SDN-Based Security Services using I2NSF

draft-jeong-i2nsf-sdn-security-services-05

Jaehoon Paul Jeong, H. Kim,
J. Park, T. Ahn, and S. Lee.

Updates from Version -04

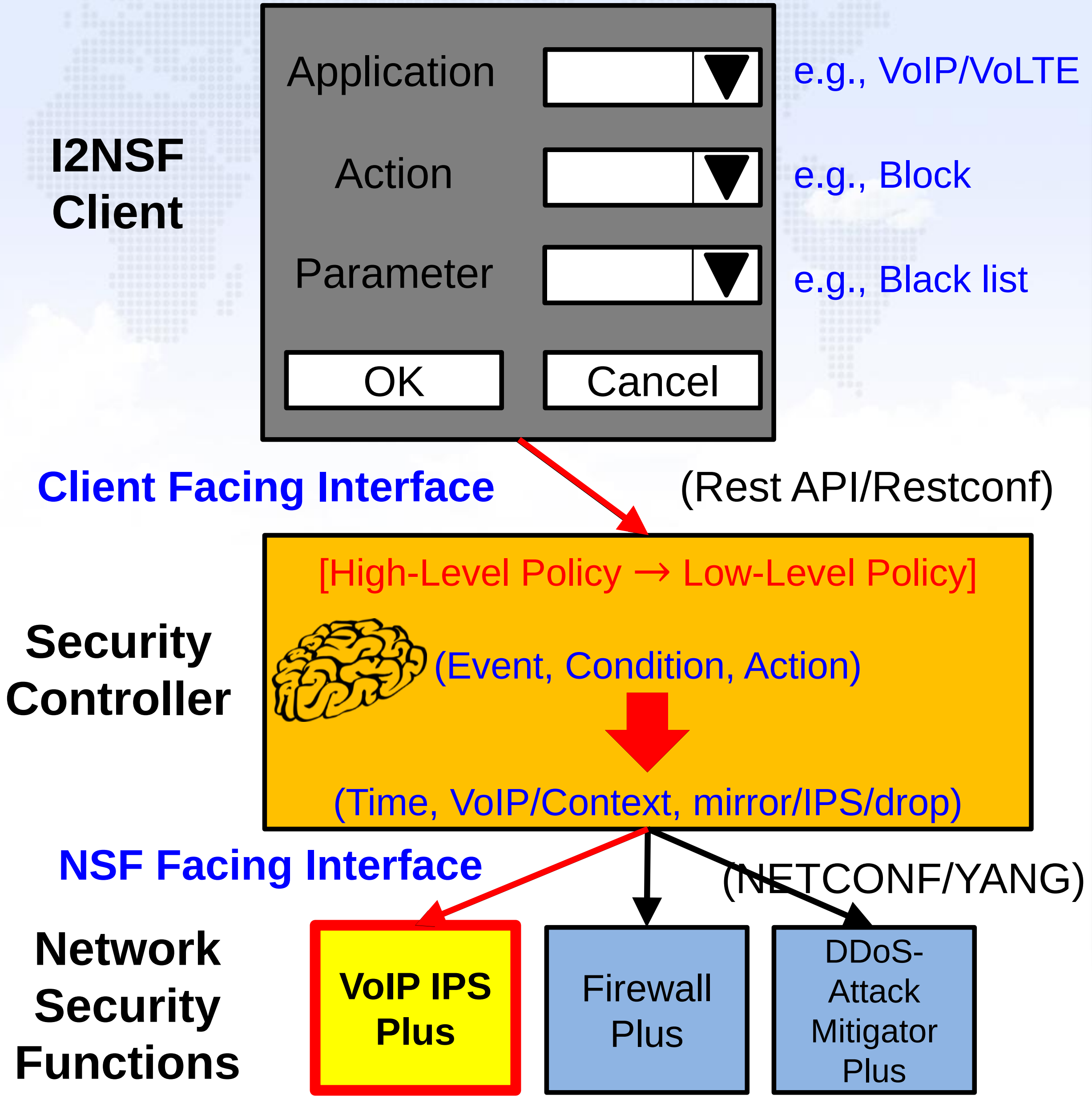
- According to the **change of terminology** in I2NSF framework, the names of the components and interfaces are updated:
  - Application Controller -> I2NSF Client,
  - Capability Layer Interface -> NSF Facing Interface, etc.

- Three use cases in this document can **use a data model** corresponding to the information model for NSF facing interface.
  - draft-jeong-i2nsf-capability-interface-yang-02

- SDN-based security services can **use a security management architecture** for handling security policies.
  - draft-kim-i2nsf-security-management-architecture-01

- Our framework can enforce low-level security policies by **using service function chaining (SFC)-enabled I2NSF architecture**.
  - draft-hyun-i2nsf-sfc-enabled-i2nsf-00

I2NSF Architecture for VoIP IPS



Development Environment

<Platform>
OS: Linux-Ubunt-14.0

<App Gateway>
Language: Javascript, html, xml

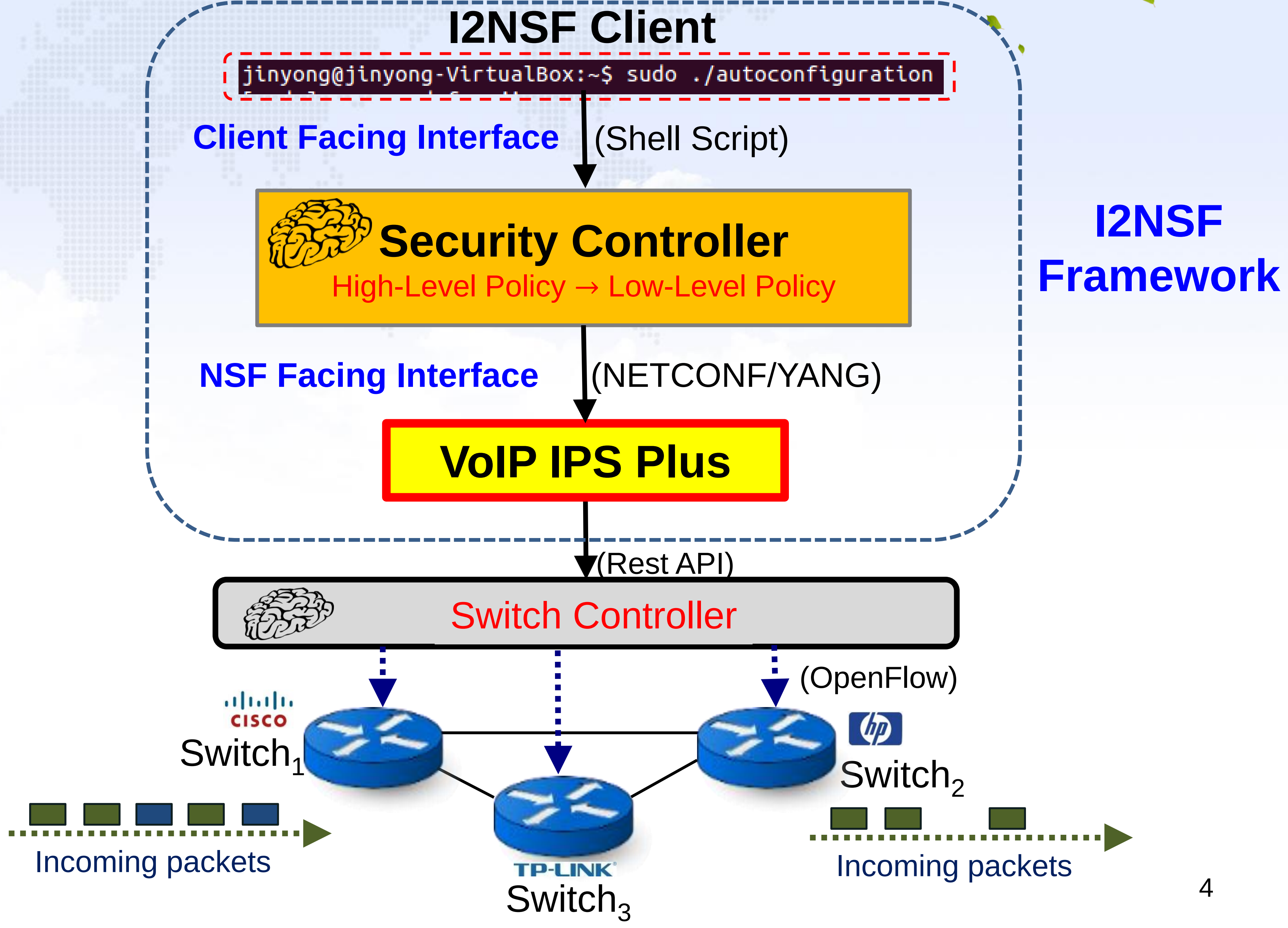
<Security Controller>
Language: Python

<Security Functions>
C Language

<App Gateway-Security Controller Interface>
Rest API

<Security Controller-Security Function Interface>
NETCONF/YANG

Implementation based on OpenDaylight



```
jinyong@jinyong-VirtualBox:~$ sudo ./autoconfiguration
```

Next Steps for this Draft

- Provisioning of the **Information Model** and **Data Model** for the VoIP/VoLTE for Security Controller, i.e.,
  - **Client Facing Interface** between I2NSF Client (for VoIP/VoLTE) and Security Controller, and
  - **Registration Interface** between Developer's Management System and NFS