# Information Model of Interface to Network Security Functions Capability Interface

draft-xia-i2nsf-capability-interface-im-03

Liang Xia Huawei

DaCheng Zhang Alibaba

Edward Lopez Fortinet

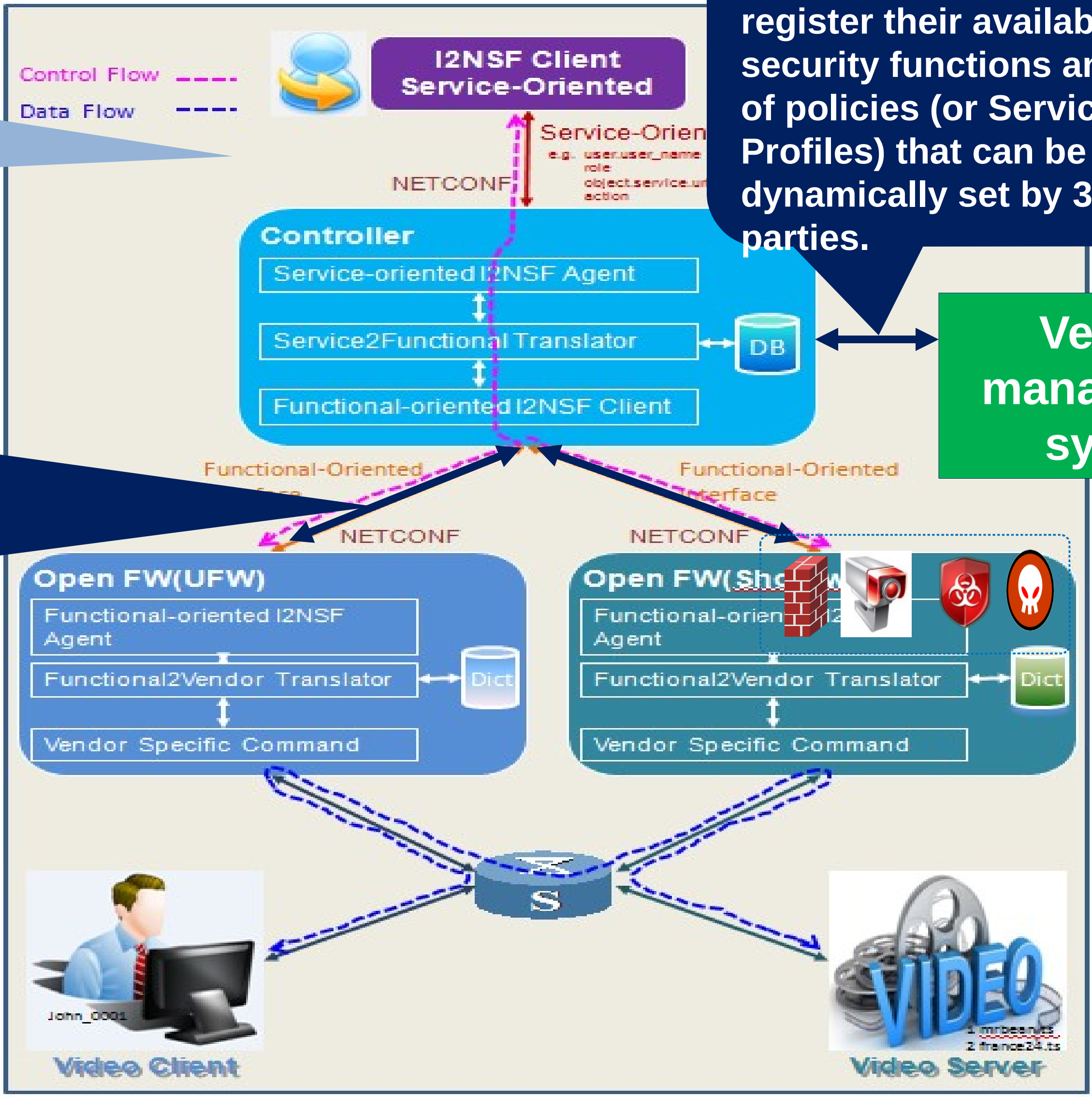
Nicolas BOUTHORS Qosmos

July 2015 Prague

# 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.



**Vendor management system**

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

# Current Dilemma of NSF Provisioning

- A lot of security vendors with its proprietary interface (i.e., management plane protocol, information model and data model);
- Various network security capabilities/functions provided by security vendors can not be integrated and applied as a whole. More seriously, more new network security capabilities are appearing;
- NSaaS market grows very fast, which requires the automatic provision of massive NSF instances with high efficiency and flexibility.

# Answer

- *A standard capability interface(by I2NSF)*
  - Decouple network security controller from s ecurity devices of specific vendors, and vice versa;
  - Only be oriented to the logic network securit y capabilities, independent with specific devi ce model;
  - Flow-based paradigm builds a concrete basi s for most common security capabilities.

Start from a limited set of NSFs (do not boil the ocean), and be patient for its self-evolvement!

# Information Model for I2NSF Capability Interface

**- Match values based on packet data**

Packet header - Can be standardized

Packet payload - Provided by NSF capabilities

**– Match values based on context**

Ex.: user, State, time, geo-location, etc.

Many can (and should) be standardized, but many also from NSF capabilities

**– Egress processing**

Invoke signaling

Packet forwarding and/or transformation

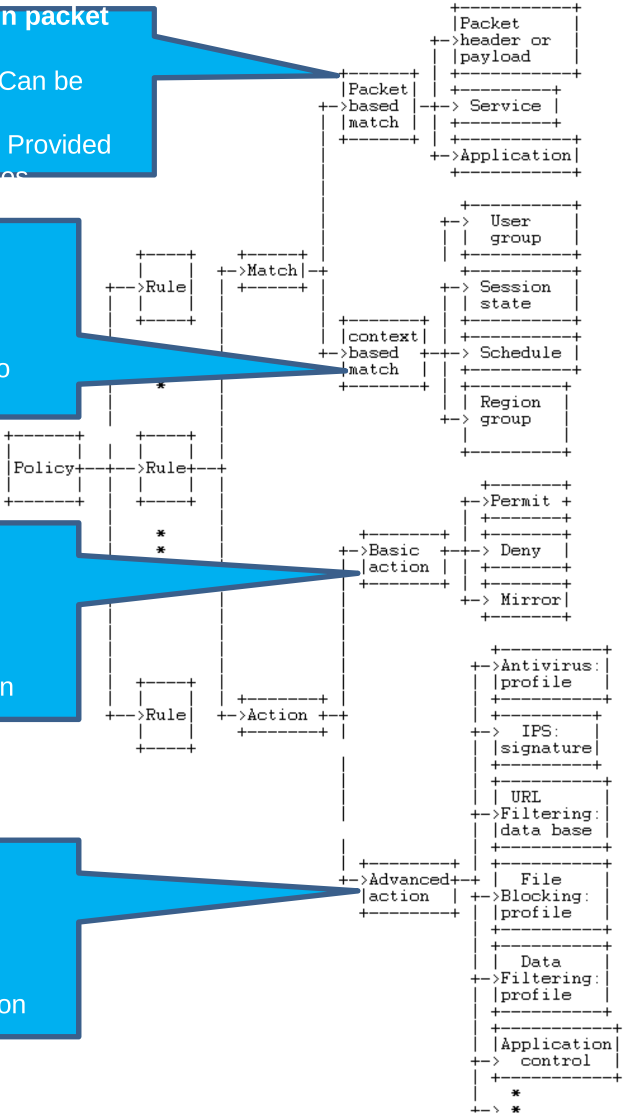
Possibility for SDN/NFV integration

**- Vendor Unique innovation , Vendor specific**

e.g. IPS:<Profile>

Profile: signature, Anti-virus, URL filtering, etc.

Integrated and one-pass checks on the content of packets



Key goal:

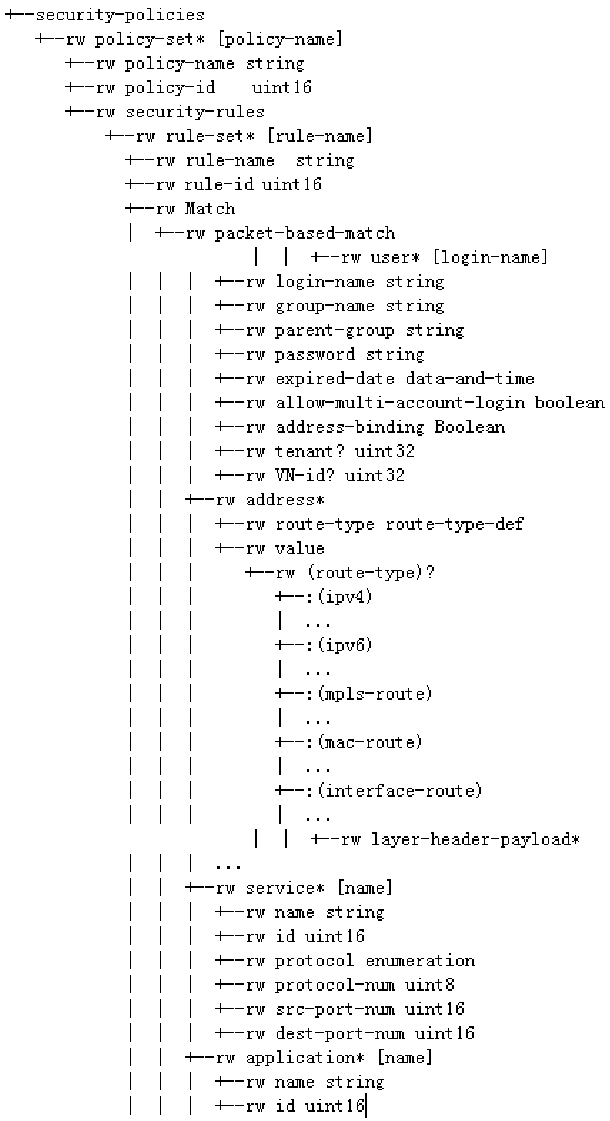
- Flexible and comprehensive semantics;
- extensible IM for containing different vendors' security capabilities, in essence, respective difference or innovation.

# Information Model Grammar Details

```
<Policy> ::= <policy-name> <policy-id> (<Rule> ...)
<Rule> ::= <rule-name> <rule-id> <Match> <Action>
<Match> ::= [<packet-based-match>] [<context-based-match>]

<packet-based-match> ::= [<packet-header-payload> ...] [<service> ...]
                         [<application> ...]
<packet-header-payload> ::= [<address-scope>] [<layer-2-header>]
                            [<layer-3-header>] [<layer-4-header>] [<payload>]
<address-scope> ::= <route-type> (<ipv4-route> | <ipv6-route> | <mpls-route> |
                    <mac-route> | <interface-route>)
<route-type> ::= <IPV4> | <IPV6> | <MPLS> | <IEEE_MAC> | <INTERFACE>
<ipv4-route> ::= <ip-route-type> (<destination-ipv4-address> |
                 <source-ipv4-address> |  (<destination-ipv4-address>
                 <source-ipv4-address>))
<destination-ipv4-address> ::= <ipv4-prefix>
<source-ipv4-address> ::= <ipv4-prefix>
<ipv4-prefix> ::= <IPV4_ADDRESS> <IPV4_PREFIX_LENGTH>

<ipv6-route> ::= <ip-route-type> (<destination-ipv6-address> |
                 <source-ipv6-address> |  (<destination-ipv6-address>
                 <source-ipv6-address>))
<destination-ipv6-address> ::= <ipv6-prefix>
<source-ipv6-address> ::= <ipv6-prefix>
<ipv6-prefix> ::= <IPV6_ADDRESS> <IPV6_PREFIX_LENGTH>
<ip-route-type> ::= <SRC> | <DEST> | <DEST_SRC>
<layer-3-header> ::= <ipv4-header> | <ipv6-header>
<ipv4-header> ::= <SOURCE_IPv4_ADDRESS> <DESTINATION_IPv4_ADDRESS>
                  <PROTOCOL>  [<TTL>] [<DSCP>]
<ipv6-header> ::= <SOURCE_IPV6_ADDRESS> <DESTINATION_IPV6_ADDRESS>
                  <NEXT_HEADER> [<TRAFFIC_CLASS>] [<FLOW_LABEL>]
```

```
<service> ::= <name> <id> <protocol> [<protocol-num>] [<src-port>] [<dest-port>]
<protocol> ::= <TCP> | <UDP> | <ICMP> | <ICMPv6> | <IP>
<application> ::= <name> <id> <category> <subcategory>
                  <data-transmission-model>  <risk-level> <signature>
<category> ::= <business-system> | <Entertainment> | <internet> | <network> |
               <general>
<subcategory> ::= <Finance> | <Email> | <Game> | <media-sharing> |
                  <social-network> |  <web-posting> | <proxy> | ...
<data-transmission-model> ::= <client-server> | <browser-based> |<networking> |
                              <peer-to-peer> | <unassigned>
<risk-level> ::= <Exploitable> | <Productivity-loss> | <Evasive> | <Data-loss> |
                 <Malware-vehicle> |<Bandwidth-consuming> | <Tunneling>
<signature> ::= <server-address> <protocol> <dest-port-num> <flow-direction>
                <object>  <keyword>
<flow-direction> ::= <request> | <response> | <bidirection>
<object> ::= <packet> | <flow>

<context based match> ::= [<user-group> ...] [<session-state>] [<schedule>]
                          [<region-group>]
<user-group> ::= <user>...
<user> ::= (<login-name> <group-name> <parent-group> <password>
           <expired-date>  <allow-multi-account-login> <address-binding>) |
           <tenant> | <VN-id>
<session-state> ::= <new> | <established> | <related> | <invalid> |  <untracked>
<schedule> ::= <name> <type> <start-time> <end-time> <weekly-validity-time>
<type> ::= <once> | <periodic>

<action> ::= <basic-action> [<advanced-action>]
<basic-action> ::= <pass> | <deny> | <mirror> | <call-function> | <encapsulation>
<advanced-action> ::= [<profile-antivirus>] [<profile-IPS>] [<profile-url-filtering>]
```

# Yang Data Model Specification



```
+--security-policies
   +--rw policy-set* [policy-name]
      +--rw policy-name string
      +--rw policy-id    uint16
      +--rw security-rules
          +--rw rule-set* [rule-name]
            +--rw rule-name  string
            +--rw rule-id uint16
            +--rw Match
            |  +--rw packet-based-match
                         |  |  +--rw user* [login-name]
            |  |  |  +--rw login-name string
            |  |  |  +--rw group-name string
            |  |  |  +--rw parent-group string
            |  |  |  +--rw password string
            |  |  |  +--rw expired-date data-and-time
            |  |  |  +--rw allow-multi-account-login boolean
            |  |  |  +--rw address-binding Boolean
            |  |  |  +--rw tenant? uint32
            |  |  |  +--rw VN-id? uint32
            |  |  +--rw address*
            |  |  |  +--rw route-type route-type-def
            |  |  |  +--rw value
            |  |  |     +--rw (route-type)?
            |  |  |        +--:(ipv4)
            |  |  |        |  ...
            |  |  |        +--:(ipv6)
            |  |  |        |  ...
            |  |  |        +--:(mpls-route)
            |  |  |        |  ...
            |  |  |        +--:(mac-route)
            |  |  |        |  ...
            |  |  |        +--:(interface-route)
            |  |  |        |  ...
                         |  |  +--rw layer-header-payload*
            |  |  |  ...
            |  |  +--rw service* [name]
            |  |  |  +--rw name string
            |  |  |  +--rw id uint16
            |  |  |  +--rw protocol enumeration
            |  |  |  +--rw protocol-num uint8
            |  |  |  +--rw src-port-num uint16
            |  |  |  +--rw dest-port-num uint16
            |  |  +--rw application* [name]
            |  |  |  +--rw name string
            |  |  |  +--rw id uint16
```

# Next Step

- Solicit Comments

- Maybe remove the Yang data model part

- Keep on improvement, including: more content about security profiles, improving information model structure and grammar, examples, etc

# Thanks!

Dacheng Zhang
Liang Xia (Frank)