AAA and Transport

Wes George

Brad Dreisbach

AAA

- Must be agnostic of controller architecture and use cases
- Controls all interaction between Apps using I2RS API(s) and NE/Controller
- Protects the network from the apps and vice versa
- Secures/validates connections
- Trust model
- Unclear whether existing AAA protocols can support/be extended to this info model

Authentication

- Bi-directional Key/Cert based
  - Including all standard PKI housekeeping
- Define levels of trust / standard roles
  - Hierarchical / inheritance modeling

Authorization - read

- Info allowed/prohibited
  - Granularity
- Frequency/rate limit
- Push vs pull
- Events & Triggers
  - Subscription vs publication
- Granular down to subsets/supersets of services, areas of config

Authorization - Write

- Most of the same things from Read
- Limits on:
  - Rate of change
  - Range of values/attributes
  - How much state must be held in NE or Controller
- Priority and precedence
  - Locking and single control (subsection or entire)
    - Required vs requested
- Rollback and Diffs

Authorization - other

- Clears:
  - Counters
  - Sessions
  - State
- Troubleshooting
- Capacity and scale limitations
  - Where is this state kept?
  - Hard threshold vs soft limit

Accounting

- Who/When
- Why (comments on changes/requests)
- Failures:
  - Permissions
  - Problem (communication or system)
- Precedence relationships
  - Conflict resolution/tiebreaking

State

- Shared
  - Synchronous (near real-time/checkpointed)
  - Asynchronous
    - Triggered/event driven
    - Periodic
- Unshared/not needed (transactional)

Transport

- Secure:
  - Payload
  - Session (optional)
  - App layer certification and validation of data
    - Which data?
    - Is transport layer security enough?
      - Probably depends on data
- Reliability
  - TCP vs UDP
    - App Acks vs transport acks