I2NSF Framework - 05

April 2016

Edward Lopez (elopez@fortinet.com)

Diego Lopez (diego.r.lopez@telefonica.com)

XiaoJun Zhuang (zhuangxiaojun@chinamobile.com)

Linda Dunbar (linda.dunbar@huawei.com)

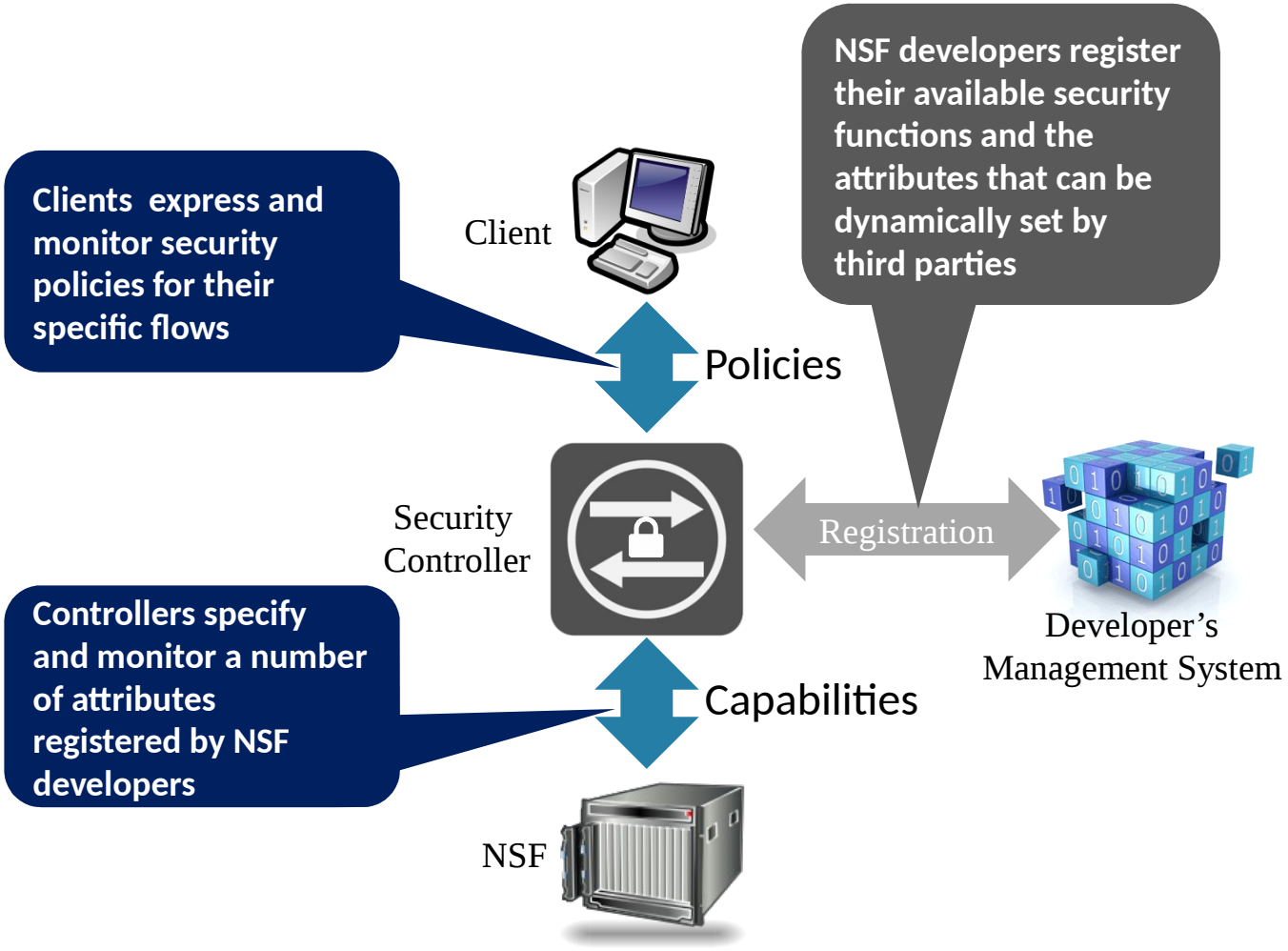
John Strassner (John.sc.Strassner@huawei.com)

Joe Parrott (joe.parrott@bt.com)

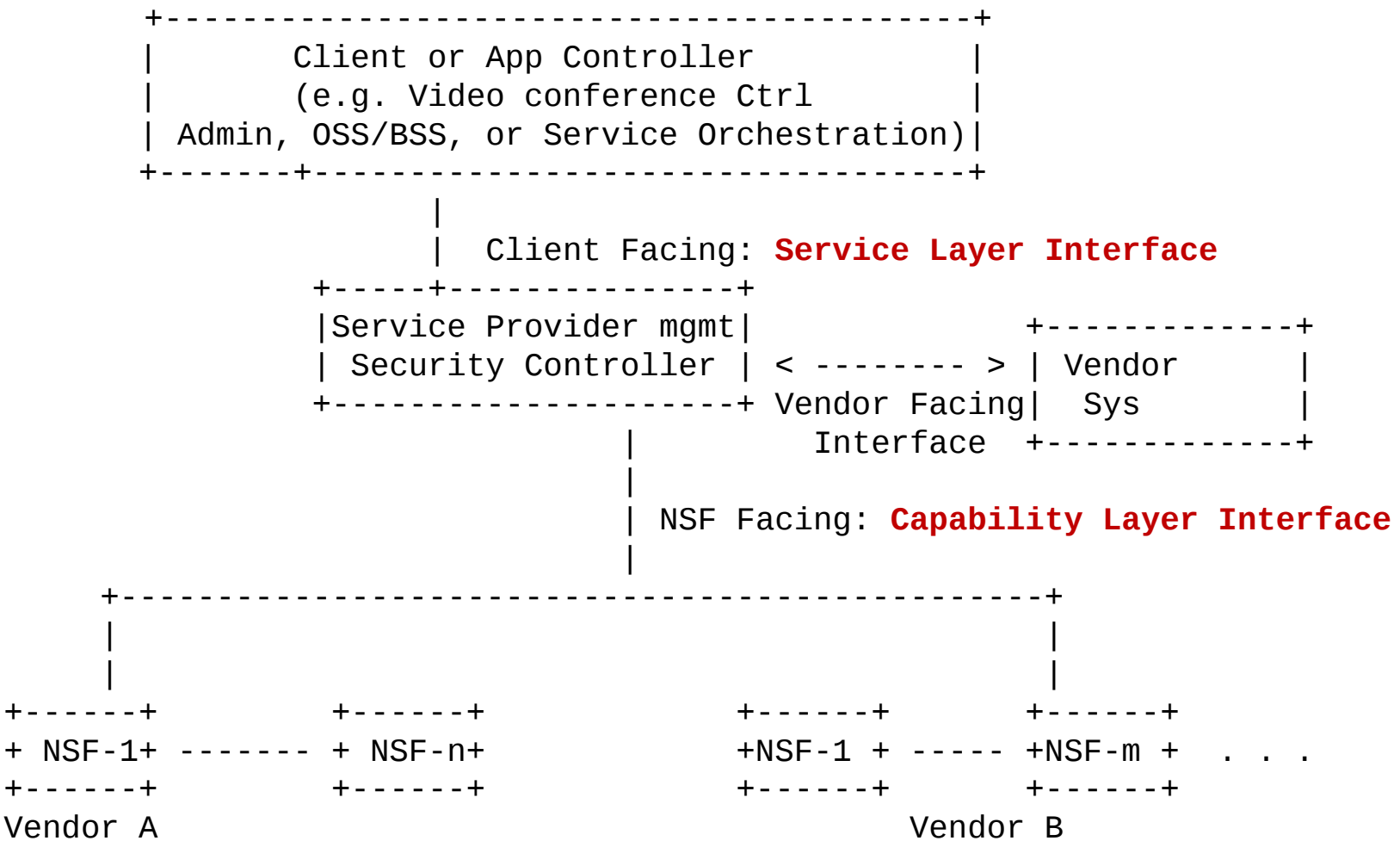
Ramki Krishnan (ramki_krishnan@dell.com)

Seetharama Rao Durbha (S.Durbha@cablelabs.com)

Major Components of I2NSF



Major Components as Shown in the Draft



Major Changes in - 05

- Terminology updated to conform to draft-hares-i2nsf-terminology-01
- Align the rule provisioning structure
  - To Event-Condition-Action rather then Subject-Object-Action-Function
  - Still using a packet-oriented paradigm focused on flow-based NSFs
- Removed reference to Service Layer extension from PCIM (RFC3060 or ITU-T X.1036)
  - Doubts about some flaws associated with PCIM

A More Detailed List of Changes

- Clarification of clarified packet- and flow-based processing
  - "This draft proposes that a rule provisioning interface to NSFs can be
- developed on a packet- or flow-based paradigm."
- State that packet- and flow-based NSFs can be standardized by using Event - Condition - Action (ECA) policy rule sets
- Definition of what an event, condition, and action mean in the context of policy rules, with examples
- Definition of what a policy rule is, and how it is used in I2NSF
- Clarifying that the rule sets and software interfaces of I2NSF aim to standardize the form and function of profile and signature files while supporting vendor-specific functions of each
- More detail on the Capability Layer Interface
- Clarification of vendor facing interface
  - More detail about vendor registration of their NSFs
- Additional security requirements

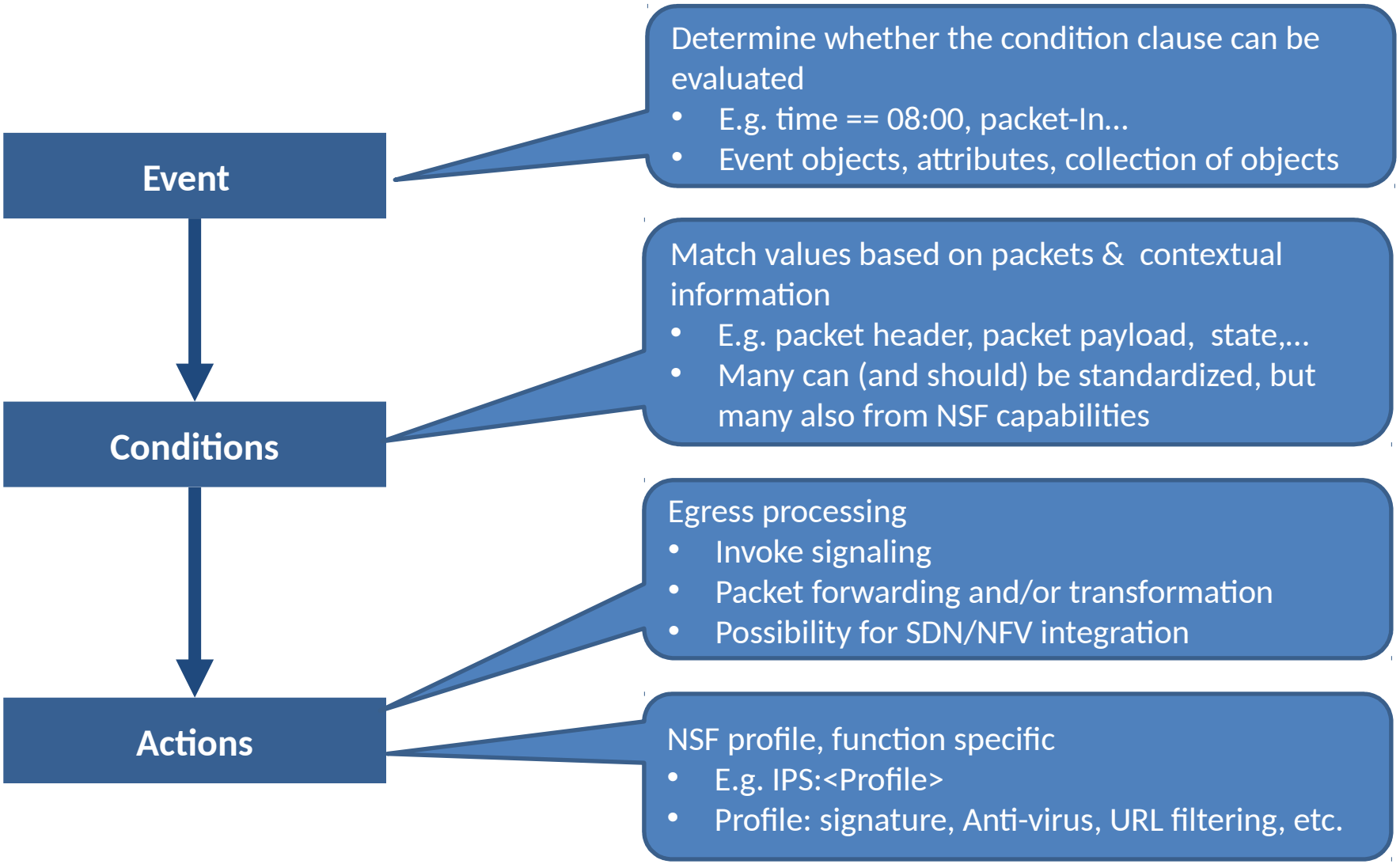
Packet-Oriented Paradigm for Flow-Based NSFs

- Rather than attempting to create a standard based on NSF classes, leverage flow-based programmability (SDN, NFV,…)
  - *Attackers don't follow standards*
  - Focus on rule provisioning for flow-based NSFs
- All NSFs, regardless of their objective, process
  - Packet headers
  - Packet payloads
  - Contextual and state information associated with packets

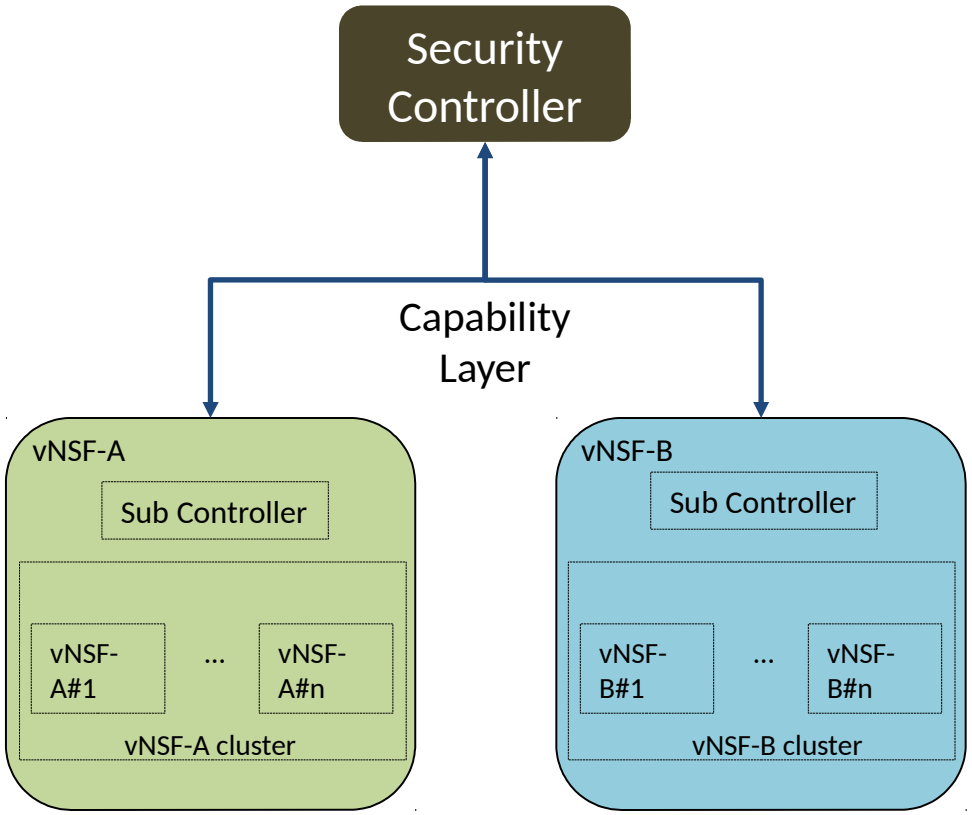
Three Types of NSF Interfaces

- Configuration
  - NSF internal configuration
  - Network attachment configuration
- Signaling
  - Status
  - Counters
  - Queries
  - Alerts
- **Rule Provisioning**
  - **Policies**
  - **Capabilities**
  - **Negotiation**

I2NSF Event – Condition – Action Rules



Considerations for vNSFs



- Single NSF can have multiple instantiations that are distributed across the network.
- Different rules/policies could be imposed to different instantiations.
- Each NSF may have its own sub-controller for all its instantiations
- Policies to one instantiation can be moved/copied to another NSF instantiation
- Multiple vNSFs (of different types or same type can share one physical server.
- Multiple vNSFs collectively together to enforce the rules for large flows

Other Aspects of the Framework

- Network connection between controller and NSFs
  - Closed and open environments
  - AAA, remote attestation in an open environment
  - Shall we consider client-controller as well?
- Rule considerations at each layer
  - Monitoring at the capability layer
  - Hints on service layer policies
- Capability negotiation
  - Considering the extension of CPP/CPNP (RFC 72976)