SFC-enabled I2NSF Architecture
(draft-hyun-i2nsf-sfc-enabled-i2nsf-00)

IETF 96, Berlin, Germany

July 21, 2016

SUNG KYUN KWAN UNIVERSITY(SKKU)

S. Hyun, S. Woo, Y. Yeo,

J. Jeong and J. Park

ETRI

Contents

Introduction

- This document describes an architecture of the I2NSF framework using security function chaining for the traffic steering.
- Service Functions (i.e., Network Security Functions) determine down-stream paths.
- What information is going from Security Controller to Service Function and vice versa?

Motivation

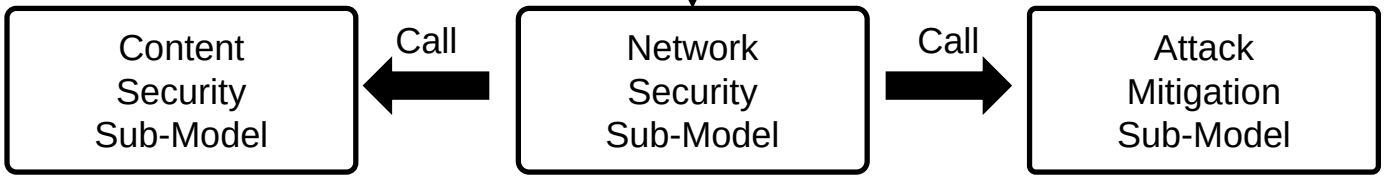
New Information Model & New Data Model

We focused on HOW TO REALIZE??

We need to STEER the packet/flow to the subsequent module(function)

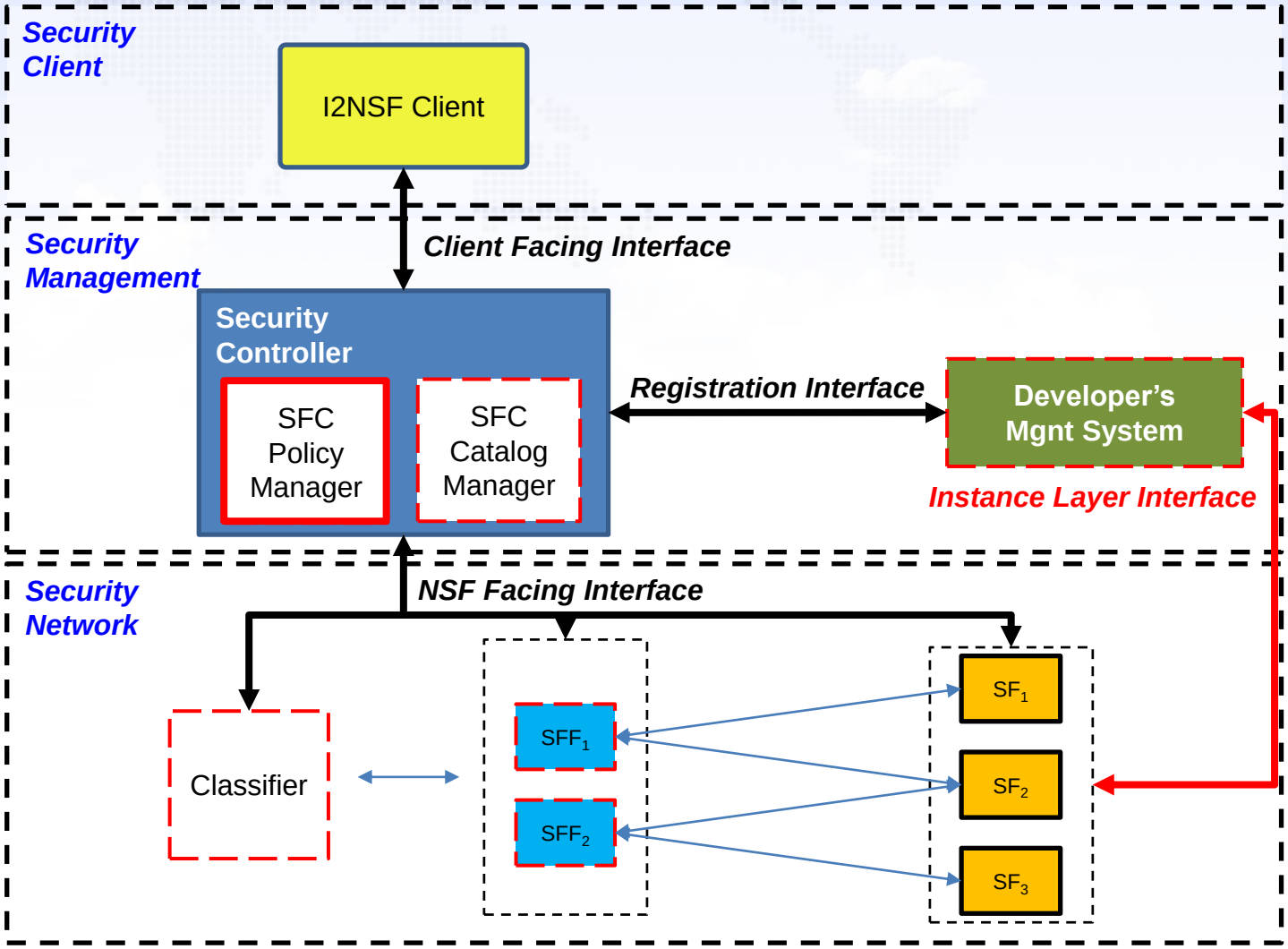
Leveraging SFC to STEER the packet/flow

It usually runs as the first step to handle traffic. According to the results of network security function, the content security and attack mitigation sub-models **can be enforced**



<Information model for I2NSF capability interface>
Reference: draft-xia-i2nsf-capability-interface-im-05

Architecture & Components



SFC Policy Manager

- Interpreting a high-level SFC policy (or configuration) into a low-level SFC policy (or configuration)
- Delivering the interpreted policy to Classifiers.
- Generating and distributing an SF forwarding table

Exchanged Information

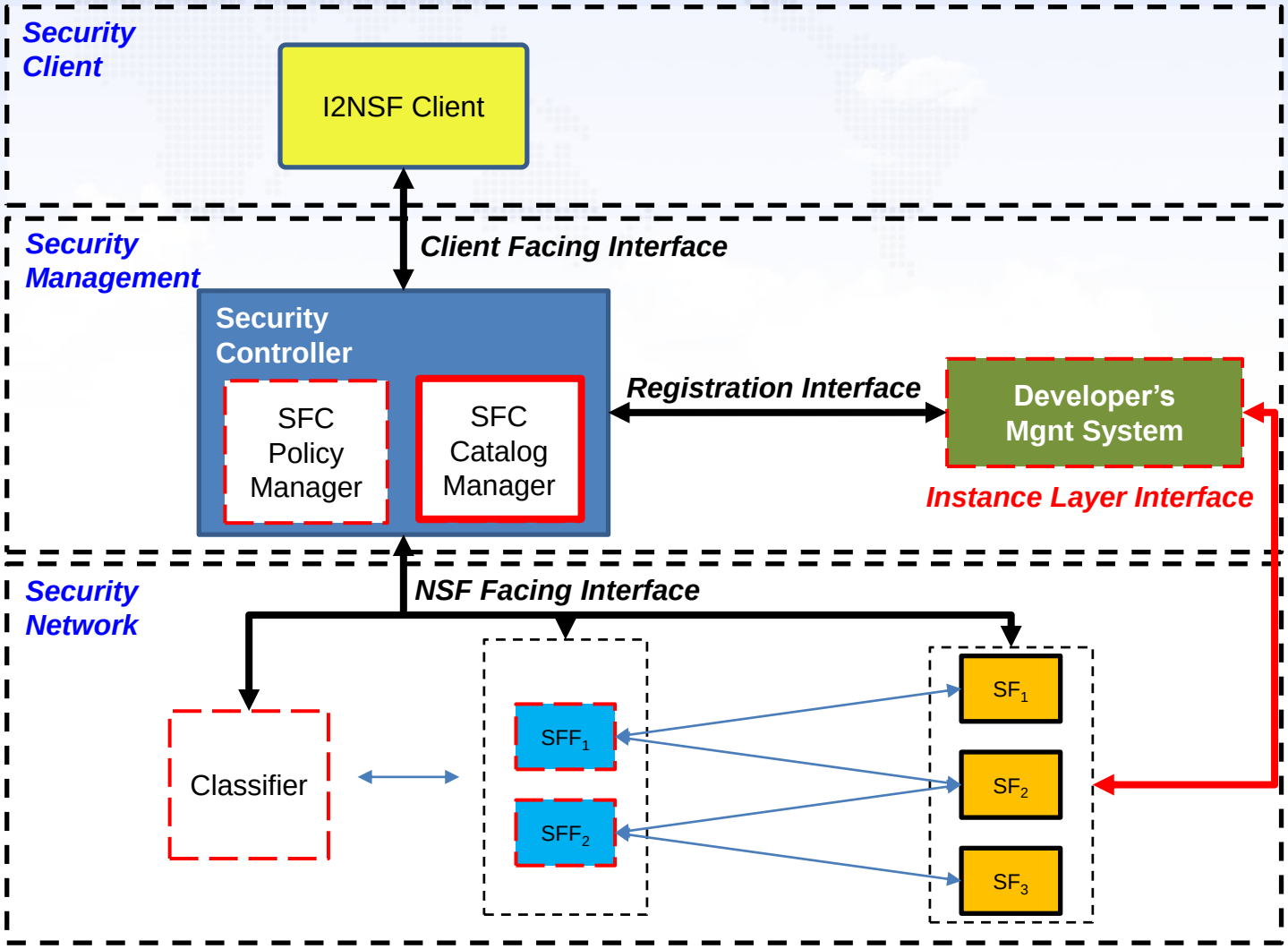
1. Policy

Condition1	Condition 2	Path ID

2. Forwarding table

Path ID	Path Index	Next

Architecture & Components



SFC Catalog Manager

- Maintaining the information of every available SF instance.
- Helping to generate a forwarding table entry relevant to a given SFP.
- Requesting Developer's Management System for the dynamic instantiation or elimination.

Exchanged Information

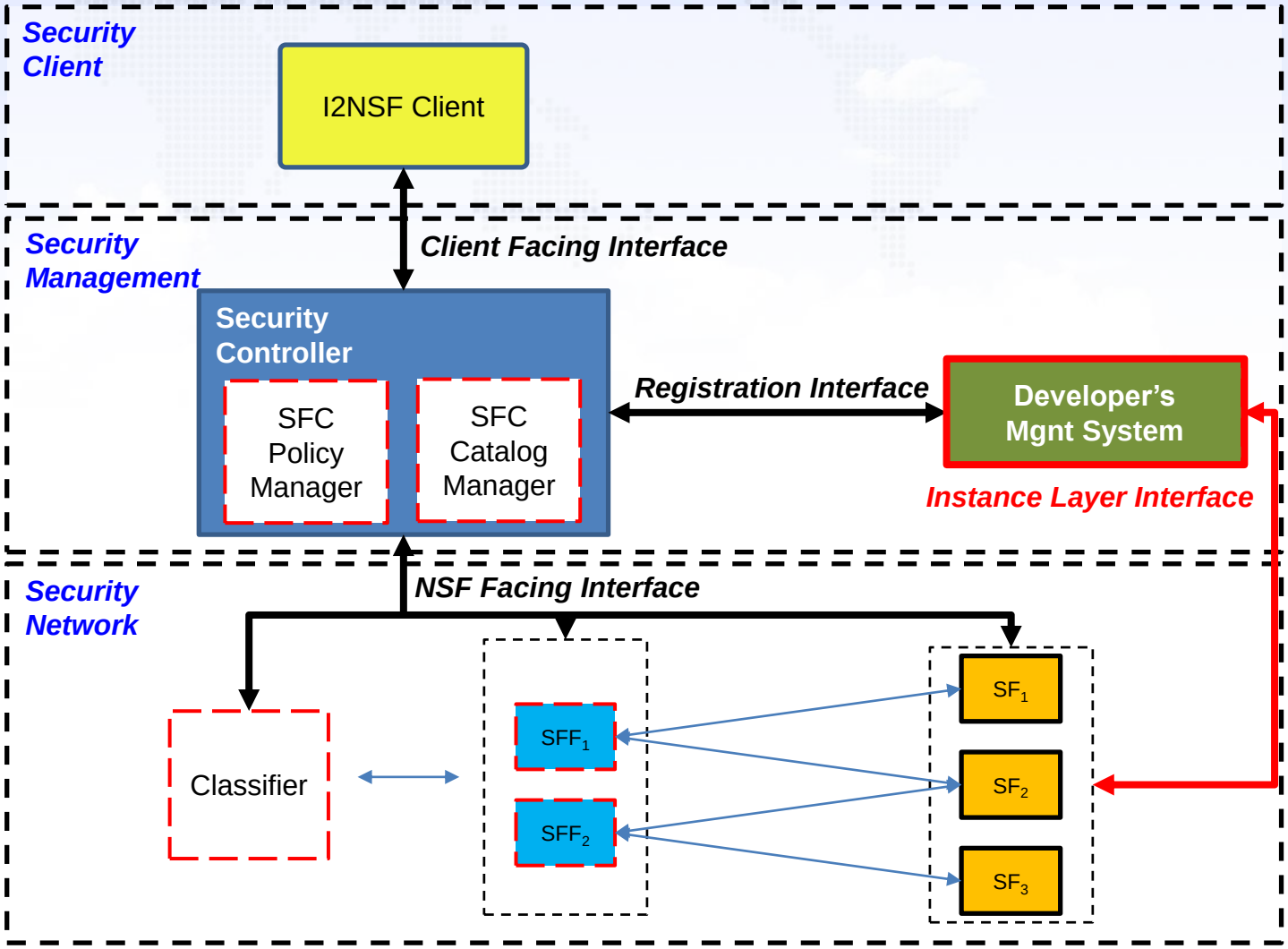
1. Reporting from SFs

SF ID	Load Status	Inspection Result

2. Instantiation or Elimination

SF Type	I or E	Necessary Spec

Architecture & Components



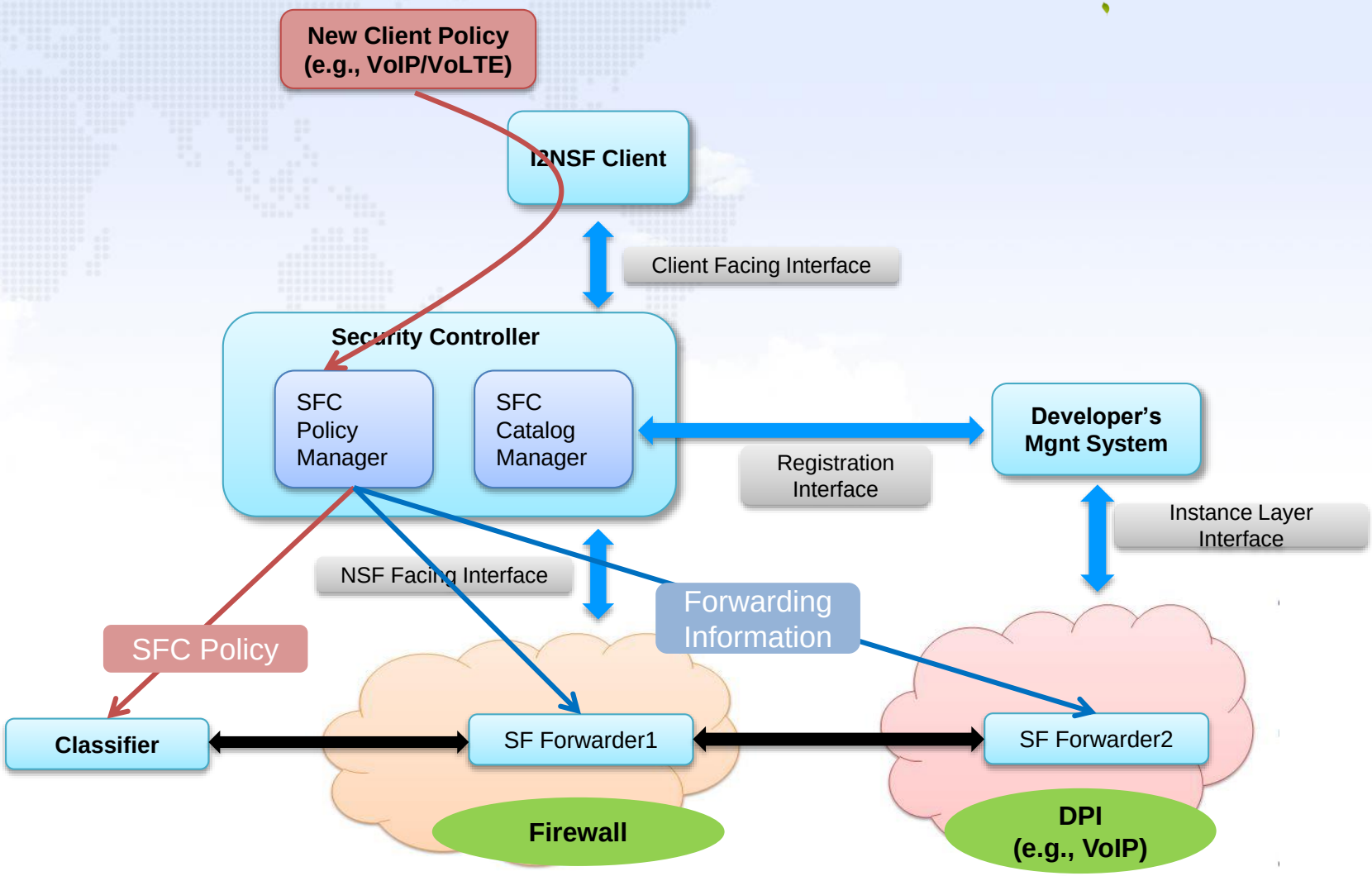
Developer's Management System

This component could exist on third party area (e.g., SF vendors' cloud)

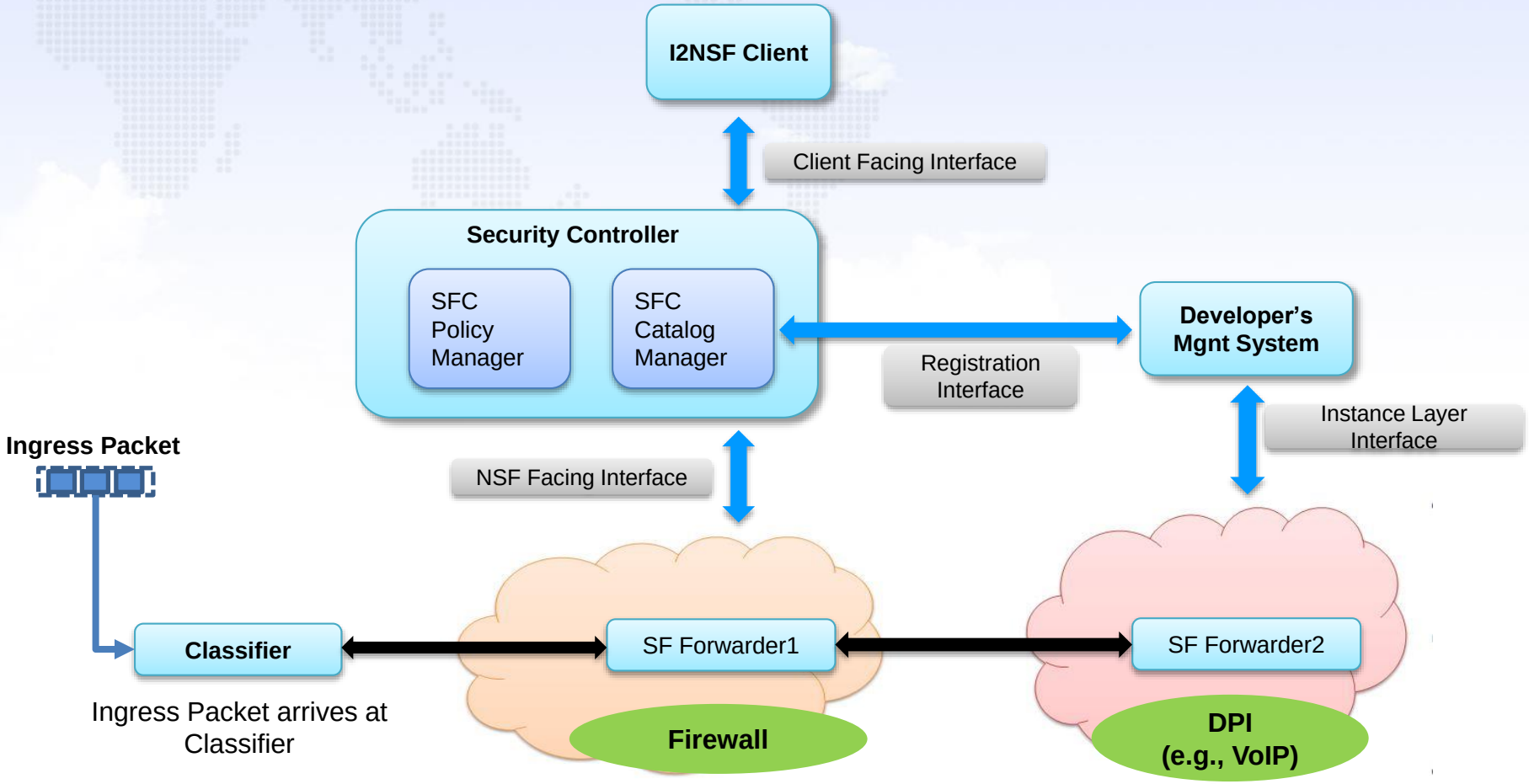
- Creating additional SF instances when the existing instances (called SF) of an NSF are congested
- Eliminating some of the SF instances which are unused

SFC Catalog Manager will request SF instantiation or elimination based on SFs' current status.

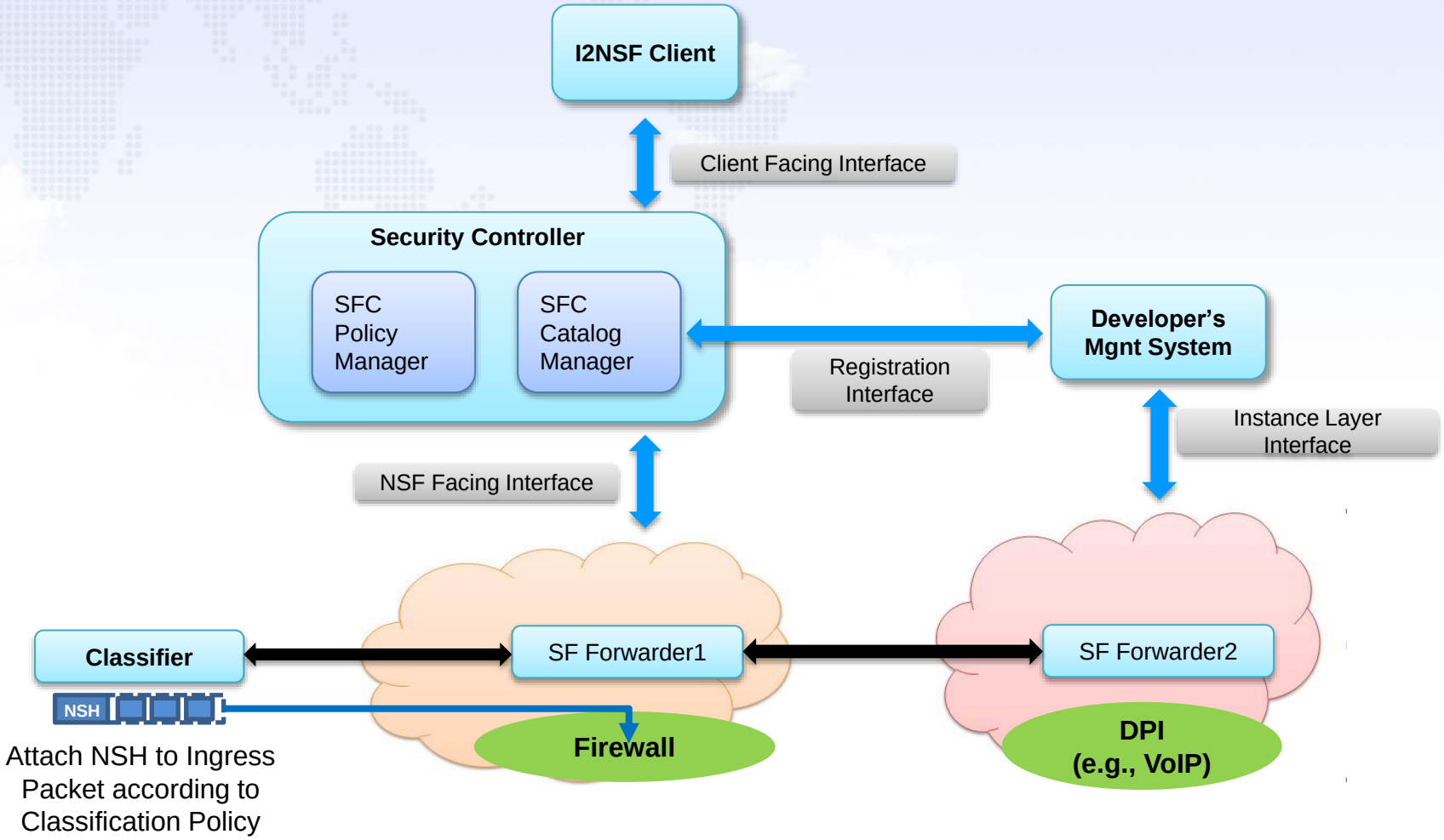
Use Case of VoIP/VoLTE (1/6)



Use Case of VoIP/VoLTE (2/6)

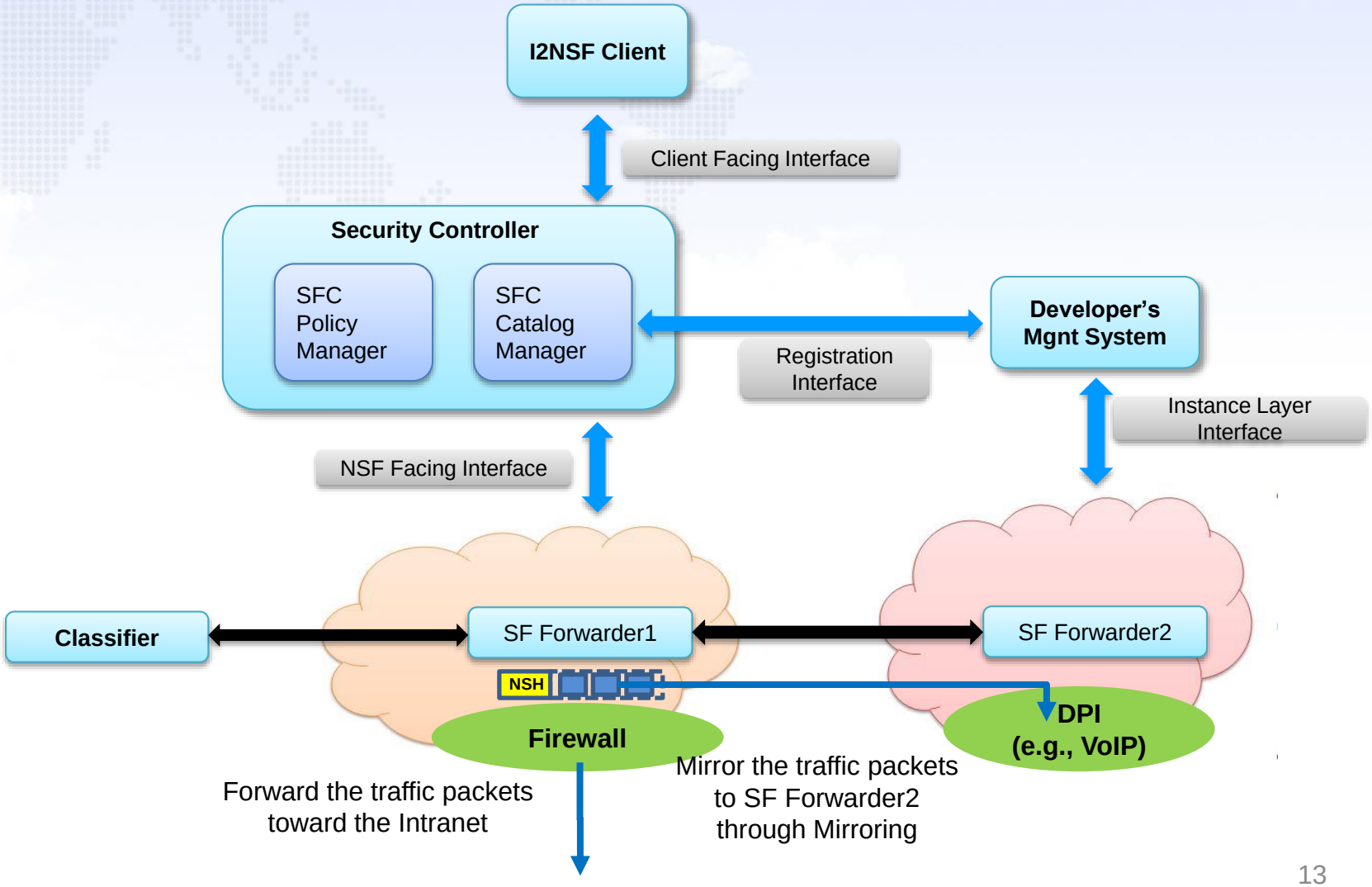


Use Case of VoIP/VoLTE (3/6)

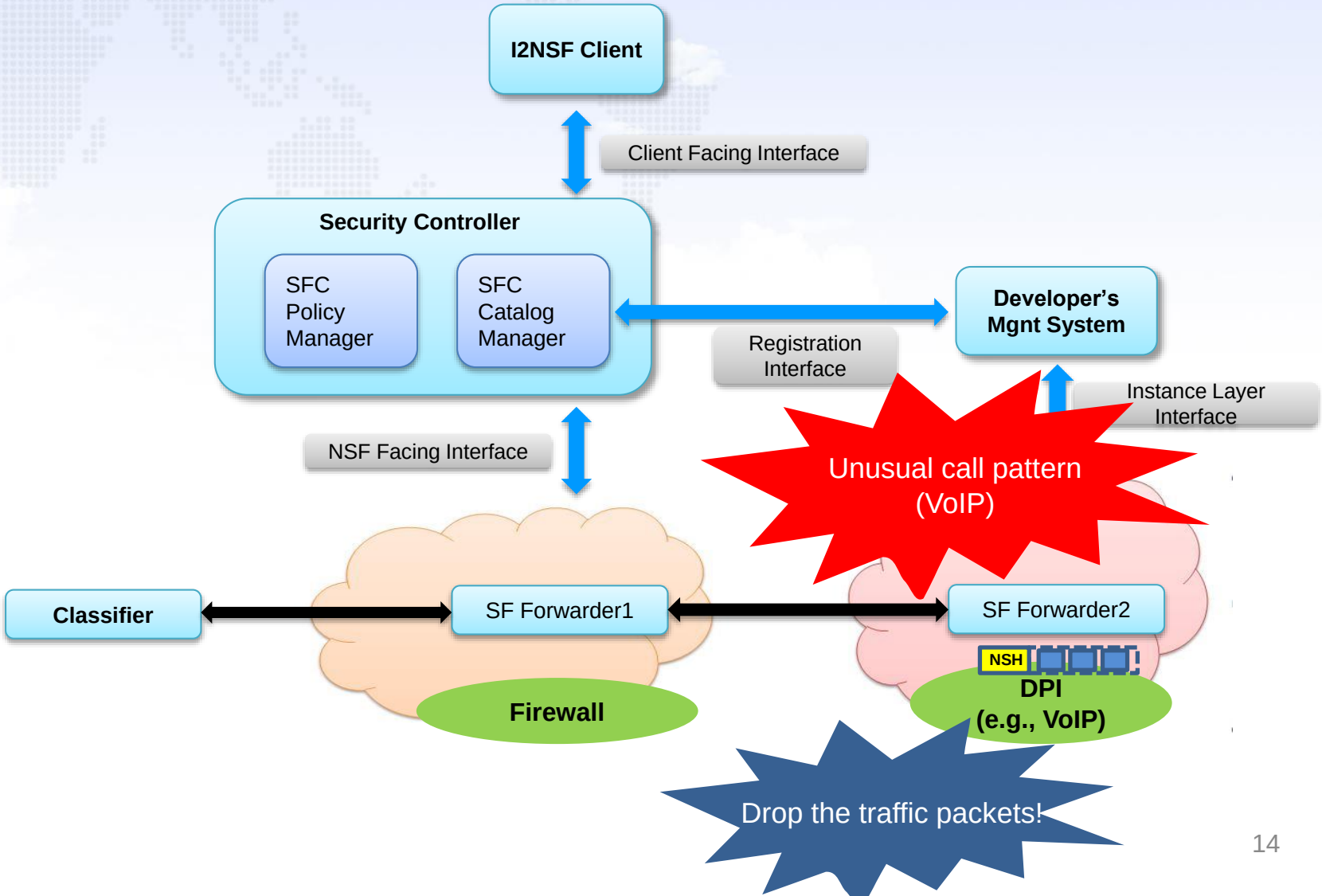


*NSH: Network Service Header

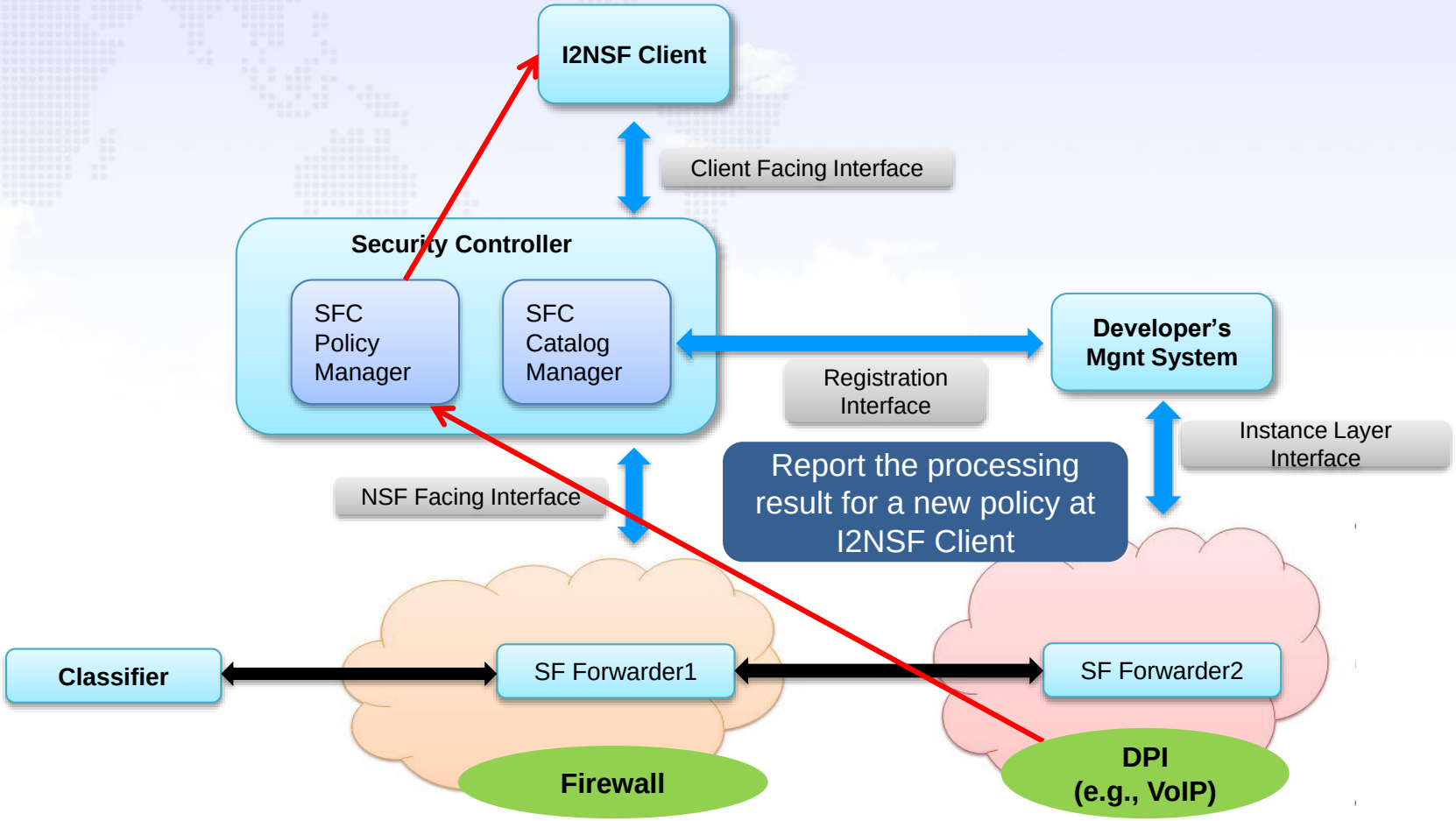
Use Case of VoIP/VoLTE (4/6)



Use Case of VoIP/VoLTE (5/6)



Use Case of VoIP/VoLTE (6/6)



Next Steps

- We will design **Information Models** & **Data Models** for the following two interfaces on traffic steering enabled I2NSF architecture.
  - **Registration Interface**
    - between Security Controller & Developer's Mgnt System
  - **Instance Layer Interface**
    - between Developer's Mgnt System & Security Function
- To prove the effectiveness of our architecture with new data models, we will implement the traffic steering enabled I2NSF architecture.