SUPA Information Model

John Strassner, Huawei
Joel Halpern, Ericsson
Sven van der Meer, Ericsson
IETF96
Agenda

• Motivation for building an Information Model
• I-D Status and Open Issues
• Next Steps
Motivation (1)

• Information Model is independent of language, protocol, repository, and content and structure of policy

• Without an Information Model…
  – Resulting data models will be silos, making interoperability difficult
  – Different types of policies (e.g., imperative, procedural, declarative, functional) will themselves be silos
  – Different actors want to author different policies using different grammars
  – No Interoperability with Chef/Puppet/, AWS Cloud Formation Templates, …
Motivation (2)

- Policies are used by multiple actors
  - App developers, operators, security and compliance teams, administrators, end-users, … each has different concepts and terms

- Policies exist at different levels of abstraction
  - Per-port, -device, -network, -VM, -application, -service, …

- Different Policies exist for different operations on the same device
  - Monitoring vs. configuration vs. audit
  - Deployment vs. backup vs. provisioning vs. billing vs. retirement …

- Policies focused on different technologies and vendors must be able to work collaboratively
  - Requires a common set of concepts and vocabulary across domains
  - An E2E policy affects multiple actors, technologies, and vendors

- Standardized policies help heterogeneous systems interoperate
Agenda

• Motivation for building an Information Model
• I-D Status and Open Issues
• Next Steps
Main Changes from -06 (Individual Draft)

- Renamed SUPAVendorDecoratedComponent to SUPAGenericDecoratedComponent
  - Its function was made more generic

- Clarifications made in response to questions from the SUPA mailing list

- Relationship multiplicity has been fine-tuned
  - Otherwise, class model is stable

- Produced a **preliminary** YANG model (covered in Joel’s presentation)

- Various additional typos have been fixed
Main Changes from -00 (WG Draft)

• The Declarative Model Has Ceased To Be
  – Even removed the Appendices
• Introduction rewritten and clarified
• Redesigned SUPAPolicyVersionMetadataDef
  – Otherwise, class model is stable
• Added Fully Qualified Path Names
• Various additional typos have been fixed
• Content now stable
GPIM Class Hierarchy

- Policy Rules
  - Policy Rule Components
    - Pattern for adding 0 or more PolicyComponents to PolicyClause
  - Metadata can be attached to ANY SUPAPolicyObject
    - Collection of PolicyComponents
    - Represents vendor-specific functionality
  - Canonical Form for PolicyClause
    - Encodes content as attributes
  - Statement building block

(SUPA Information Model)
EPRIM Class Hierarchy

(Class of another model that SUPA is integrating into)

- SUPAPolicyObject (5.2)
  - SUPAPolicyStructure (5.3)
    - SUPAECAPolicyRule (6.4)
      - SUPAECAPolicyRuleAtomic (6.5)
      - SUPAECAPolicyRuleComposite (6.6)
    - SUPAPolicyComponentStructure (5.6)
      - SUPAPolicyClause (5.7)
        - SUPABooleanClause (6.7)
          - SUPAECAPolicyRuleAtomic (6.8)
          - SUPAECAPolicyRuleComposite (6.9)
    - SUPAPolicyComponentDecorator (5.9)
      - SUPAECAComponent (6.10)
        - SUPAPolicyEvent (6.11)
        - SUPAPolicyCondition (6.12)
        - SUPAPolicyAction (6.13)

ECA (Imperative) Policy Rules

Individual ECAPolicyRule

Hierarchy of ECAPolicyRules

Individual Boolean PolicyClause

Hierarchy of Boolean PolicyClauses
Agenda

• Motivation for building an Information Model
• I-D Status and Open Issues
• Next Steps
Next Steps

• **Current Work**
  – Incorporating additional software patterns
  – Working on further optimizations
  – Adding lots of examples

• **Feedback is Always Welcome!**
Questions?

“Create like a god. Command like a king. Work like a slave”
- Constantin Brancusi
The GPIM

Base class for Policy Rules and Components of Policy Rules

Different types of Policy Rules

Different types of Policy Rule Components

SUPA Information Model - Strassner
Types of Policy Components

Policies are made up of PolicyClauses

PolicyClauses may be wrapped by other PolicyComponents

Concrete Subclasses, Concrete Subclasses
(e.g., SUPAEncodedClause) (e.g., SUPAPolicyCollection)
(object being wrapped) (wrapping object(s))
The Decorator Pattern

Concrete Subclasses, Concrete Subclasses
(e.g., SUPAEncodedClause) (e.g., SUPAPolicyCollection)
(object being wrapped) (wrapping object(s))
Decorated Policy Components

A

| SUPAPolicyComponentDecorator |

/ \ 
I I I
I
I
I

A I I C I I I

| SUPAPolicyTerm | I | SUPAPolicyCollection | I

| SUPAGenericDecoratedComponent | |SUPAECACComponent | I

(for defining I (for defining sets and/or I clauses in I groups of objects) I canonical form) I I I

C I A I

(for decorating concrete (for defining reusable subclasses of SUPAPolicyClause) event, condition, and action objects)
Metadata Hierarchy Overview
Constructing ECA Policies