Telecom Cloud Side Meeting at IETF 120

ICSS: Intent-Based Cloud Security System with Interface to Network Security Functions (I2NSF)

Jaehoon (Paul) Jeong

Computer Science and Engineering at Sungkyunkwan University

Email: pauljeong@skku.edu

Contents

- Introduction
- Interface to Network Security Functions (I2NSF)
- Information and Data Models of I2NSF
- Security Policy Translator in I2NSF
- Closed-Loop Security Control in I2NSF
- Conclusion

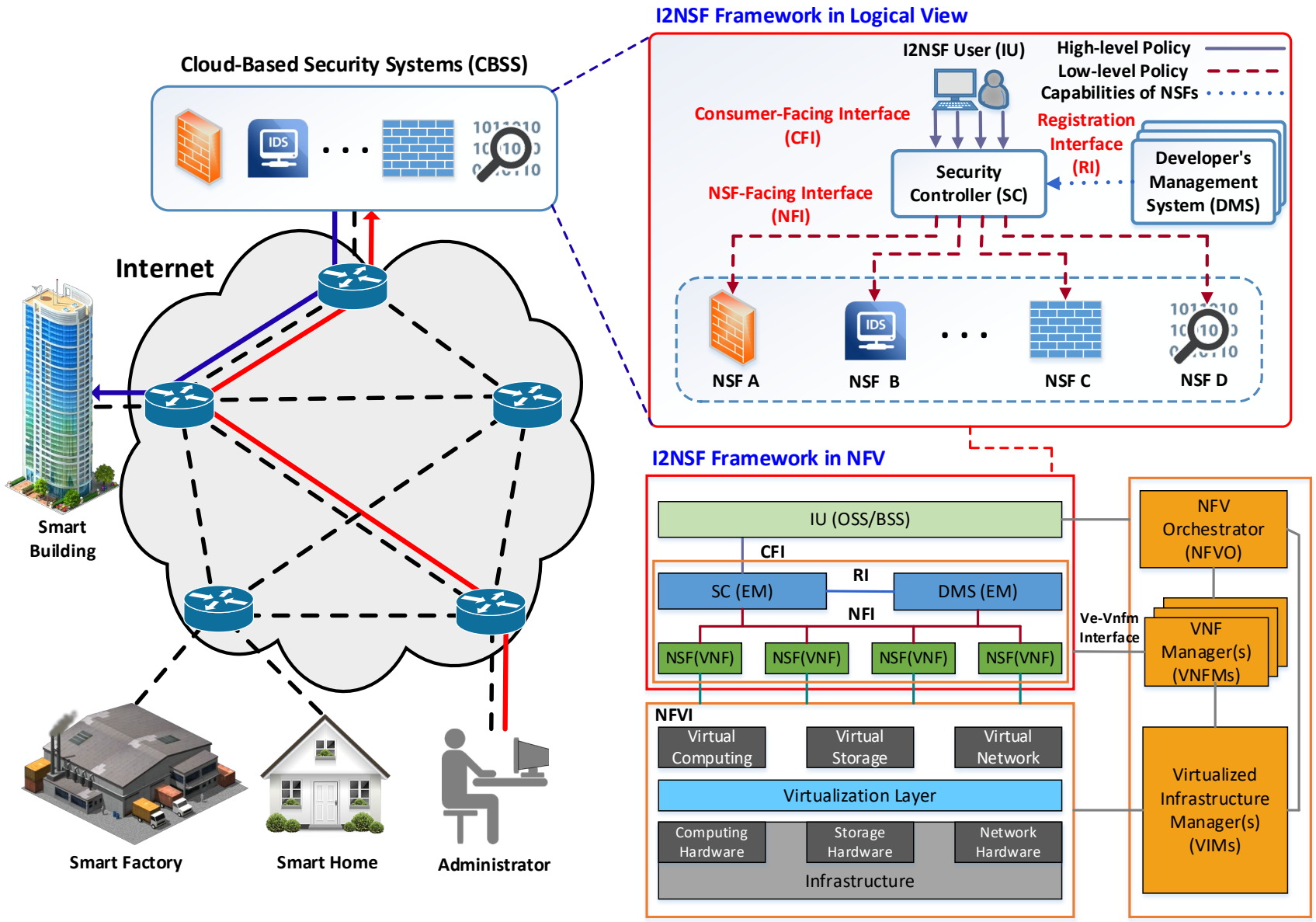
Introduction

Introduction to Interface to Network Security Functions (I2NSF)

- Background
  - Flexible and Scalable Provisioning of Security Services in Network Functions Virtualization (NFV)
  - Efficient Leverage of Physical Network Functions and Virtual Network Functions.
  - Efficient Security Task Processing in NFV along with Software-Defined Networking (SDN).

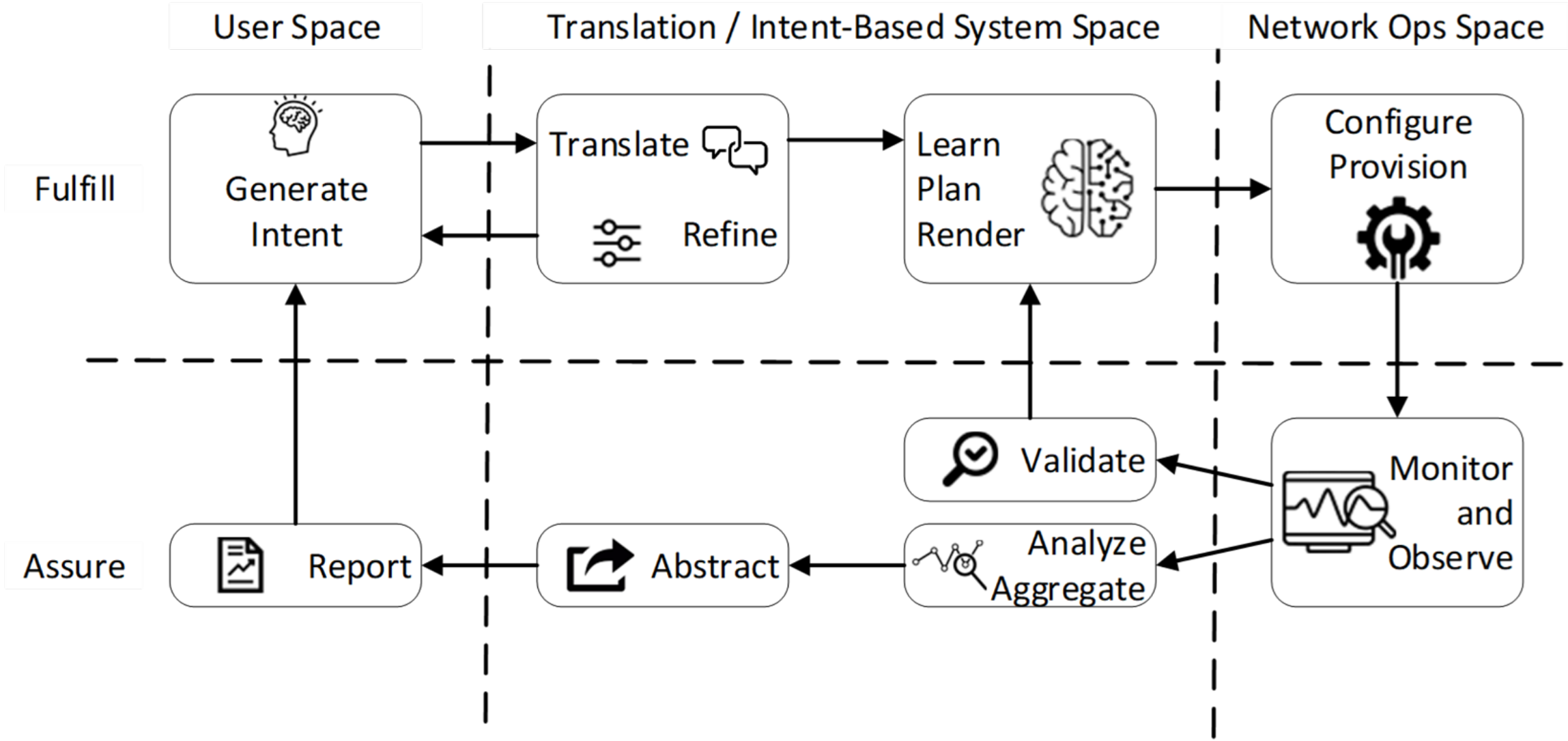
- Motivation
  - IETF I2NSF Working Group aims to make standard Interfaces to Network Security Functions (I2NSF) for cloud-based security system.
  - I2NSF WG defines a Framework and its Interfaces to control and manage Network Security Functions (NSFs) in SDN/NFV environments.
  - I2NSF WG Website: https://datatracker.ietf.org/wg/i2nsf/about/
  - I2NSF WG Documents: https://datatracker.ietf.org/wg/i2nsf/documents/

Intent-Based Cloud Security System (ICSS) with I2NSF



[Source] "IBCS: Intent-Based Cloud Services for Security Applications", IEEE Communications Magazine, Vol. 58, No. 4, pp. 45-51, April 2020. http://iotlab.skku.edu/publications/international-journal/IBCS-Communications-Magazine-2020.pdf

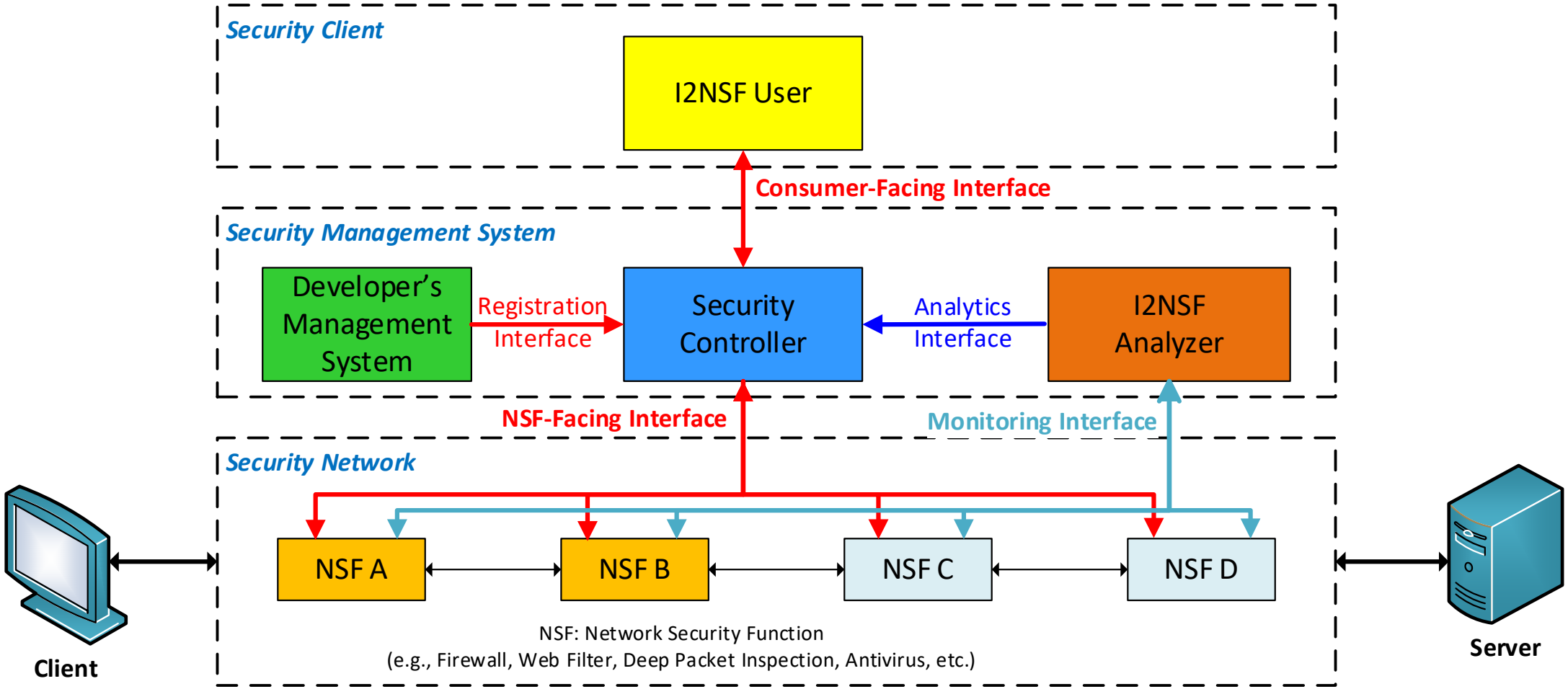
Intent-Based Networking: Concepts & Definitions [RFC9315]



[Source] https://datatracker.ietf.org/doc/html/rfc9315

Interface to Network Security Functions (I2NSF)

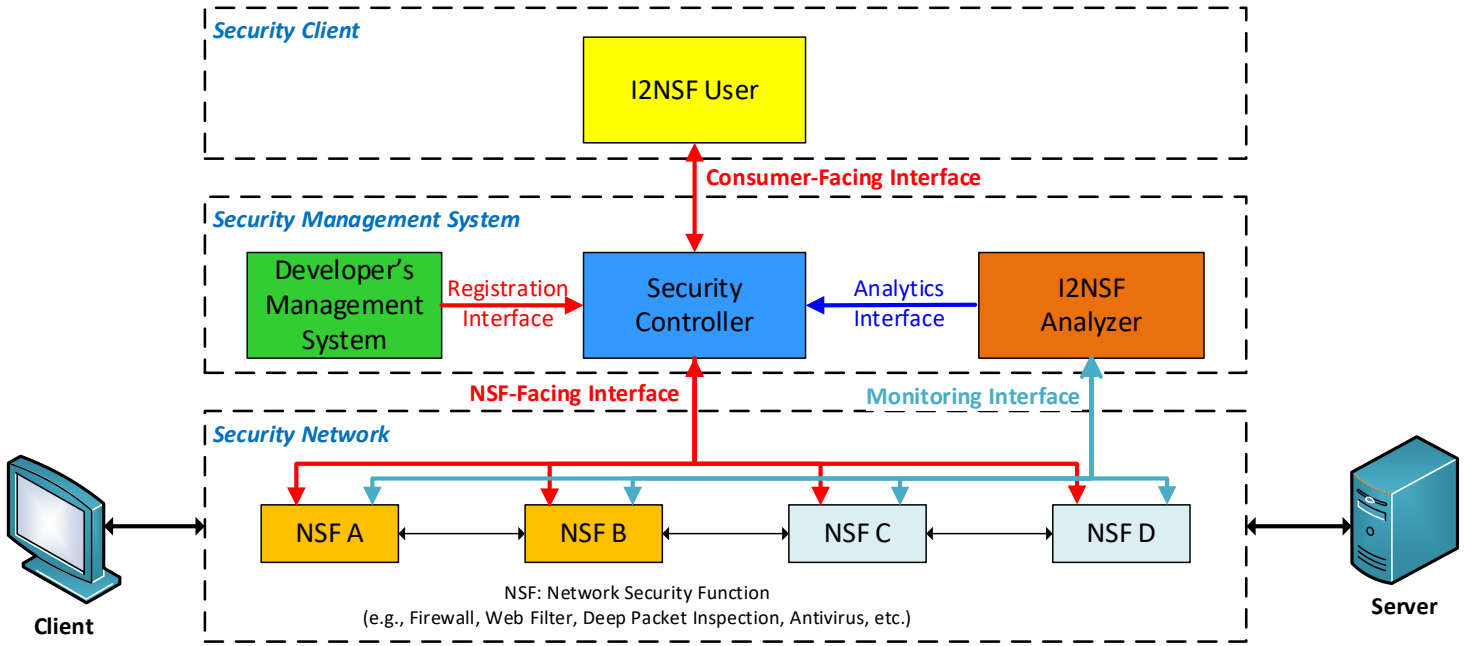
I2NSF Framework with Interfaces [RFC8329]



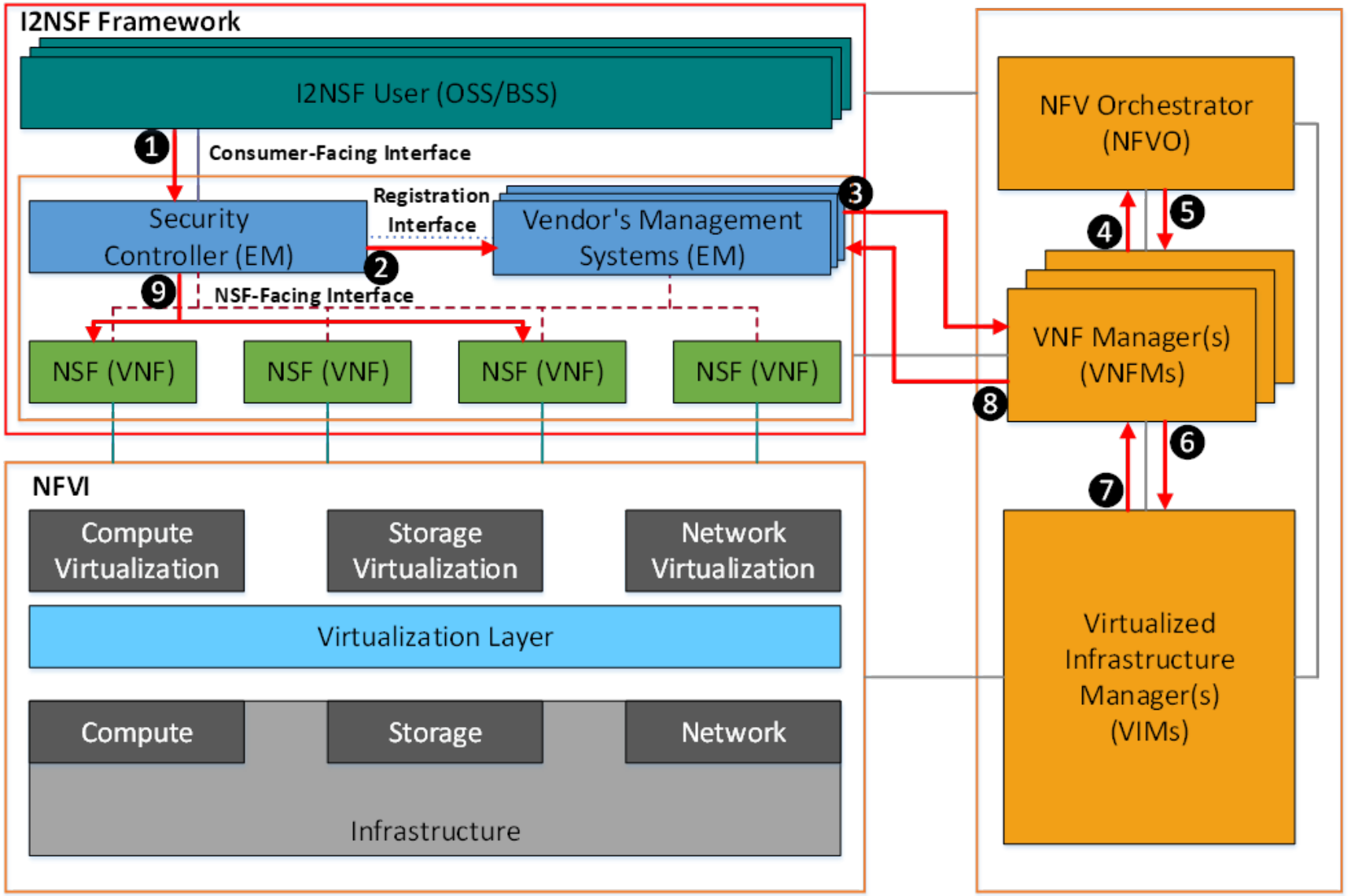
[Source] https://datatracker.ietf.org/doc/html/rfc8329

I2NSF Interfaces

- Registration Interface
  - Developer's Management System (DMS) registers an NSF with Security Controller.
- Consumer-Facing Interface
  - I2NSF User delivers a high-level security policy to Security Controller.
- NSF-Facing Interface
  - Security Controller delivers a low-level security polity to an NSF.
- Monitoring Interface
  - An NSF delivers its monitoring data to I2NSF Analyzer.
- Analytics Interface
  - I2NSF Analyzer delivers its feedback to Security Controller for policy update.



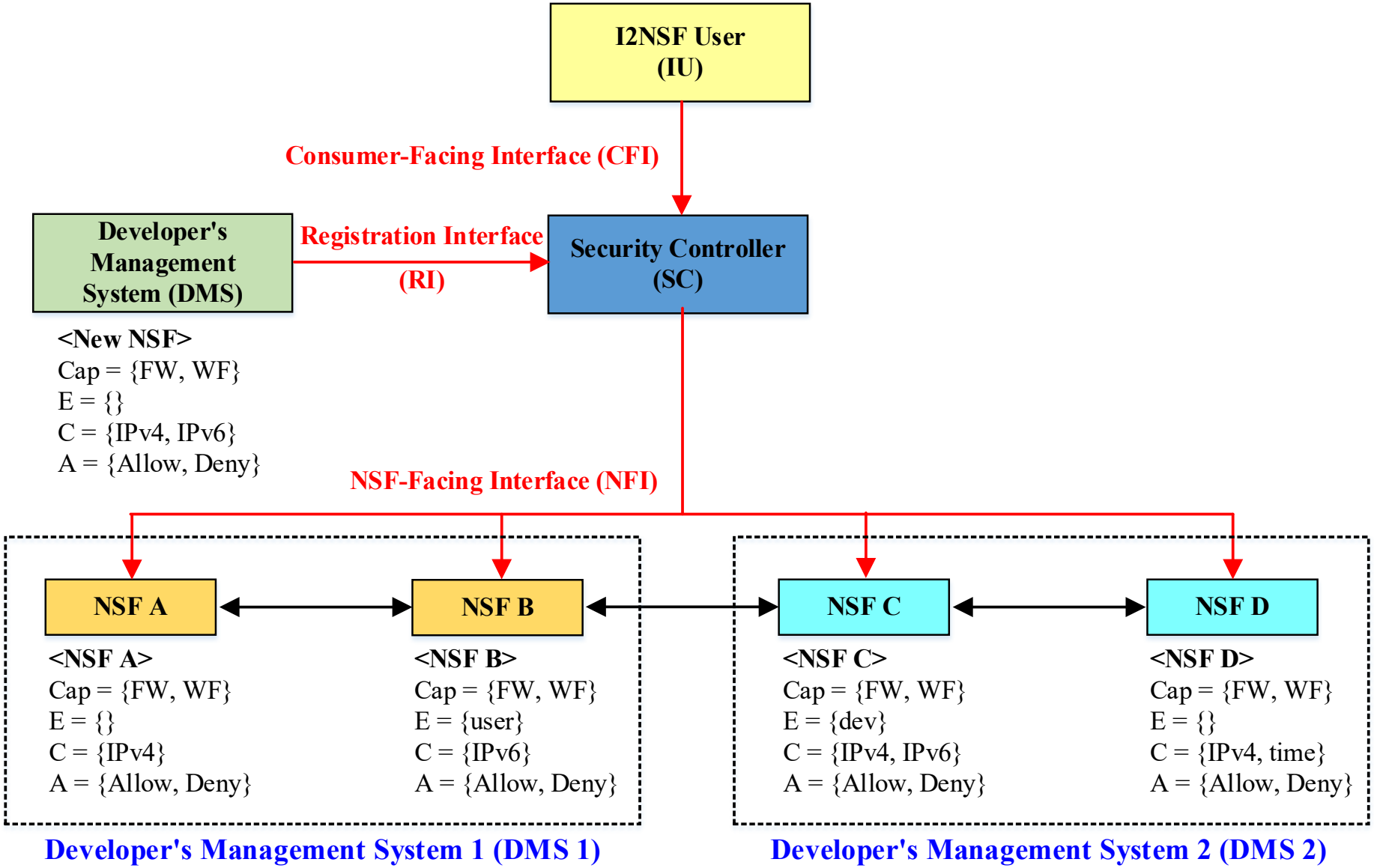
I2NSF Framework with NFV



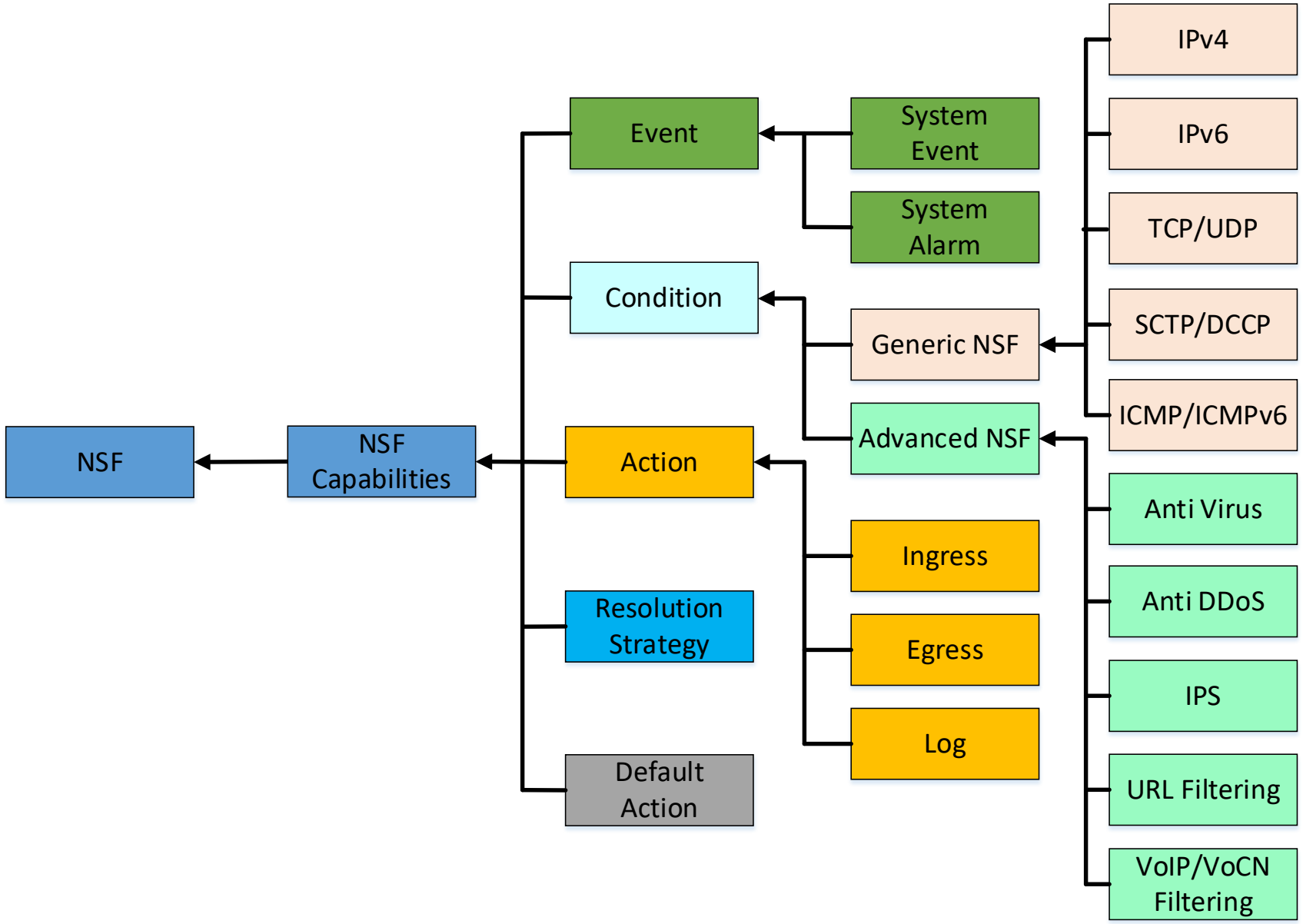
Information and Data Models of I2NSF

[Source] I2NSF YANG Data Models
https://datatracker.ietf.org/wg/i2nsf/documents/

Registration of NSF's Capability with Security Controller



Information Model of Registration Interface



YANG Data Model: YANG Tree of NSF Capability

- NSF Capability
  - It describes the capability of an NSF in terms of an Event-Condition-Action (ECA) policy.
  - Capabilities
    - Directional Capabilities
    - Event Capabilities
    - Condition Capabilities
    - Action Capabilities
    - Resolution-Strategy Capabilities
    - Default-Action Capabilities

- I2NSF Capability YANG Data Model
  - https://datatracker.ietf.org/doc/html/draft-ietf-i2nsf-capability-data-model-32

```
module: ietf-i2nsf-capability
 +--rw nsf* [nsf-name]
  +--rw nsf-name                string
  +--rw directional-capabilities*      identityref
  +--rw event-capabilities
  | +--rw system-event-capability*  identityref
  | +--rw system-alarm-capability*  identityref
  +--rw condition-capabilities
  | +--rw generic-nsf-capabilities
  | | +--rw ethernet-capability*  identityref
  | | +--rw ipv4-capability*    identityref
  | | +--rw ipv6-capability*    identityref
  | | +--rw icmpv4-capability*   identityref
  | | +--rw icmpv6-capability*   identityref
  | | +--rw tcp-capability*    identityref
  | | +--rw udp-capability*    identityref
  | | +--rw sctp-capability*    identityref
  | | +--rw dccp-capability*    identityref
  | +--rw advanced-nsf-capabilities
  | | +--rw anti-ddos-capability*       identityref
  | | +--rw ips-capability*           identityref
  | | +--rw anti-virus-capability*       identityref
  | | +--rw url-filtering-capability*     identityref
  | | +--rw voip-vocn-filtering-capability*  identityref
  | +--rw context-capabilities
  |   +--rw time-capabilities*         identityref
  |   +--rw application-filter-capabilities*  identityref
  |   +--rw device-type-capabilities*     identityref
  |   +--rw user-condition-capabilities*    identityref
  |   +--rw geographic-capabilities*      identityref
  +--rw action-capabilities
  | +--rw ingress-action-capability*  identityref
  | +--rw egress-action-capability*  identityref
  | +--rw log-action-capability*    identityref
  +--rw resolution-strategy-capabilities*  identityref
  +--rw default-action-capabilities*    identityref
```

Security Policy Translator in I2NSF

[Source] "SPT: Security Policy Translator for Network Security Functions in Cloud-Based Security Services", IEEE Transactions on Dependable and Secure Computing, February 2024.
http://iotlab.skku.edu/publications/international-journal/IEEE-TDSC-SPT-2024.pdf

Security Policy Translator

- Security Policy Translation in Interface to Network Security Functions
  - Objective
    - It aims at the specification of a scheme of security policy translation (i.e., Security Policy Translator) in Interface to Network Security Functions (I2NSF) Framework.
    - This security policy translator enables an intent-based security service in the I2NSF system.
  - Document Title
    - Guidelines for Security Policy Translation in Interface to Network Security Functions
  - Document link
    - https://datatracker.ietf.org/doc/draft-yang-i2nsf-security-policy-translation/
  - Document status
    - Individual draft

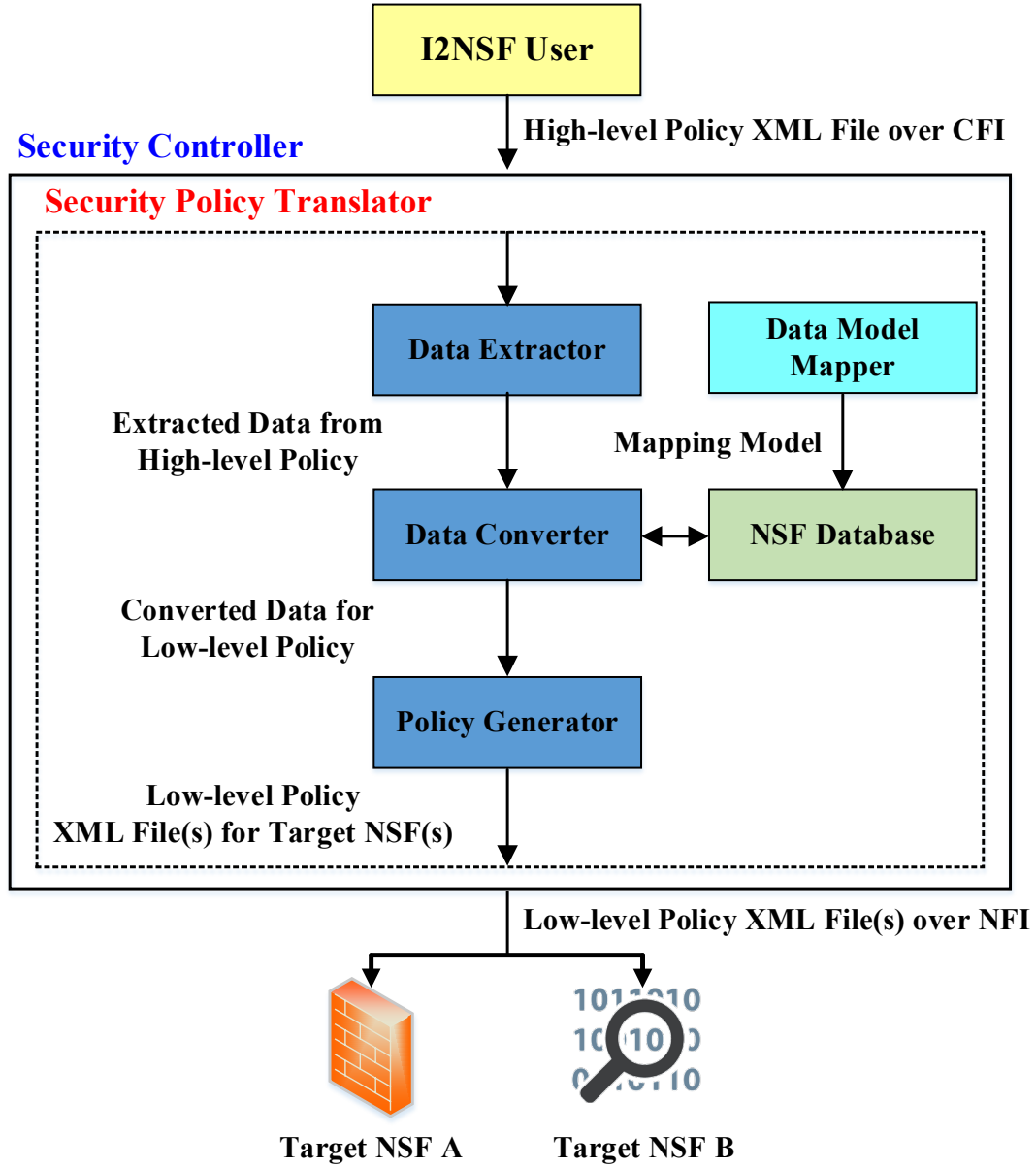
Necessity for Security Policy Translator

- Policy Representation according to Users
  - A high-level policy is for I2NSF Users, and a low-level policy is for NSFs.

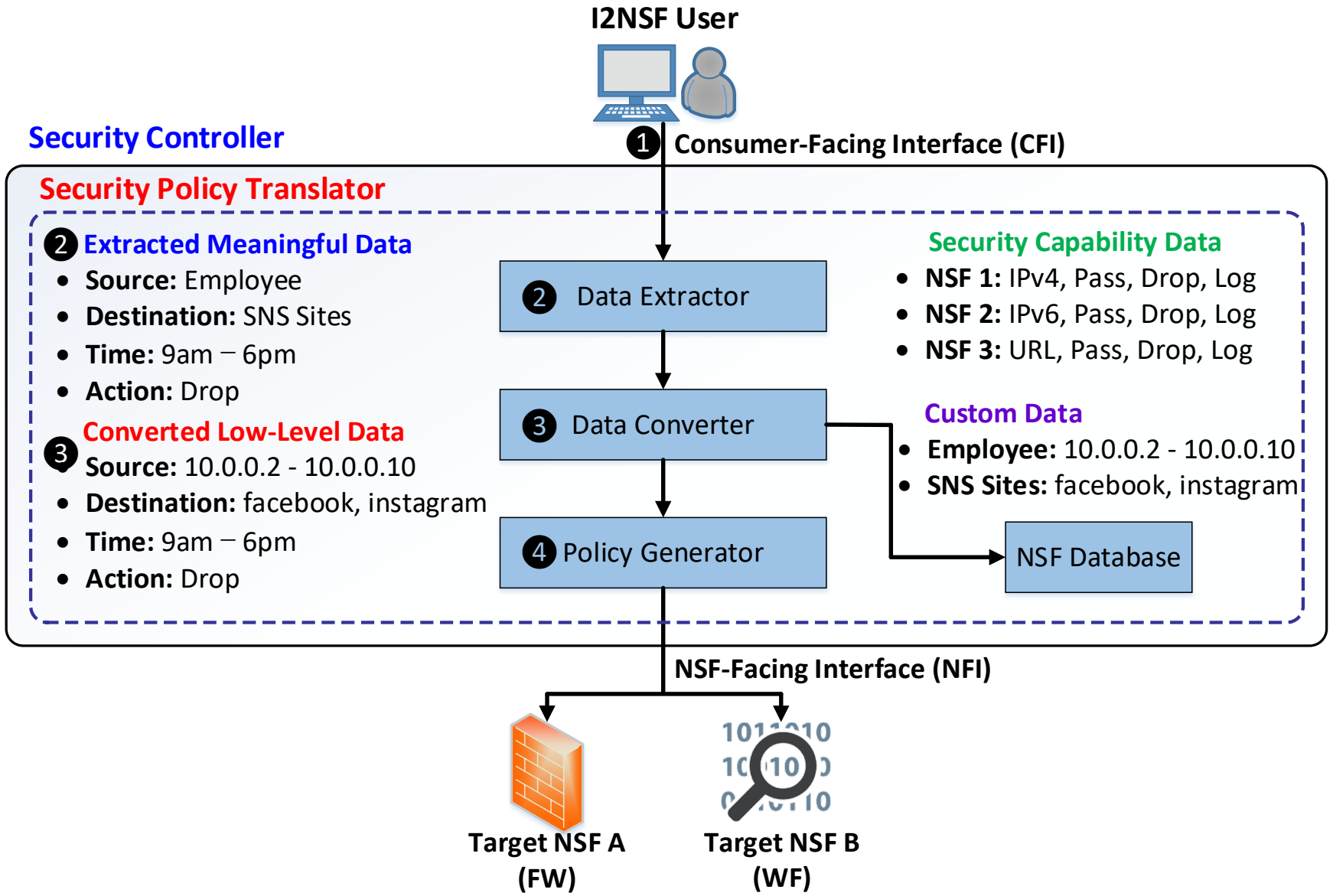
```
o  Block my son's computers from malicious websites.

o  Drop packets from the IP address 10.0.0.1 and 10.0.0.3 to harm.com
   and illegal.com
```

  - Even if I2NSF User gives the first high-level policy, I2NSF System needs to automatically translate it into the second low-level policy.

Architecture of Security Policy Translator (SPT)



Example of Security Policy Translator (SPT)



IETF-114 I2NSF Hackathon Project

I2NSF (Interface to Network Security Functions) Framework Project

Champion: Jaehoon (Paul) Jeong

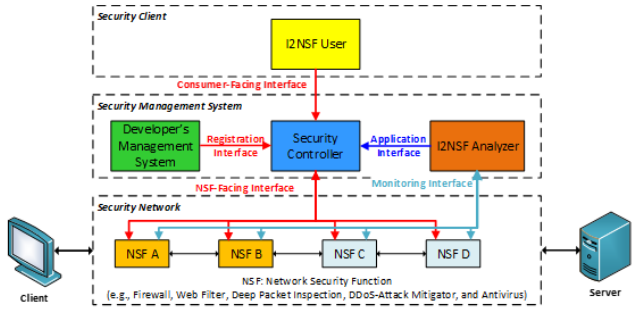
I2NSF Hackathon Project

Professors:
- Jaehoon (Paul) Jeong (SKKU)
- Younghan Kim (SSU)

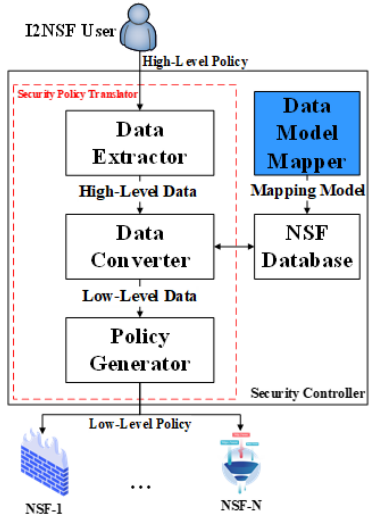
Researchers:
- Jung-Soo Park (ETRI)
- Yunchul Choi (ETRI)
- Jinyong Kim (SKKU)

Students:
- Patrick Lingga (SKKU)
- Jeonghyeon Kim (SKKU)
- Hadong Park (Calvin University)

I2NSF Framework



I2NSF Security Policy Translator



Where to get Code and Demo Video Clip
- Github – Source Code
  - ✓ https://github.com/jaehoonpaul/i2nsf-framework

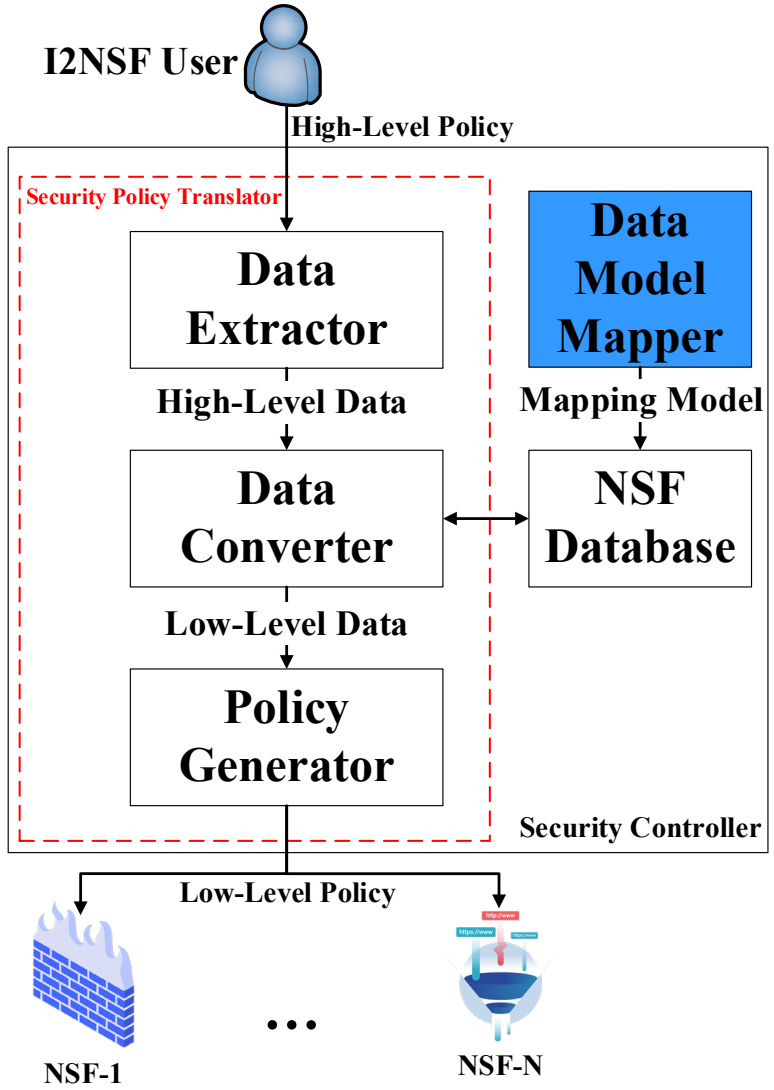
What to pull down to set up an environment
- OS: Ubuntu 16.04 LTS
- ConfD for NETCONF: 6.6 Version
- Jetconf for RESTCONF
- OpenStack: Queens version
- NSF: Suricata
- Hyperledger Fabric: 2.2 version

Manual for Operation Process
- I2NSF-Manual-Hackathon.md contains detailed description about operation process. It can be found in the GitHub.

Contents of Implementation
- Cloud-based Security Service System using I2NSF Framework
  - ✓ Web-based I2NSF User
  - ✓ Console-based Security Controller
  - ✓ Console-based Developer's Management System
  - ✓ I2NSF Framework in OpenStack NFV Environment
  - ✓ I2NSF Capability YANG Data Model
  - ✓ Registration Interface via NETCONF/YANG
  - ✓ Consumer-Facing Interface via RESTCONF/YANG
  - ✓ NSF-Facing Interface via NETCONF/YANG
  - ✓ Monitoring Interface via NETCONF/YANG
  - ✓ Web-based NSF Monitoring
  - ✓ Application Interface as Feedback from I2NSF Analyzer
- Network Security Functions
  - ✓ Firewall and Web-filter using Suricata
- Advanced Functions
  - ✓ Security Policy Translation: Automatic Generation of Low-Level Policy with Policy Provisioning
  - ✓ Blockchain-based Auditing for I2NSF Policy and Data Transactions

I2NSF Hackathon Project Plan

- Implementation of Security Policy Translator



The overall architecture of our scheme consists of five components:

✓Data Extractor.
✓Data Converter.
✓NSF Database.
✓Policy Generator.
✓Data Model Mapper.

What got done (1/4)

- Data Model Mapper Results:

Consumer-Facing Interface's YANG Data Model Attributes

NSF-Facing Interface's YANG Data Model Attributes

```
mysql> select * from attributes;
+-------+-----------------------------------------------------+-------+-----------------------------------------------------------------------+
| cfiID | cfiPath                                             | nfiID | nfiPath                                                               |
+-------+-----------------------------------------------------+-------+-----------------------------------------------------------------------+
|     1 | /i2nsf-cfi-policy/name                              |     1 | /i2nsf-security-policy/name                                           |
|     2 | /i2nsf-cfi-policy/language                          |     2 | /i2nsf-security-policy/language                                       |
|     3 | /i2nsf-cfi-policy/resolution-strategy               |     4 | /i2nsf-security-policy/resolution-strategy                            |
|     5 | /i2nsf-cfi-policy/rules/name                        |     7 | /i2nsf-security-policy/rules/name                                     |
|     6 | /i2nsf-cfi-policy/rules/priority                    |     9 | /i2nsf-security-policy/rules/priority                                 |
|     8 | /i2nsf-cfi-policy/rules/event/system-event          |    16 | /i2nsf-security-policy/rules/event/system-event                       |
|     9 | /i2nsf-cfi-policy/rules/event/system-alarm          |    17 | /i2nsf-security-policy/rules/event/system-alarm                       |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |    24 | /i2nsf-security-policy/rules/condition/layer-2/source-mac-address     |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |    49 | /i2nsf-security-policy/rules/condition/ipv4/source-ipv4-network       |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |    51 | /i2nsf-security-policy/rules/condition/ipv4/source-ipv4-range         |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |    71 | /i2nsf-security-policy/rules/condition/ipv6/source-ipv6-network       |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |    73 | /i2nsf-security-policy/rules/condition/ipv6/source-ipv6-range         |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |    81 | /i2nsf-security-policy/rules/condition/tcp/source-port-number         |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |   120 | /i2nsf-security-policy/rules/condition/udp/source-port-number         |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |   152 | /i2nsf-security-policy/rules/condition/sctp/source-port-number        |
|    12 | /i2nsf-cfi-policy/rules/condition/firewall/source   |   185 | /i2nsf-security-policy/rules/condition/dccp/source-port-number        |
```

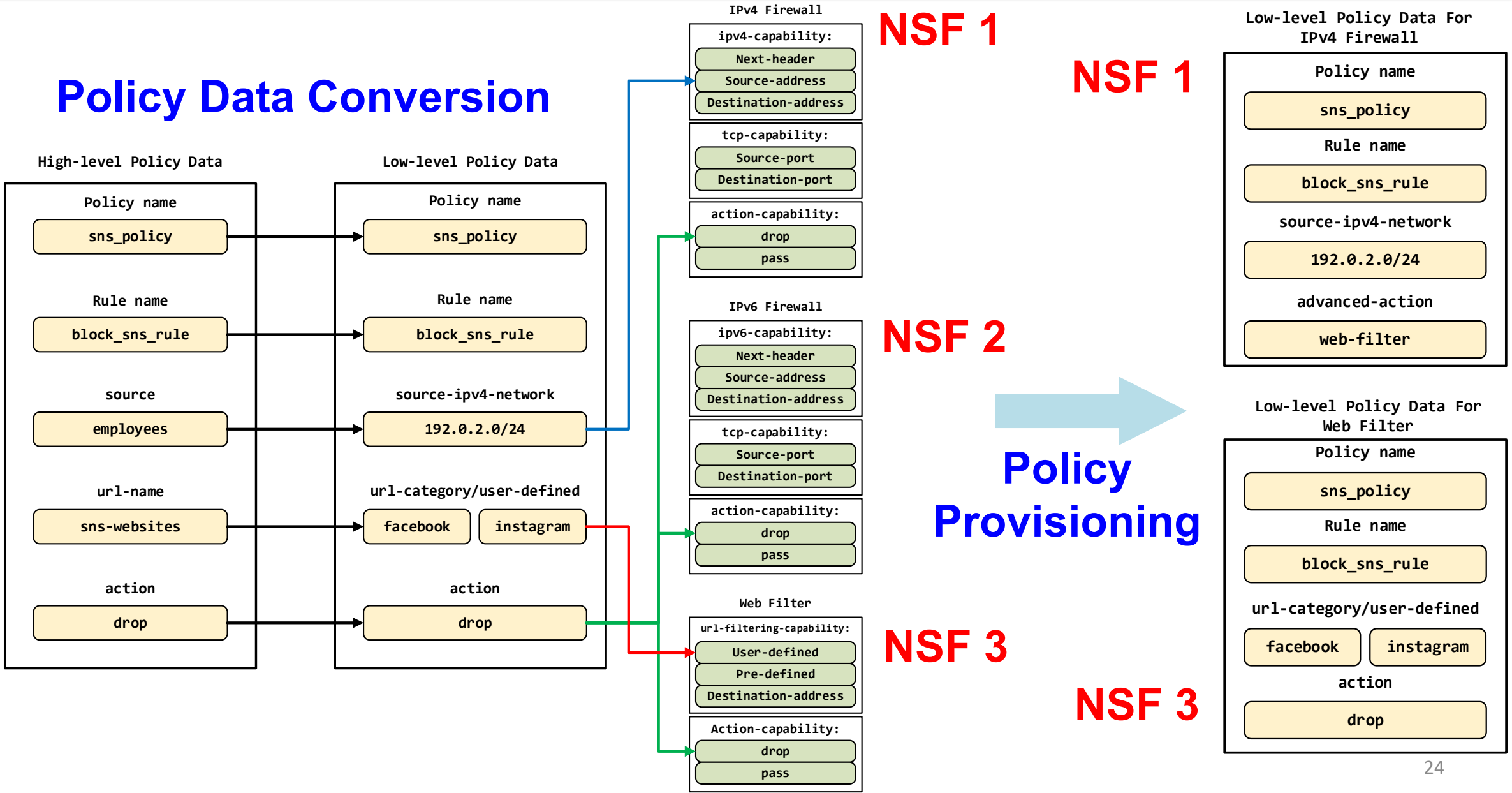
What got done (2/4)

High-level Security Policy

```
<i2nsf-cfi-policy
 xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-cfi-policy">
  <name>security_policy_for_blocking_sns</name>
  <rules>
    <name>block_access_to_sns_during_office_hours</name>
    <condition>
      <firewall>
        <source>employees</source>
      </firewall>
      <url>
        <url-name>sns-websites</url-name>
      </url>
    </condition>
    <actions>
      <primary-action>
        <action>drop</action>
      </primary-action>
    </actions>
  </rules>
</i2nsf-cfi-policy>
```

Extraction of High-Level Information

What got done (3/4): Policy Provisioning



What got done (4/4)

Generated Low-Level Policies

1. Low-Level Policy for Firewall

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-facing-interface">
  <name>sns_access</name>
  <rules>
    <name>block_sns_access_during_operation_time_for_ipv4</name>
    <condition>
      <ipv4>
        <source-ipv4-network>192.0.2.0/24</source-ipv4-network>
      </ipv4>
    </condition>
    <action>
      <advanced-action>
        <content-security-control>
          url-filtering
        </content-security-control>
      </advanced-action>
    </action>
  </rules>
</i2nsf-security-policy>
```

employees translated to an IPv4 network address

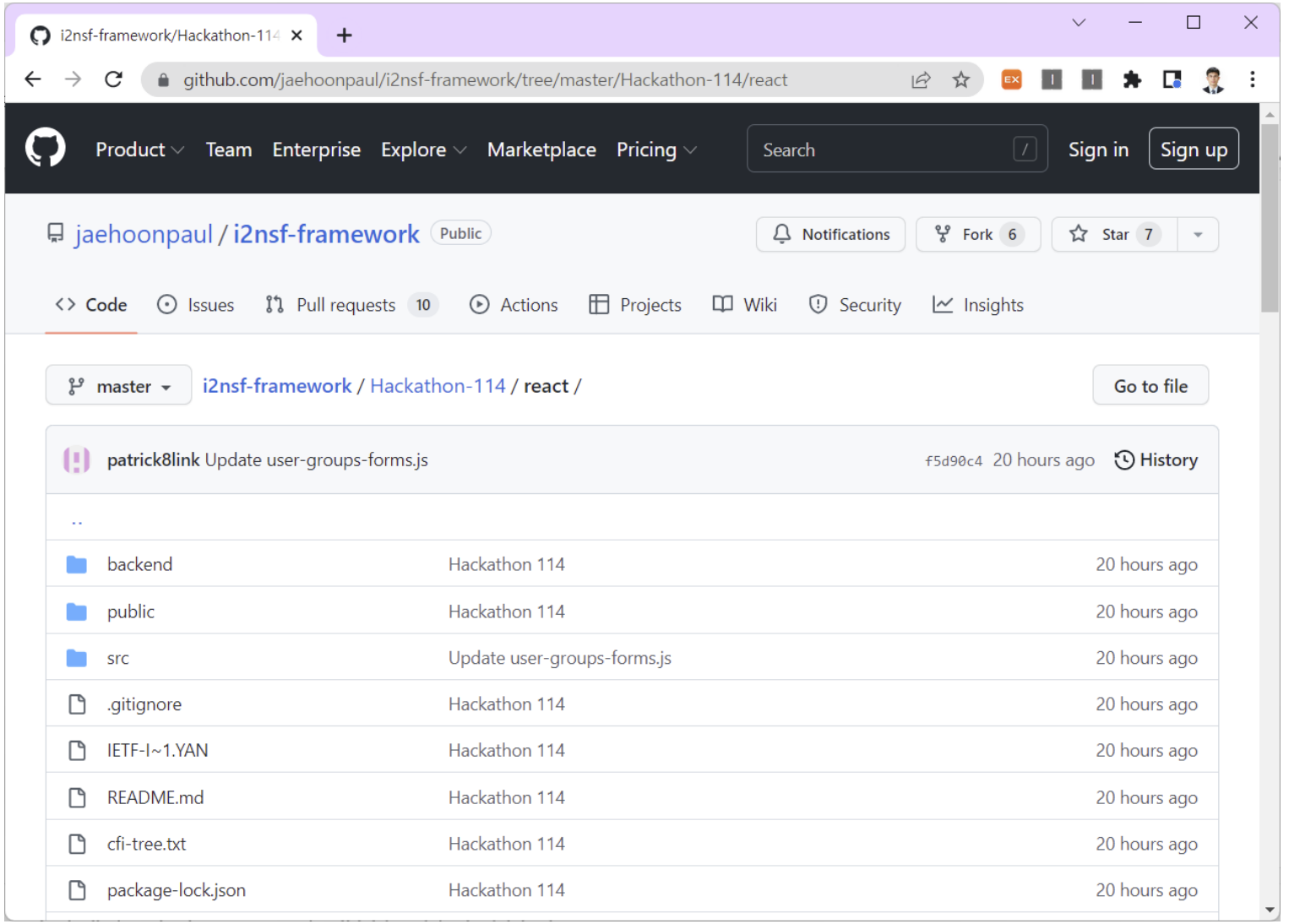
2. Low-Level Policy for Web Filter

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-facing-interface">
  <name>sns_access</name>
  <rules>
    <name>block_sns_access_during_operation_time</name>
    <condition>
      <url-category>
        <user-defined>Facebook</user-defined>
        <user-defined>Instagram</user-defined>
      </url-category>
    </condition>
    <action>
      <packet-action>
        <egress-action>drop</egress-action>
      </packet-action>
    </action>
  </rules>
</i2nsf-security-policy>
```

sns-websites translated into Facebook and Instagram

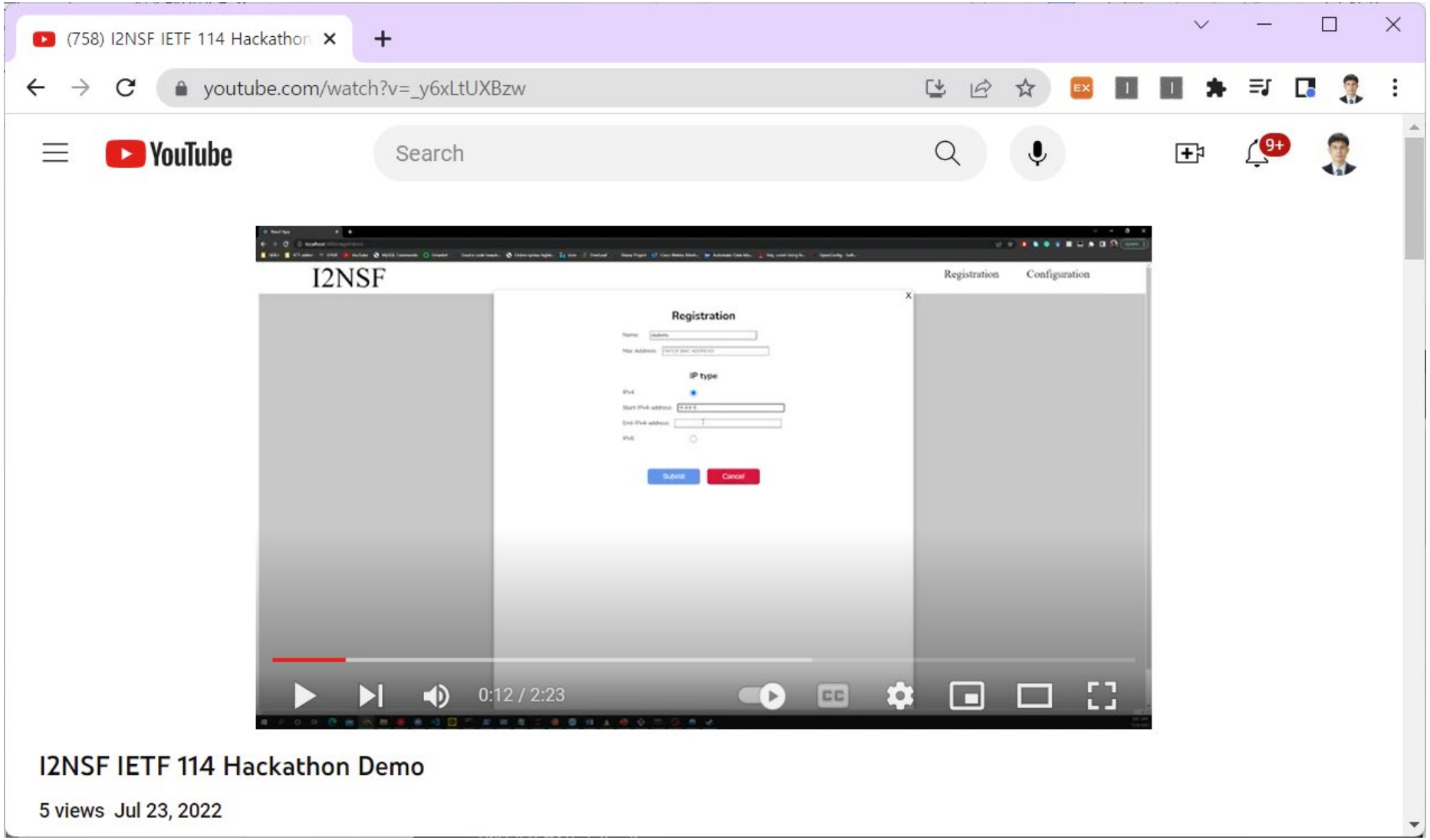
Open-Source Project at GitHub

URL: https://github.com/jaehoonpaul/i2nsf-framework

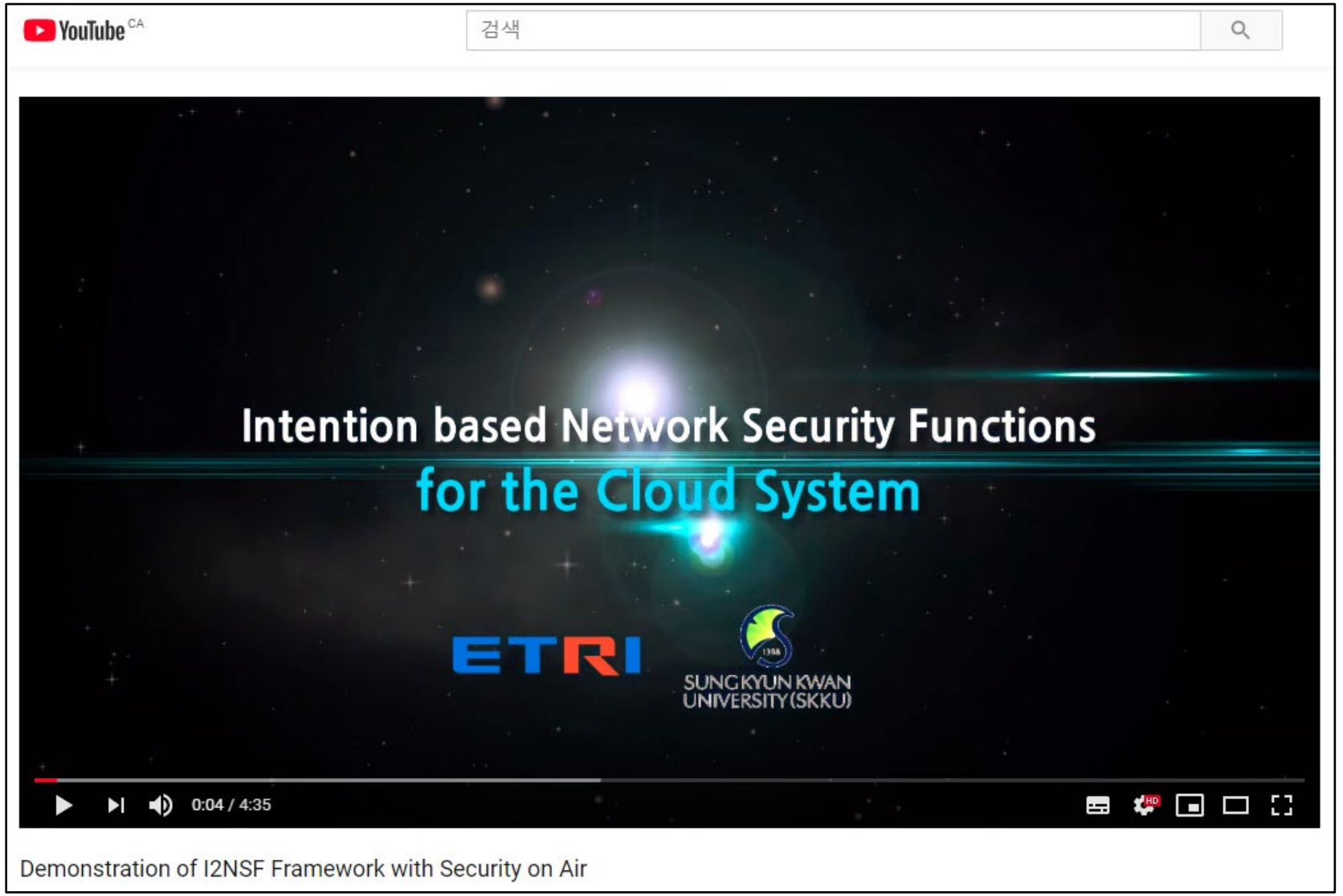


Demonstration Video Clip at YouTube

URL: https://youtu.be/_y6xLtUXBzw



Demonstration for I2NSF Hackathon Project



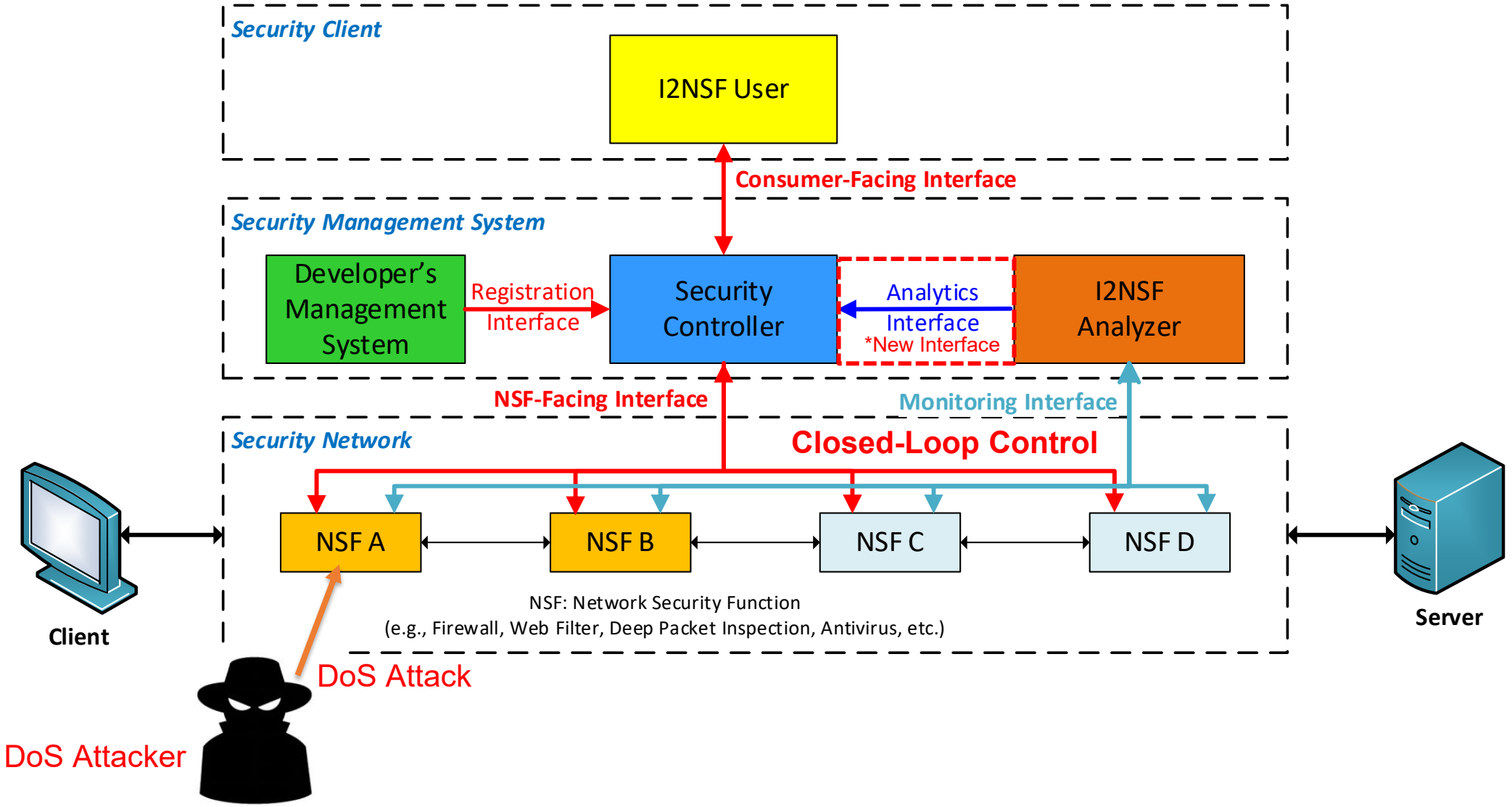
URL: https://www.youtube.com/watch?v=jD4ndqzN0is&t=5s

Closed-Loop Security Control in I2NSF

[Source 1] "Security Management Automation of Cloud-Based Security Services in I2NSF Framework", draft-jeong-i2nsf-security-management-automation-07, February 7, 2024. https://datatracker.ietf.org/doc/draft-jeong-i2nsf-security-management-automation/

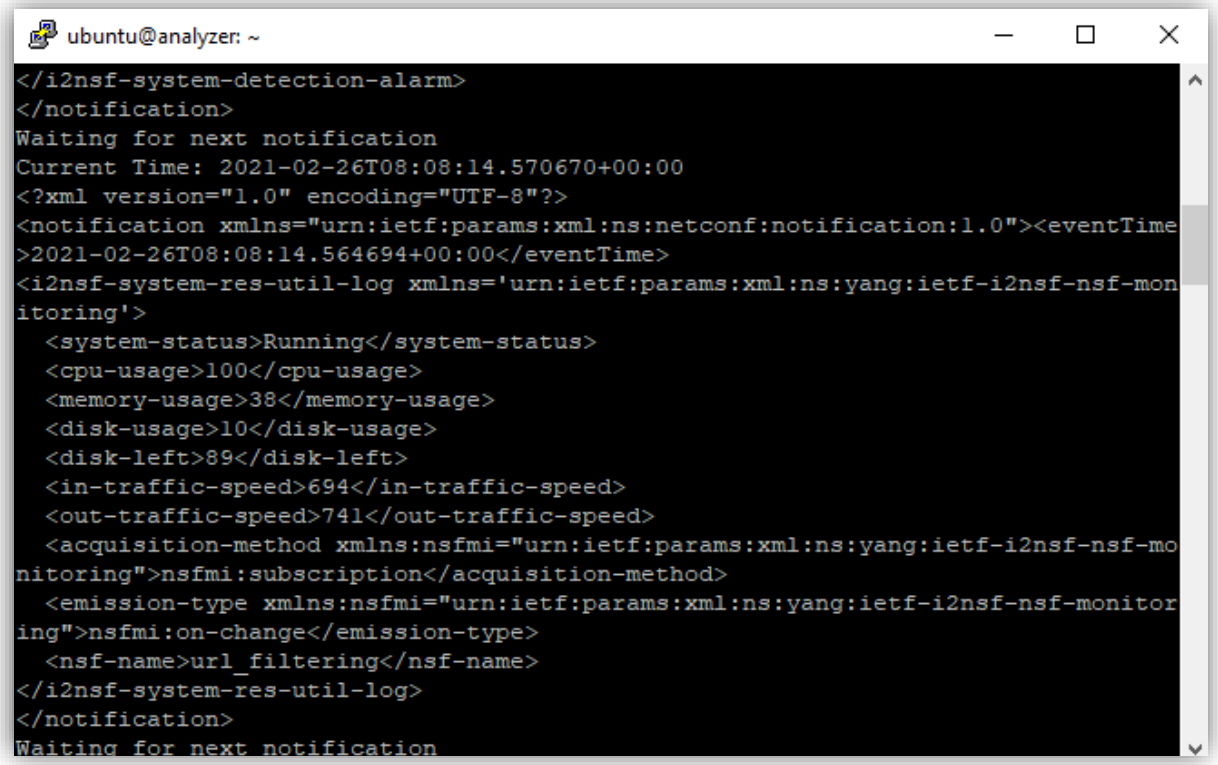
[Source 2] "CBSS: Cloud-Based Security System with Interface to Network Security Functions", ICMU 2023, Kyoto, Japan, November 29, 2023. http://iotlab.skku.edu/publications/international-conference/ICMU2023-CBSS.pdf

Closed-Loop Security Control in I2NSF

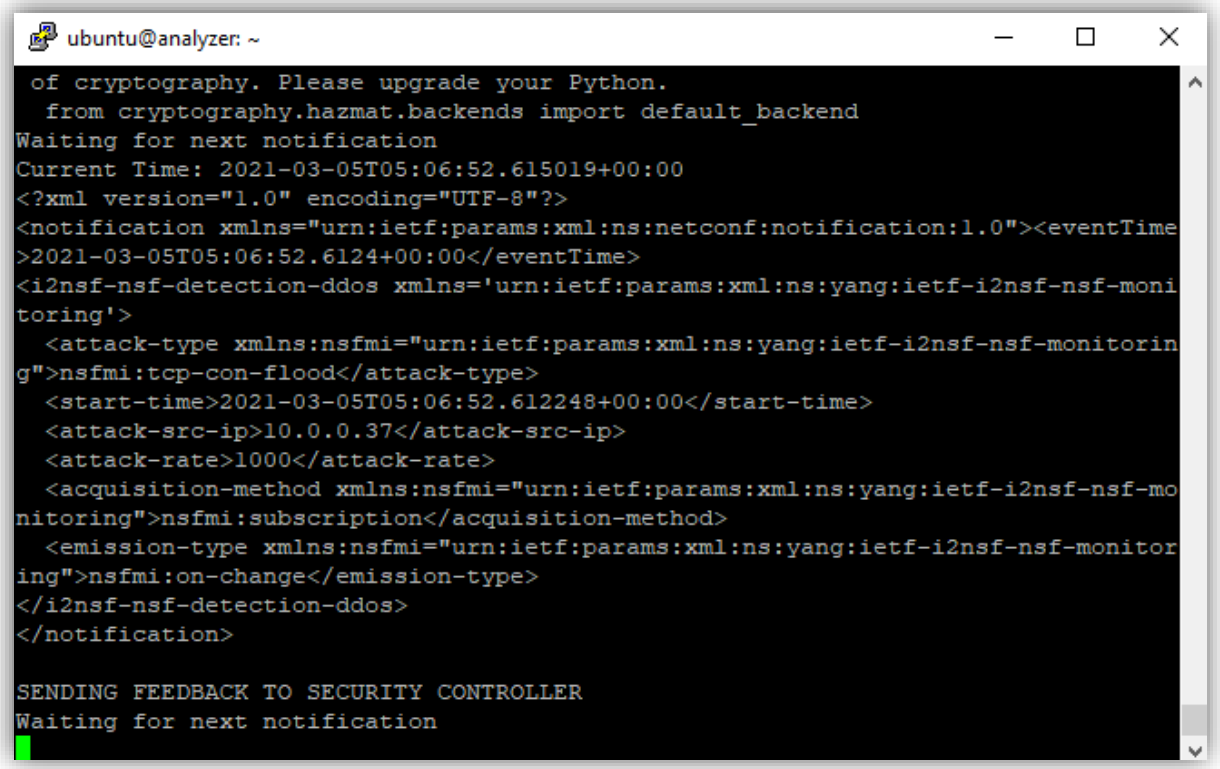


Closed-Loop Security Control (1/3)

- NSF Monitoring using I2NSF Monitoring Interface via NETCONF.
  - ➢ Subscription-based NSF Monitoring.



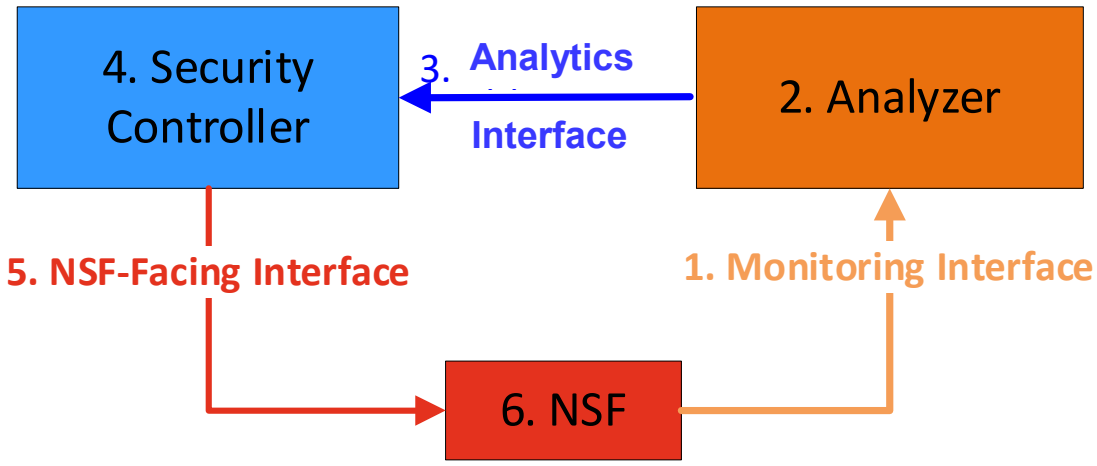
Monitoring NSF's Resources



Monitoring DDoS Detection

Closed-Loop Security Control (2/3)

- Implementation of Application Interface for Feedback delivery to create a closed-loop system of I2NSF Framework.

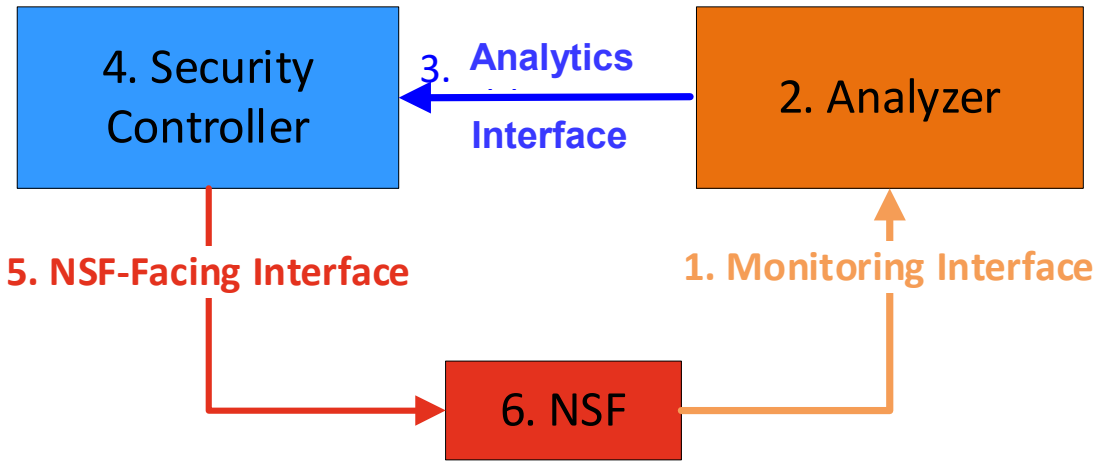


Closed-Loop Security Control

1. **NSF sends monitoring data to Analyzer via Monitoring Interface, such as DoS Detection Report.**
2. **Analyzer creates a new policy based on the received data through machine learning.**
3. **Analyzer sends the new policy to Security Controller via Application Interface.**

Closed-Loop Security Control (3/3)

- Implementation of Application Interface for Feedback delivery to create a closed-loop system of I2NSF Framework.



Closed-Loop Security Control

4. **Security Controller translates a high-level security policy of Application Interface to a low-level security policy of NSF-Facing Interface.**
5. **Security Controller sends the new low-level security policy to NSF via NSF-Facing Interface.**
6. **NSF enforces the requested security policy.**

Conclusion

- This talk introduced an Intent-Based Cloud Security System (ICSS) with I2NSF (Interface to Network Security Functions).
  - Interface to Network Security Functions (I2NSF)
  - Information and Data Models of I2NSF
  - Security Policy Translator with Hackathon Project
  - Closed-Loop Security Control

- I2NSF Framework can work well in OpenStack-Based NFV Environments.

- ICSS is a good structure for network security services for 5G networks.
  - ICSS can provide Open Radio Access Network (O-RAN) with security services.

- As future work, ICSS will be equipped for Policy Assurance and Policy Optimization for Intent-Based Networking (IBN) in network security.