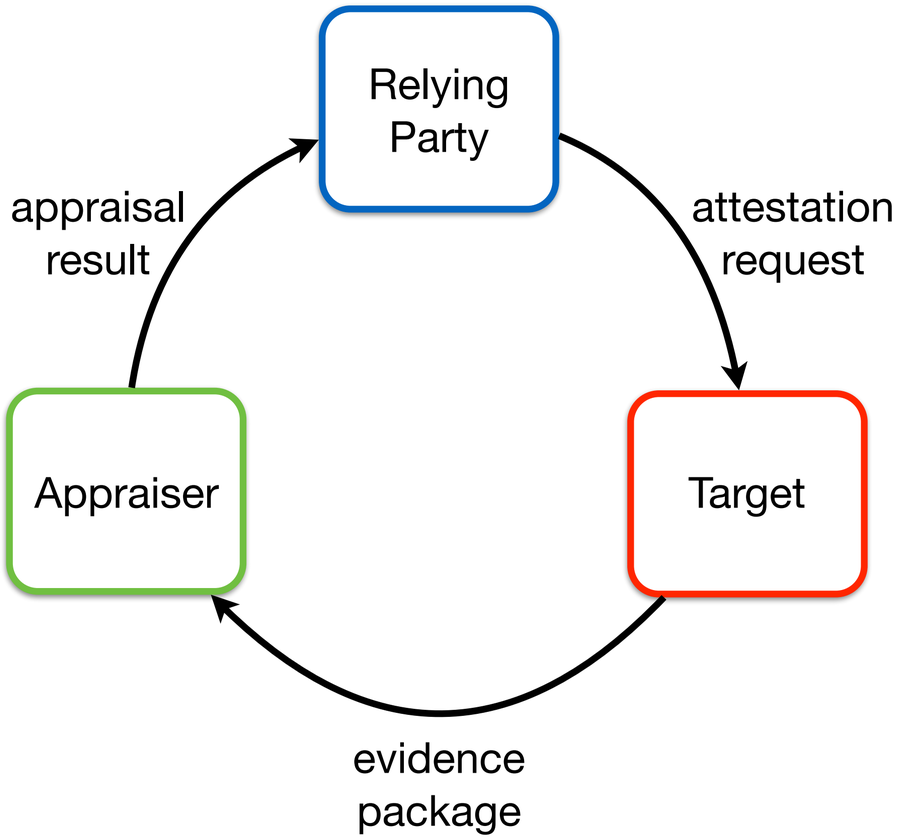
System Appraisal

- Building a perfect system is ~~hard~~ impossible
  - The Inevitability of Failure (Loscocco et. al.)
- Building an expected system is easier
  - not perfect, but expected
  - correct boot from good, known components
  - a good initial state
- Maintaining expectation is hard
  - stays in expected states as it runs
  - while it interacts with the world
  - good reachable states
- Semantics of expectation
  - where might my system be?
  - where might my adversary be?
  - evaluating expectation over time



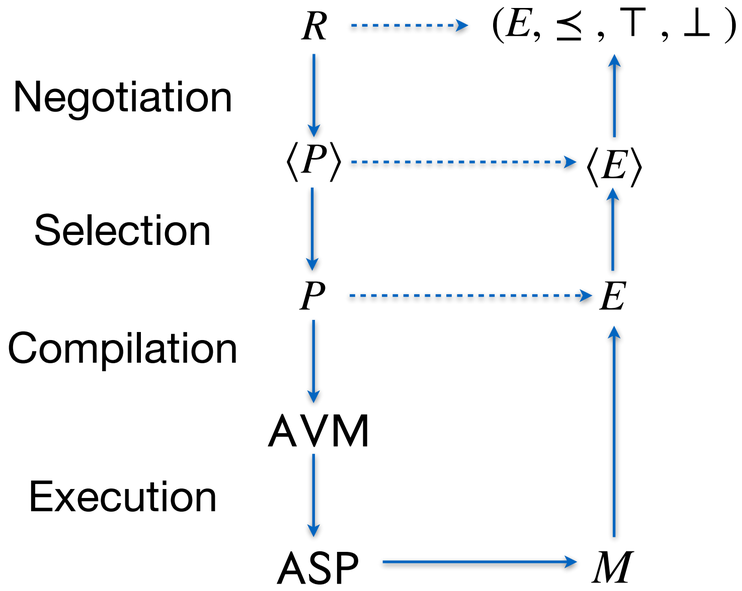
Remote Attestation

- Measurement and Attestation
  - gathering evidence of booting system
  - gathering evidence of executing system
  - gathering evidence of evidence gathering
- Appraisal
  - evaluating evidence of expectation
  - is a system behaving as expected?
- Today - Boot and runtime appraisal
  - relying party requires trust
  - attestation generates evidence
  - appraiser checks expectations over evidence
- Tomorrow - Systems over time
  - records and ledgers for evidence
  - system and local manifests for configuration
  - flexible mechanism for system appraisals



Semantics of Remote Attestation

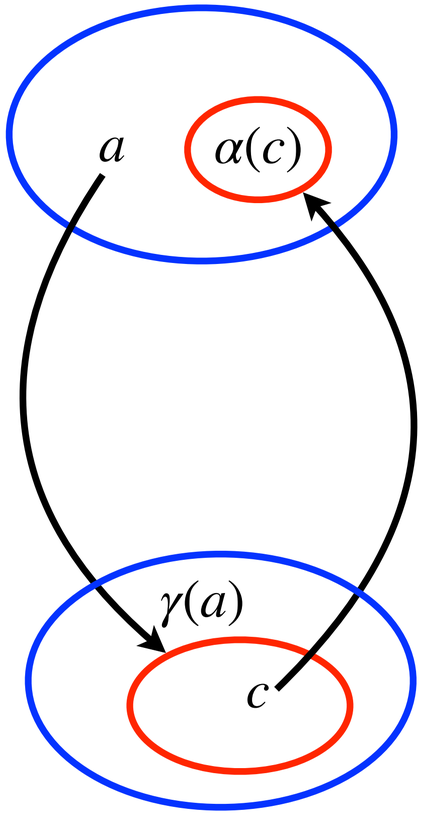
- Copland-Based Attestation
  - ensuring the protocol ran correctly
  - a formal DSL for attestation protocols
  - rich, precise semantics in a simple language
  - verified/synthesized Copland environments
- Manifests, Executability and Negotiation
  - ensuring the correct protocol runs
  - manifests formally define attestation systems
  - executability formally defines protocol soundness
  - negotiation determines a best protocol for two parties
- Executability is decidable for protocols and manifests
  - statically ensures a protocol will execute
  - statically ensures what evidence type it will provide
  - considers ASP selection, communication, and access control
- Negotiation among attestation managers
  - know what protocols run under selection and access control policy
  - know what evidence is produced
  - choose a mutual based protocol or fail



Semantics of Evidence

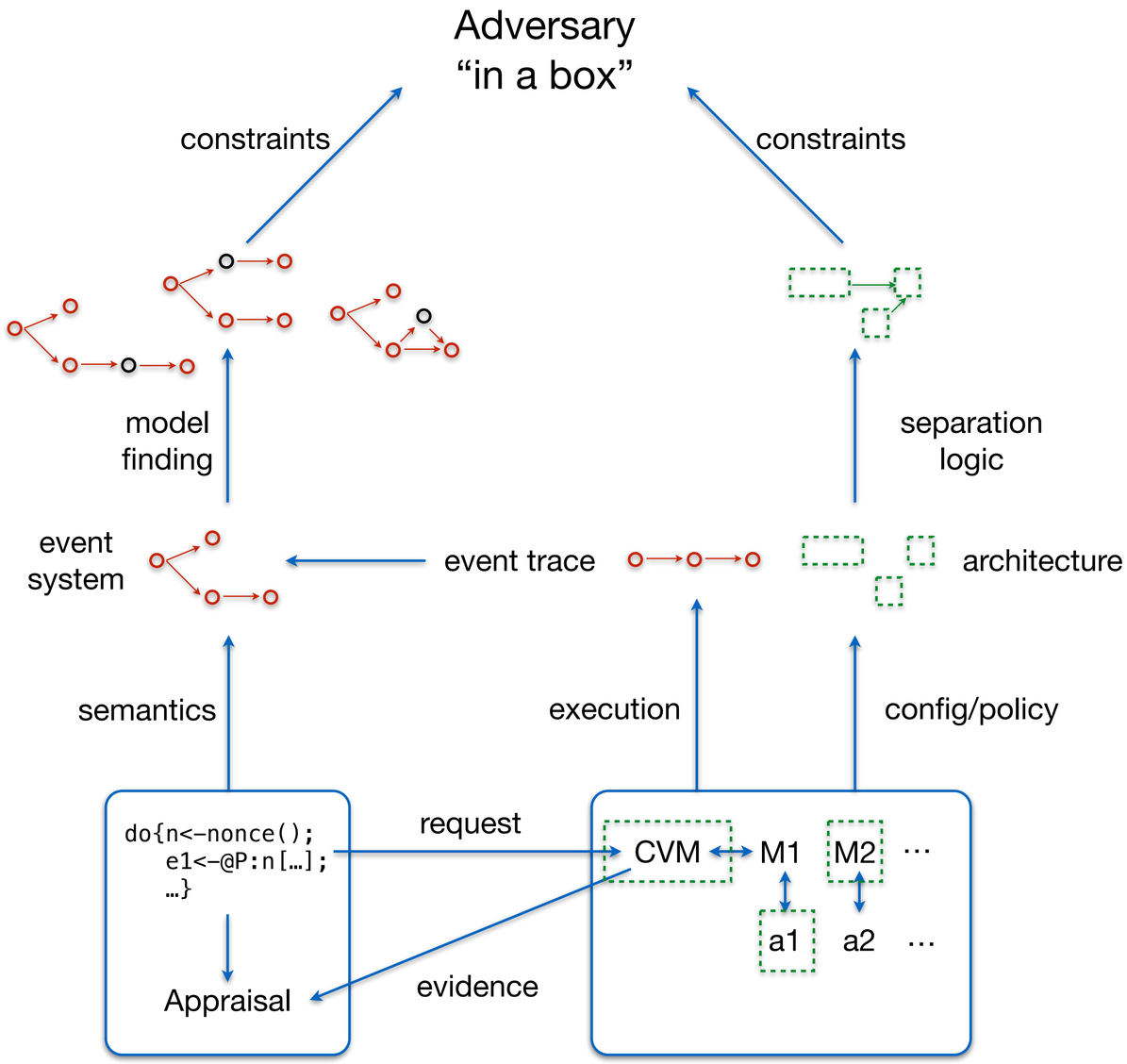
- Good Measurement is a Sound Abstraction
  - Galois Connection is a good model
  - measurement is an abstraction
  - appraisal is a concretiziation
- Composing Evidence
  - sequential execution (`p->p'`)
  - evidence preserving sequential (`p+<+p'`)
  - parallel (`p+~+p`')
  - remote (`@P(p)`)
  - temporal order matters!!
- Meta-Evidence
  - signatures over evidence and nonces
  - ensures integrity of evidence and order
  - evidence describing evidence gathering
- Ranking Evidence
  - what evidence is preferred by appraiser and target?
  - supports choosing between executable protocols
  - rich information vs. constrained disclosure

$$(\alpha(c) \leq a) \Leftrightarrow (\gamma(a) \geq c)$$



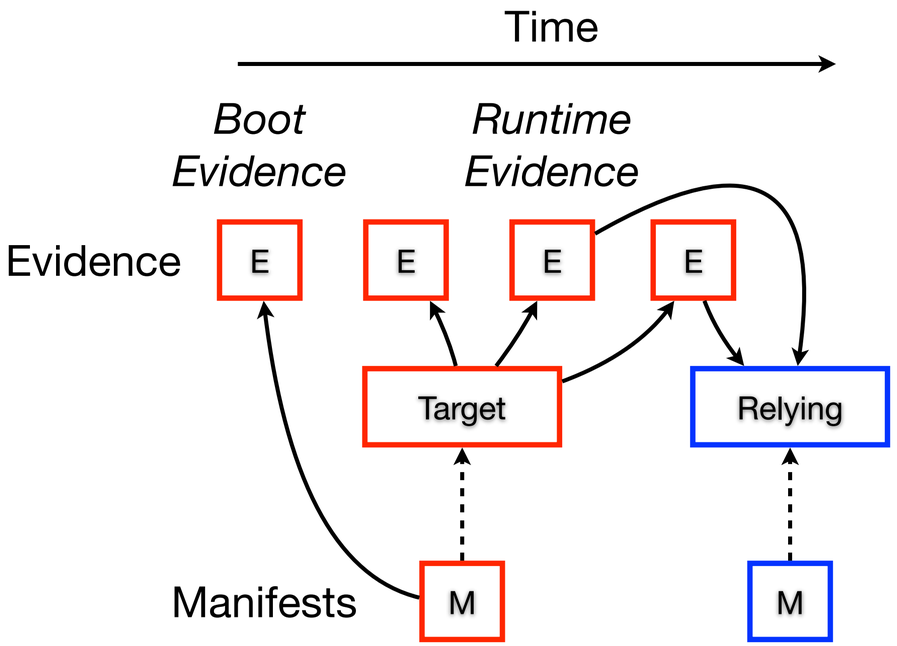
Systematic Analysis

- Correct attestation platform (Coq,CakeML,seL4)
  - correctly executes Copland protocols and appraises results
  - verified with respect to Copland semantics
  - synthesize from Coq to CakeML
- Protocol Analysis (Coq,Copland)
  - adversaries acting among protocol actions
  - adversaries accessing protected information
- Model Finding (CHASE)
  - discovers adversary models consistent with attestation protocols
  - allows evaluation of potential adversary behavior outside the attestation protocol
- Separation Analysis (seL4)
  - CAmkES specifications define allowed communication
  - synthesize or analyze architectures to evaluate allowed interaction
- Adversary “in a box”
  - analysis specifies what an adversary might do in the presence of the protocol
  - “the box” constrains the adversary making them do things they don’t want to
  - balance the level of constraint against the threat



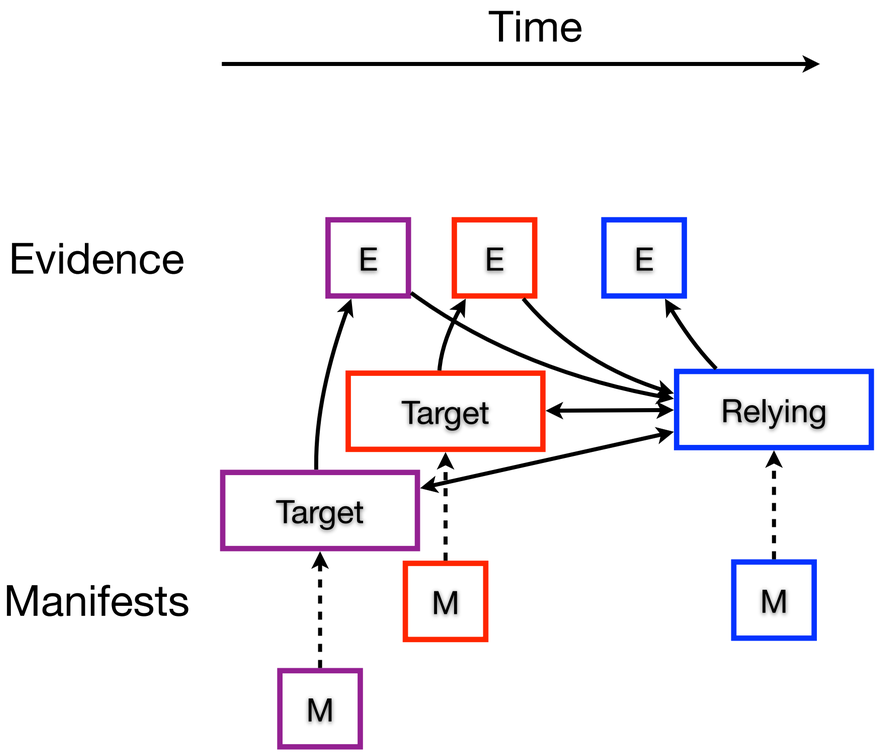
Attestation and Appraisal

- Manifests configure attestation components
  - individual components
  - systems of components
- Ledgers record evidence
  - measurement of component state
  - structured data for appraising systems
  - stored over time
- Boot evidence memorializes startup
  - evidence of good components
  - evidence of boot order
  - initial state
- Runtime evidence memorializes execution
  - moving away from boot state
  - evidence of runtime behavior
  - reachable states



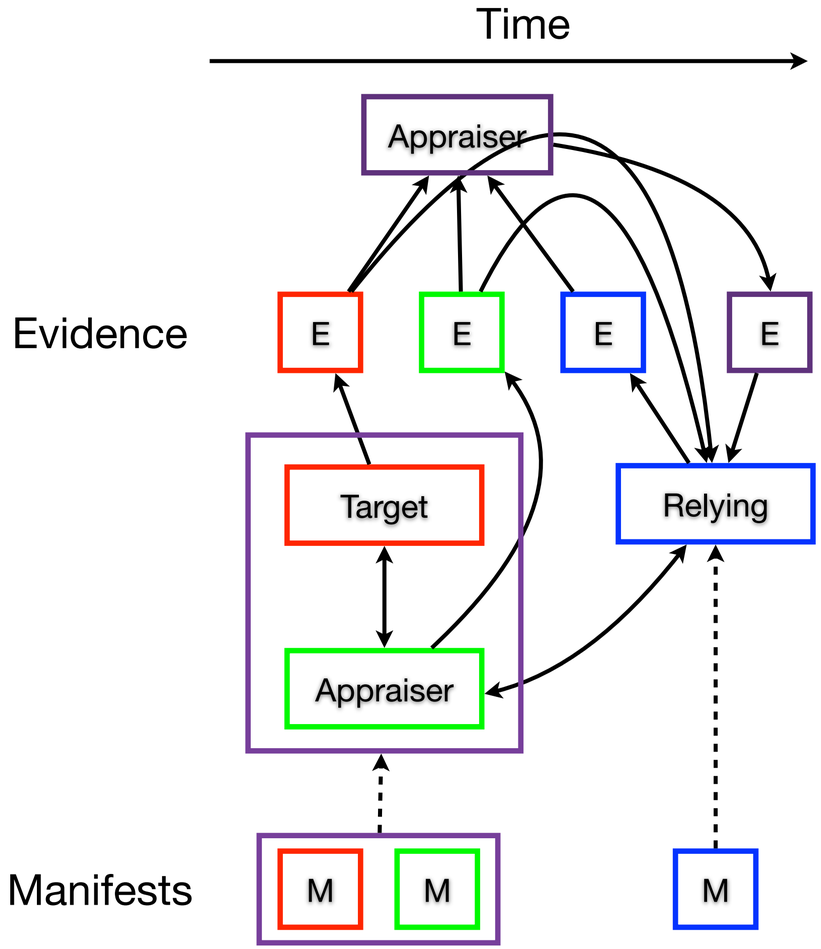
Composing Evidence

- Components request measurements
  - on demand evidence from targets
  - custom evidence for relying party
  - caching increases efficiency and increases complexity
- Components appraise evidence
  - evidence from target
  - evidence from ledger
  - baseline from manifest
- Components produce meta-evidence
  - signing for integrity and identity
  - record ordering assurance
- Components share results
  - updated evidence records
  - new external perspective



Targeted Appraisal

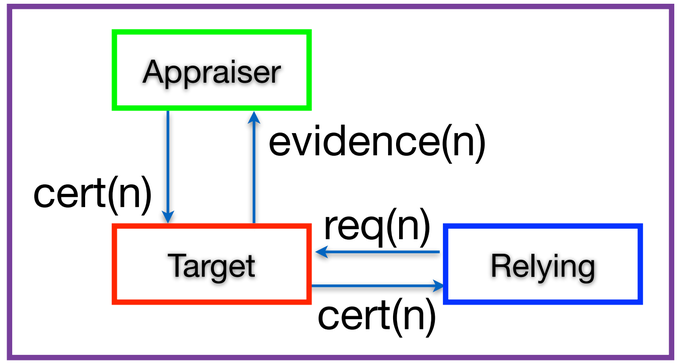
- Manifests configure multi-component systems
  - multiple component manifests
  - allowed communication
  - measurement responsibilities
  - service availability
- Specialized Components
  - target systems
  - attestation and appraisal components
  - out-of-band attestation and appraisal
- Heterogeneous evidence
  - consumed directly
  - written to the ledger
  - cached for later use



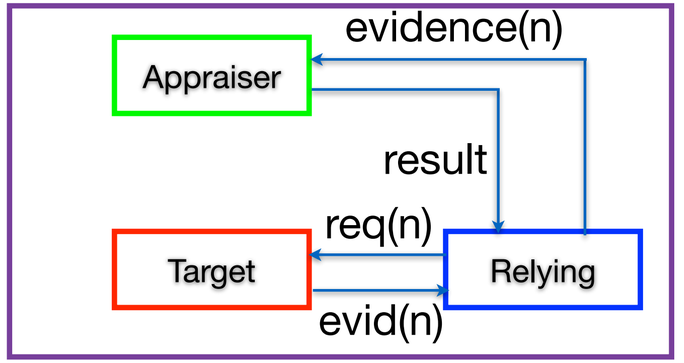
Flexible Mechanisms

- Attestation Protocol templates for common shapes
  - Layered
  - Certificate-Style
  - Cached
  - Background Check
- Implemented using communicating Attestation Manager instances
  - attestation service providers for measurement and other services
  - requires “plumbing” for communication, scheduling, and access control
- Principled composition
  - assembling attestation ecosystems
  - scaling to the enterprise
  - assessing impacts on adversaries

```
*P0,n: @P1[(attest P1 sys) ->
          @P2[(appraise P2 sys) ->
             (certificate P2 sys) ]]
```

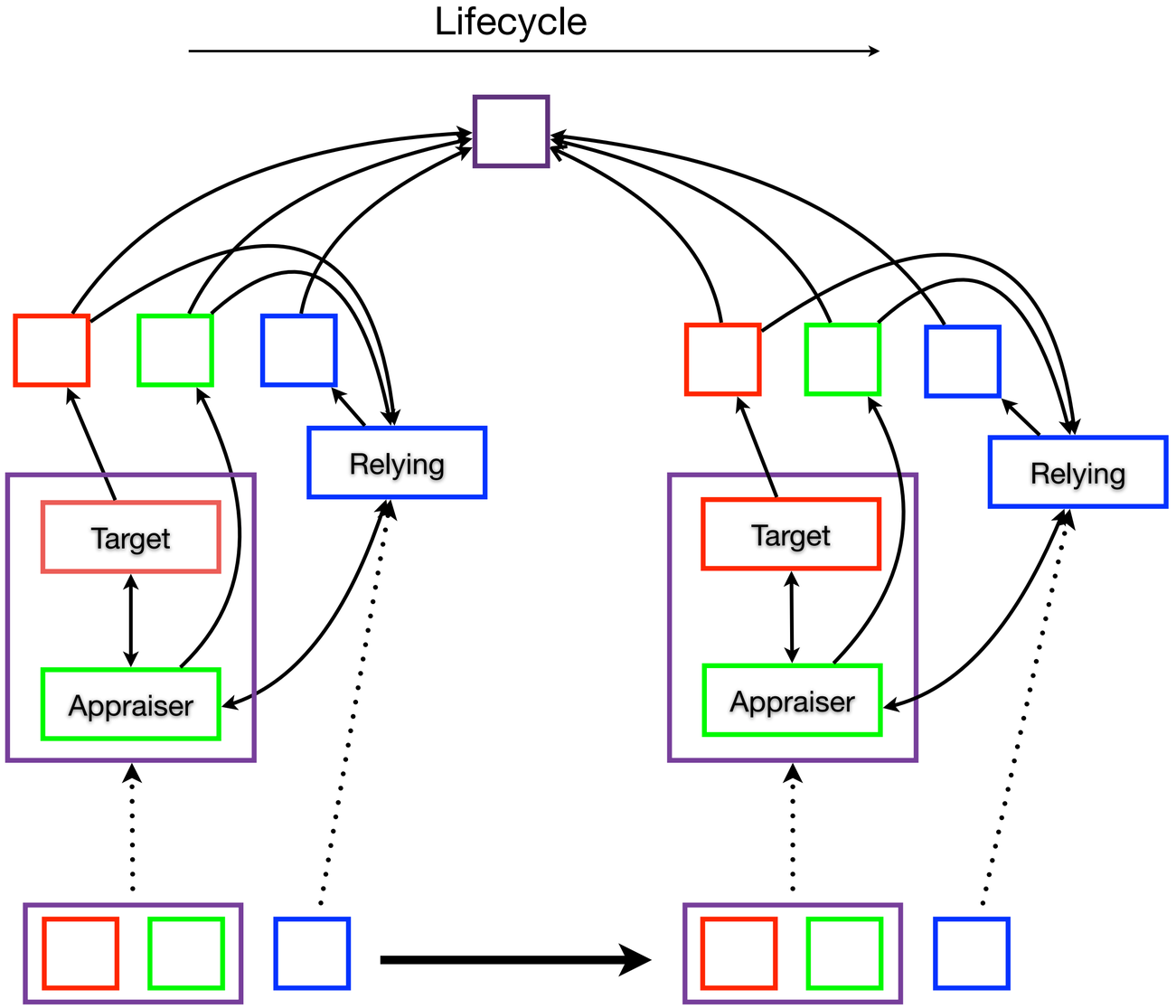



```
*P0,n: @P1[(attest P1 sys)] -> @P2[(appraise P2 sys)]
```



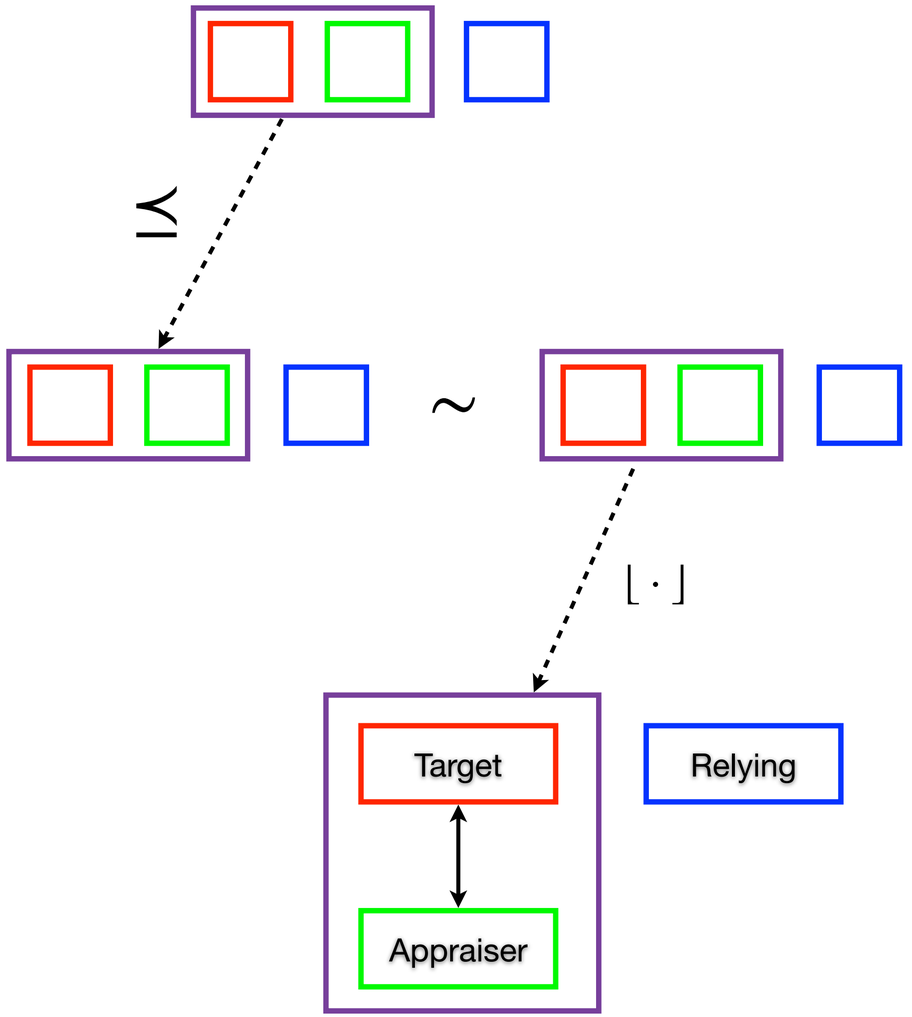
Lifecycle Attestation

- Systems & Environments change over time
  - requirements to implementation
  - retrofit, upgrade, legacy systems
  - sitting on the shelf, recertification
- Attestation & Appraisal should track changes
  - static verification and simulation
  - functional testing
  - constraint checking
  - certification and recertification
- Lifecycle Attestation
  - requirements elicitation through retirement
  - move attestation among lifecycle stages
  - combine evidence among lifecycle stages
  - complete system history



Lifecycle Attestation

- Manifests define systems in context
  - target - system of interest
  - appraiser - means for evaluating target
  - relying party - system consumer
- Manifests can be related
  - simulation relations define good abstractions
  - safety, liveness and invariant properties describe common system requirements
- Manifests can be synthesized to implementations
  - configuring systems (SVP'06)
  - compile from traditional languages to systems
  - model-to-implementation synthesis
- Manifests can be transformed
  - design lifecycle steps
  - manifest-to-manifest transformations
  - workflows



Some Open Hard Questions

- What is good evidence?
  - high integrity
  - sound abstractions
  - constrained disclosure
- How do we gather evidence?
  - remote attestation
  - monitoring and logging
  - sampling of other forms
- How “long” does evidence have utility?
  - measures other than time
  - re-measurement strategies
  - seeding evidence caches
- How do we compose evidence?
  - from different components
  - from different abstractions
  - over time and across system events
- How does evidence relate to adversary behavior?
  - how big is the adversary’s box?
  - can we monitor complex supply chains?
  - can we automatically analyze adversary behavior?
- Attestation over system lifecycle
  - from concept to decommission
  - move models among lifecycle stages
  - generalize measurement and attestation