I2NSF Framework

July, 2015

Edward Lopez (elopez@fortinet.com)
DIEGO LOPEZ GARCIA (diego.r.lopez@telefonica.com)
XiaoJun Zhuang (zhuangxiaojun@chinamobile.com)
Linda Dunbar (linda.dunbar@huawei.com)
Joe Parrott (joe.parrott@bt.com)
Ramki Krishnan (ramki_krishnan@dell.com)
Seetharama Rao Durbha (S.Durbha@cablelabs.com)

Problems

- Unlike traditional networking device, network-based security functions (NSFs) do not operate relative to standards
  - Many evaluative bodies exist, which review the efficacy of network security product
  - Many regulatory/compliance directives call for the use of loosely defined classes of network security
- How do we define interfaces to devices that have no standardized implementations?

Potential For Imposed Constraints

- Narrowly defined NSF categories, or their roles when implemented within a network
- Attempts to impose functional requirements or constraints, either directly or indirectly, upon NSF developers
- Result in a limited lowest-common denominator approach, where interfaces can only support a limited set standardized functions, without allowing for vendor-specific functions
- Results in endorsing a best-common-practice for the implementation of NSFs

Packet-Based Paradigm for FlowBased NSF

- Rather than attempting to create a standard based on NSF classes, a solution may exist in provisioning packet processing
- All NSFs, regardless of function, process:
  - Packet headers
  - Packet payloads
  - Contextual and state information associated with packets

Three Sub-Interface Types

- Configuration
  - Device configuration
  - Network configuration
- Signaling
  - Status
  - Counters
  - Queries
  - Alerts
- **Provisioning**
  - **Capabilities**
  - **Policy**
  - **Object Configuration**

Suggested Framework - Provisioning

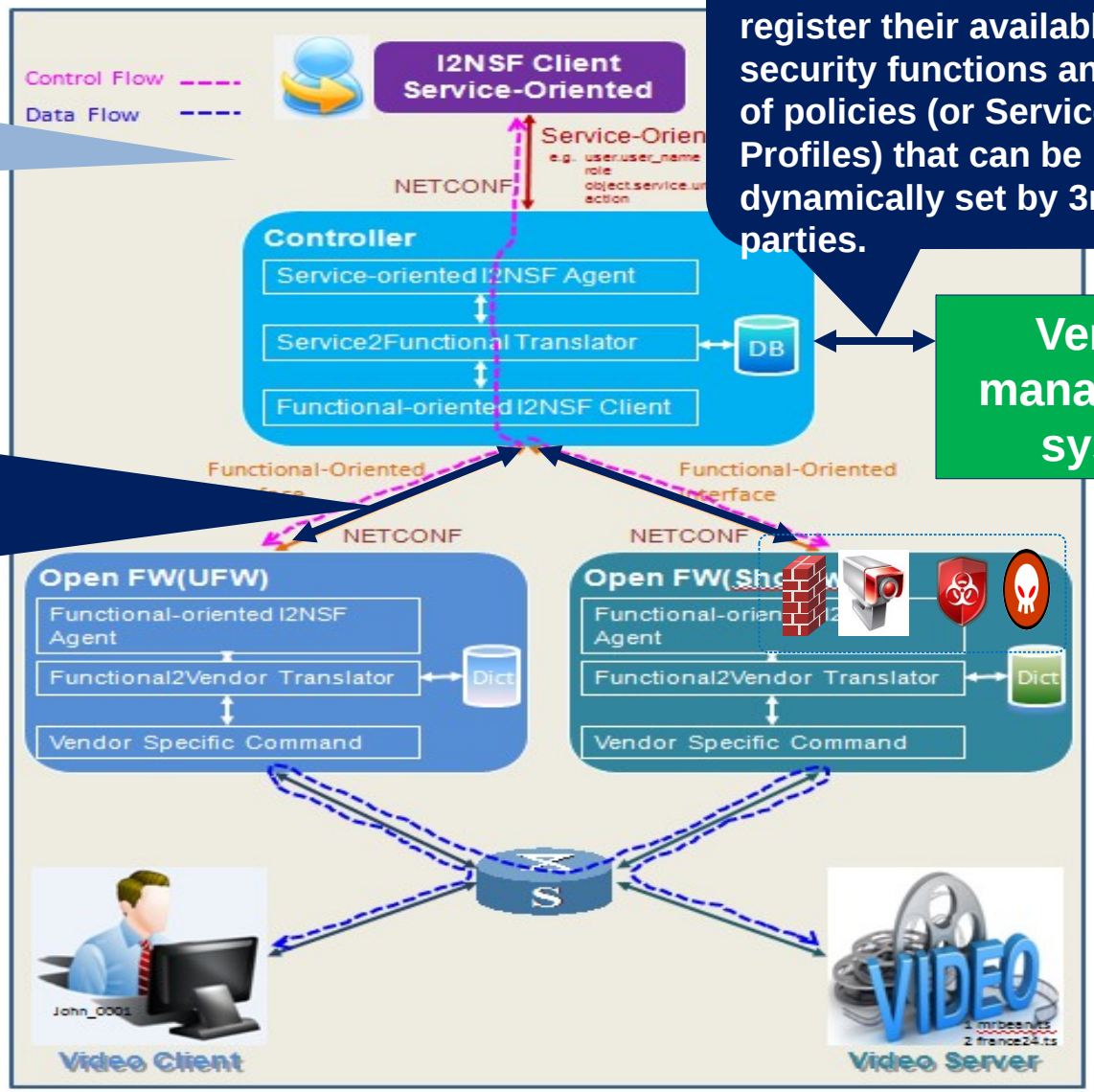
- Four root tree structure:
  - Subject – match values based on packet data
    - Packet header - Can be standardized
    - Packet payload - Provided by NSF capabilities
  - Object – match values based on context
    - Ex.: State, time, geo-location, etc.
    - Many can (and should) be standardized, but many also from NSF capabilities
  - Function – invoked security function
    - Defined by NSF capabilities
      - Function:Instance (ex. IPS:<signature base>)
  - Action – egress processing
    - Invoke signaling
    - Packet forwarding and/or transformation
    - Possibility for SDN/NFV integration

I2NSF Architecture

Security Service Layer
For clients or App Gateway to express and monitor security policies for their specific flows,

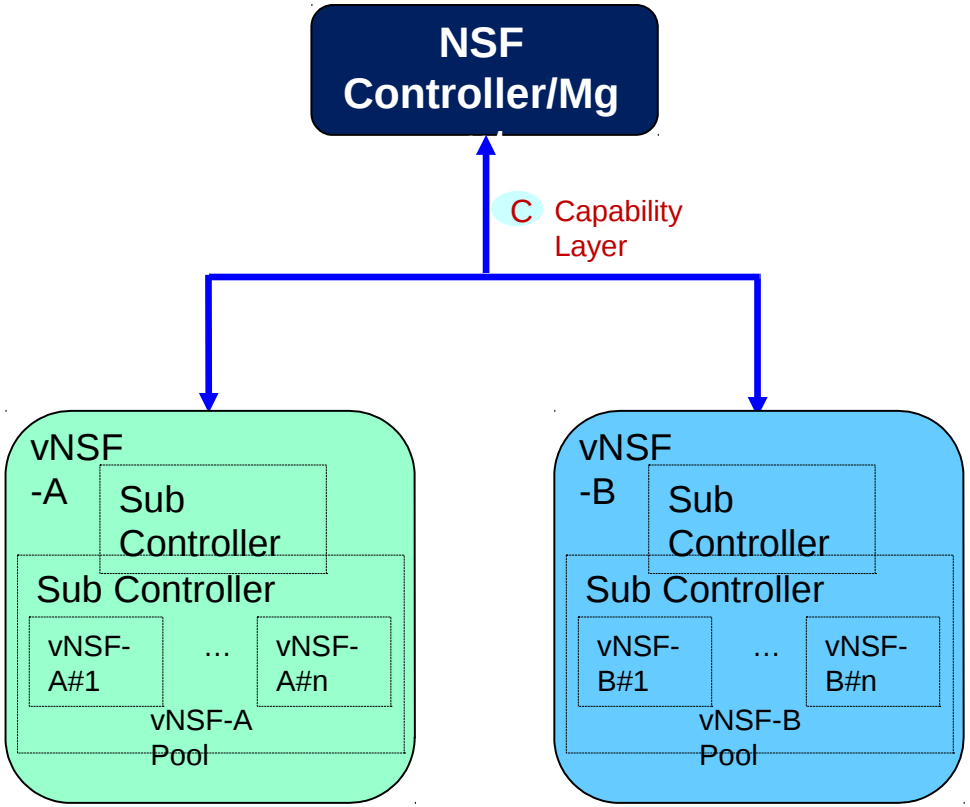
NSF Registration
For NSF vendors to register their available security functions and set of policies (or Service Profiles) that can be dynamically set by 3rd parties.

Capability Layer
For Controller to specify and monitor the limited number of attributes (or Service Profiles) that are allowed by the respective vendors to the NSFs



Vendor management system

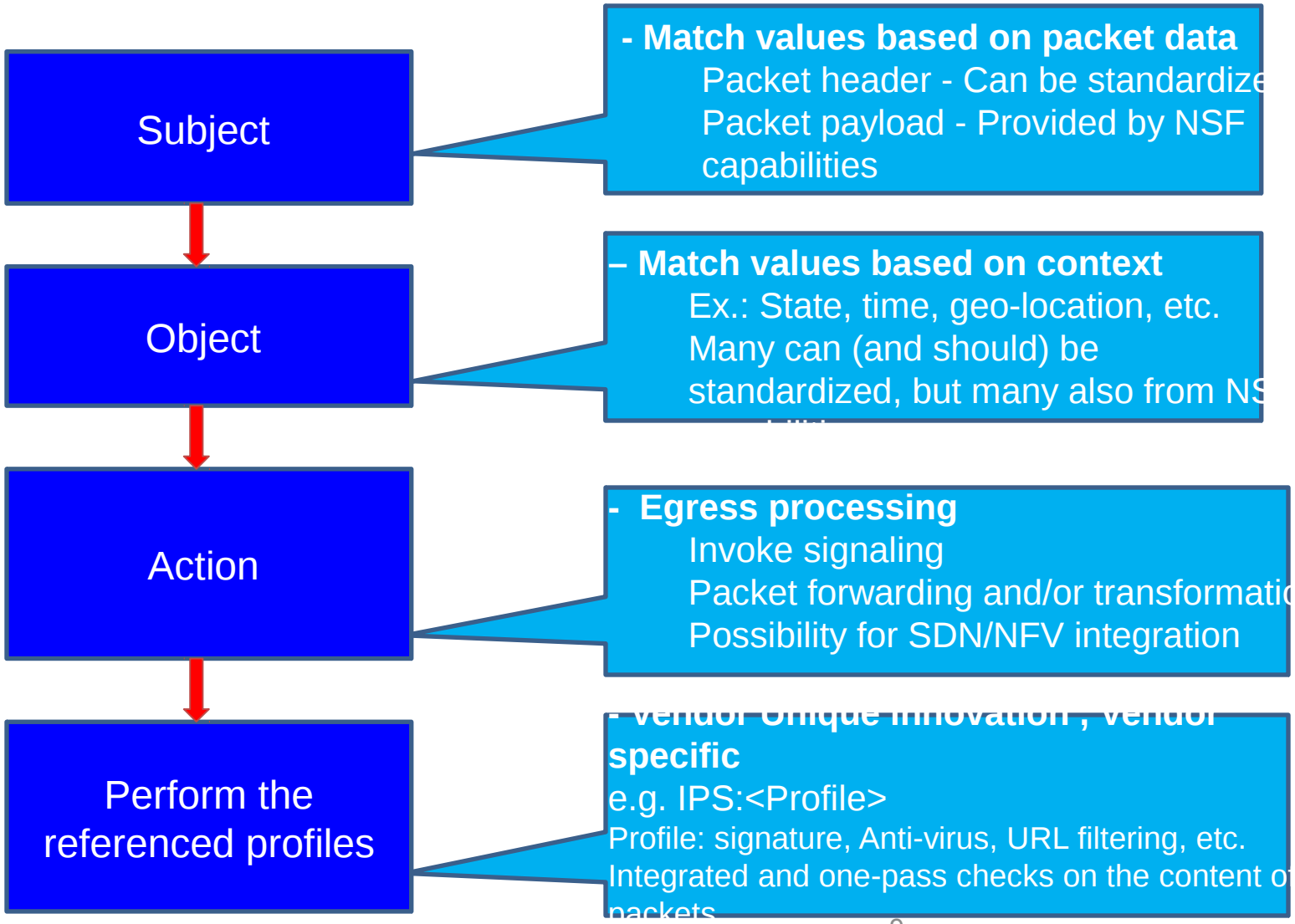
Interface to vNSFs



Characteristics：

- 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

NSF Provisioning Components Breakdown



Data Over the Registration Interface

Flow Based NSF Capability Index

Subject (header fields, payload, ..)	Object Context , external to bits/bytes in packets	Functions	Actions	Description
Layer 2 Header (Src/Dst, Vid, VxLAN, TRILL, EtherTypes) Layer 3 (Src/Dst, MPLS, GRE, IPv4/IPv6, …) TCP (Port, flags, SYN, FIN, ..) UDP layer (port, HTTP Layer	Access domain,	WebFilter, App Control Authentication Encryption, IPS/IDS/AV URL filter …	Pass/ drop/ mirror/ Statistics (report Destination)	Name-value pairs that describe Service capability, or the URL of a Heat template that describes the SF. Service layer attributes
	Time: Start/end/duration			
	Zone (corresponding header bits in the packets)			
	Tenant ID (corresponding header bits in the packets)			
	Application ID (corresponding header bits in the packets)			
		IETF PCP?	Open/Close	
		IETF TRAM		

Security Function Catalog DB

SF Catalog DB is built by Network SF Manager or orchestration system based on the SF Registration Process

Vendors	Function name	Type	instances	Flow based Security Polices Objects supported (Potentially IANA registered in the future)	Flow based Security Polices Action supported
X		FW		Layer 2/3/4; IETF PCP	Pass/drop
		IPS		Time span	
		IDS			
		Webfilter		HTTP, App ID	Call VideoOptimization
Y