Security Policy Translation in Interface to Network Security Functions

(draft-yang-i2nsf-security-policy-translation-01)

IETF 102, Montreal

July 18, 2018

Jinhyuk Yang [Presenter], Jaehoon Paul Jeong, and Jinyong (Tim) Kim

Sungkyunkwan University

Motivation

- The **Limitations** of **XSLT-Based Policy Translation**
  1. **Difficulty of Security Policy Construction**
     - I2NSF User MUST select target NSFs for a high-level security policy by himself.
     - This selection requires the knowledge of NSFs corresponding to capabilities from I2NSF User.
     - Thus, I2NSF User MUST be a security expert.

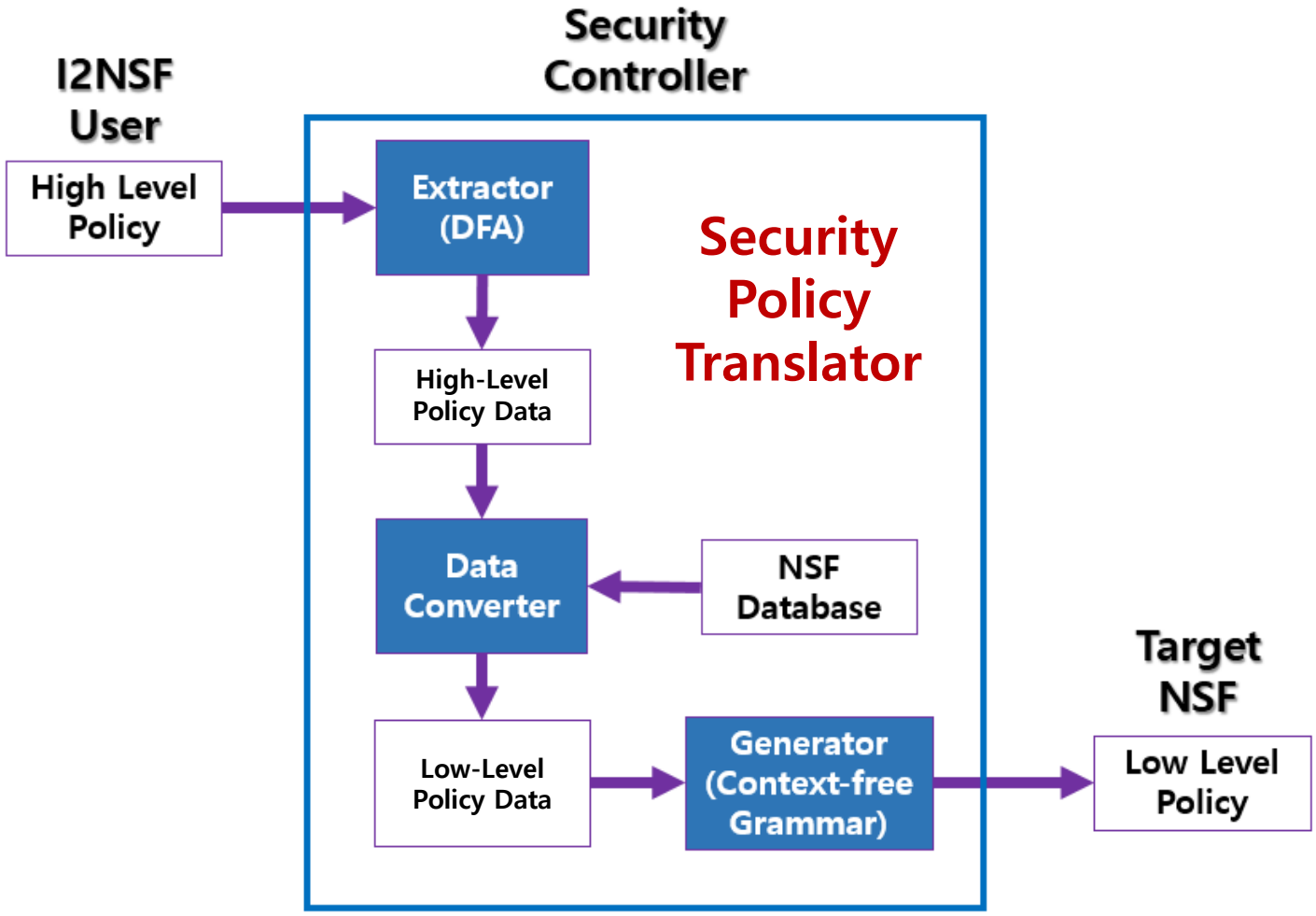
  2. **Inefficient Maintenance in Policy Translation**
     - If a Data Model (in either Consumer-Facing Interface or NSF-Facing Interface) is revised, a system manager SHOULD revise all XSLT stylesheets (i.e., xml files) of each NSF.

Our Approach

- **Automata-Based Policy Translation**
  1. **Ease of Security Policy Construction**
     - I2NSF User doesn't need to select target NSFs for a high-level security policy by himself.
     - This selection will be performed by Security Controller having knowledge of NSFs corresponding to capabilities for the sake of I2NSF User.
     - Thus, I2NSF User doesn't need to be a security expert.

  2. **Efficient Maintenance in Policy Translation**
     - If a Data Model (in either Consumer-Facing Interface or NSF-Facing Interface) is revised, a system manager needs to update only Translation Mapping Information in Security Controller.

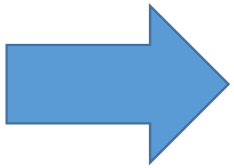
Architecture of Security Policy Translator



Security Policy Translation (Web Filter)

High-level Policy

```
<I2NSF>
  <Policy_web>
      <Rule_id>7</Rule_id>
      <Rule_name>google_block</Rule_name>
      <Position>Staff</Position>
      <Web>google</Web>
      <Time_range>
          <Start_time>09:00</Start_time>
          <End_time>13:00</End_time>
      </Time_range>
      <Action>reject</Action>
  </Policy_web>
</I2NSF>
```



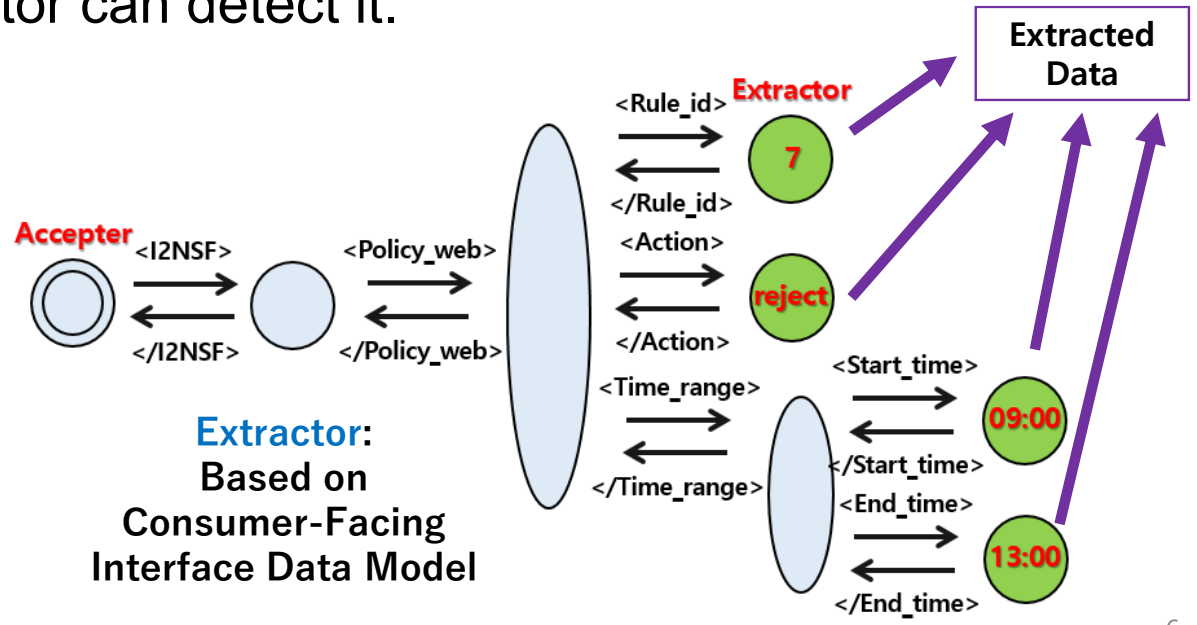
Low-level Policy

```
<pol:policy>
  <pol:policy-id>2</pol:policy-id>
  <pol:policy-name>i2nsf-web-filter</pol:policy-name>
  <pol:rules nc:operation="create">
    <pol:condition>
      <pol:packet-security-condition>
        <pol:packet-security-ipv4-condition>
          <pol:ipv4-src>10.0.0.2</pol:ipv4-src>
          <pol:ipv4-src>10.0.0.4</pol:ipv4-src>
        </pol:packet-security-ipv4-condition>
      </pol:packet-security-condition>
    </pol:condition>
    <pol:payload-content>google</pol:payload-content>
    <pol:schedule>
      <pol:start-time>09:00:00Z</pol:start-time>
      <pol:end-time>13:00:00Z</pol:end-time>
    </pol:schedule>
    <pol:action>
      <pol:action-type>reject</pol:action-type>
    </pol:action>
  </pol:rules>
</pol:policy>
```

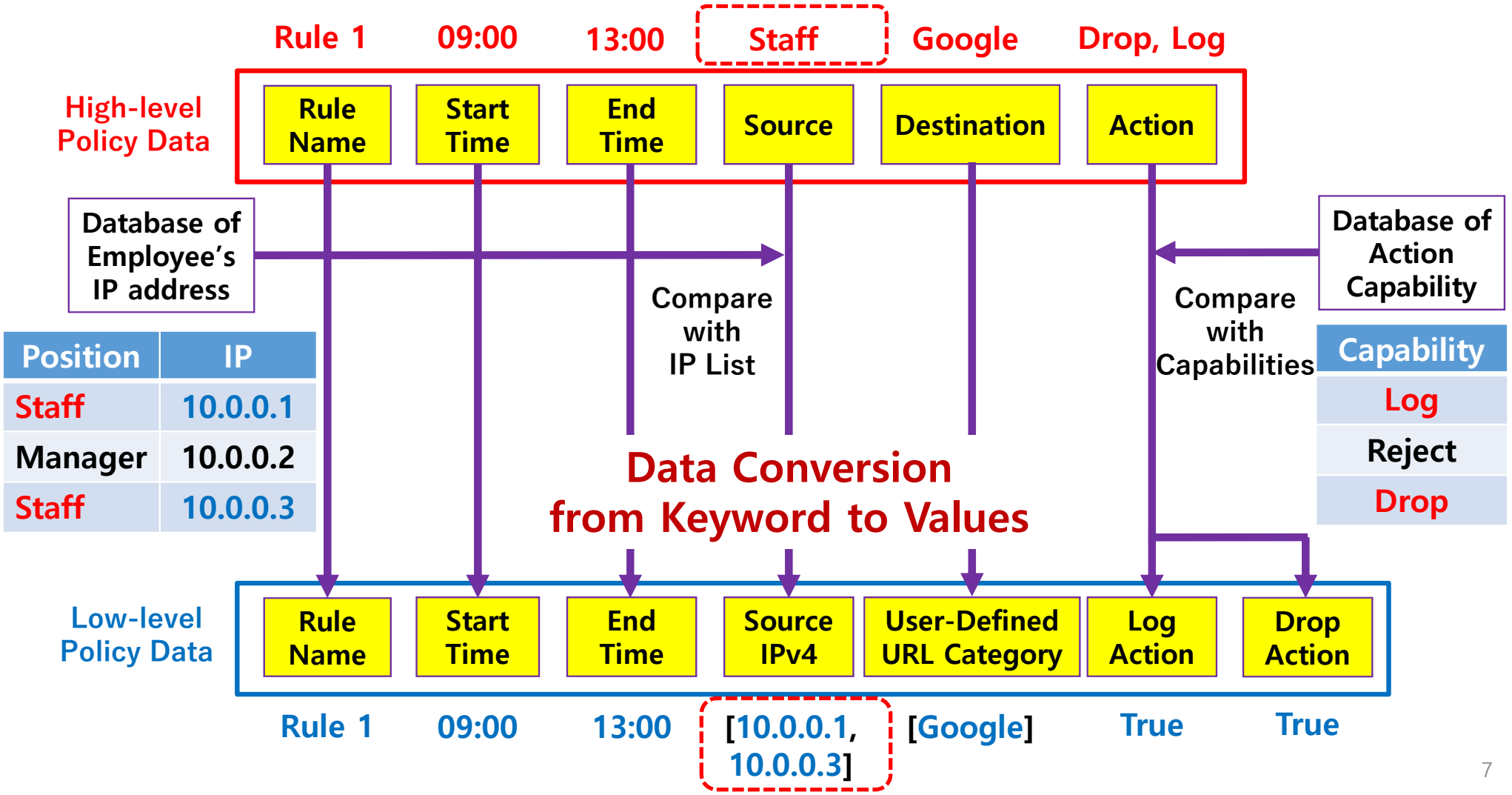
Step 1: Extractor (DFA)

- Easily Extract Data from High-Level Policy
  - Acceptable if a high-level policy follows the rules of a data model hierarchy.
- Detection of Grammar Error
  - If the hierarchy of the policy is wrong or there are some wrong tags, Extractor can detect it.

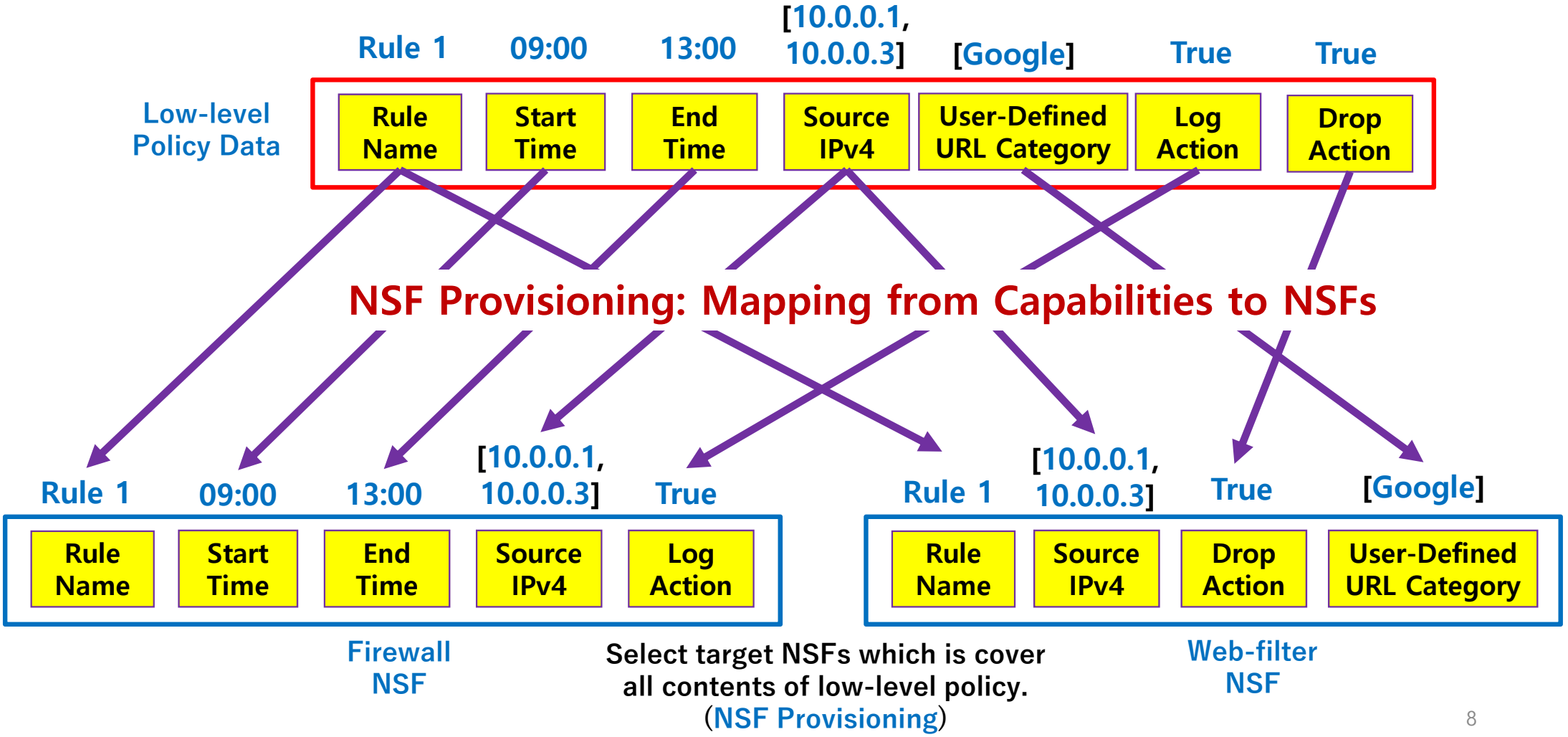
```
<I2NSF>
  <Policy_web>
      <Rule_id>7</Rule_id>
      <Rule_name>google_block</Rule_name>
      <Position>Staff</Position>
      <Web>google</Web>
      <Time_range>
          <Start_time>09:00</Start_time>
          <End_time>13:00</End_time>
      </Time_range>
      <Action>reject</Action>
  </Policy_web>
</I2NSF>
```



Step 2: Data Converter (1/2)

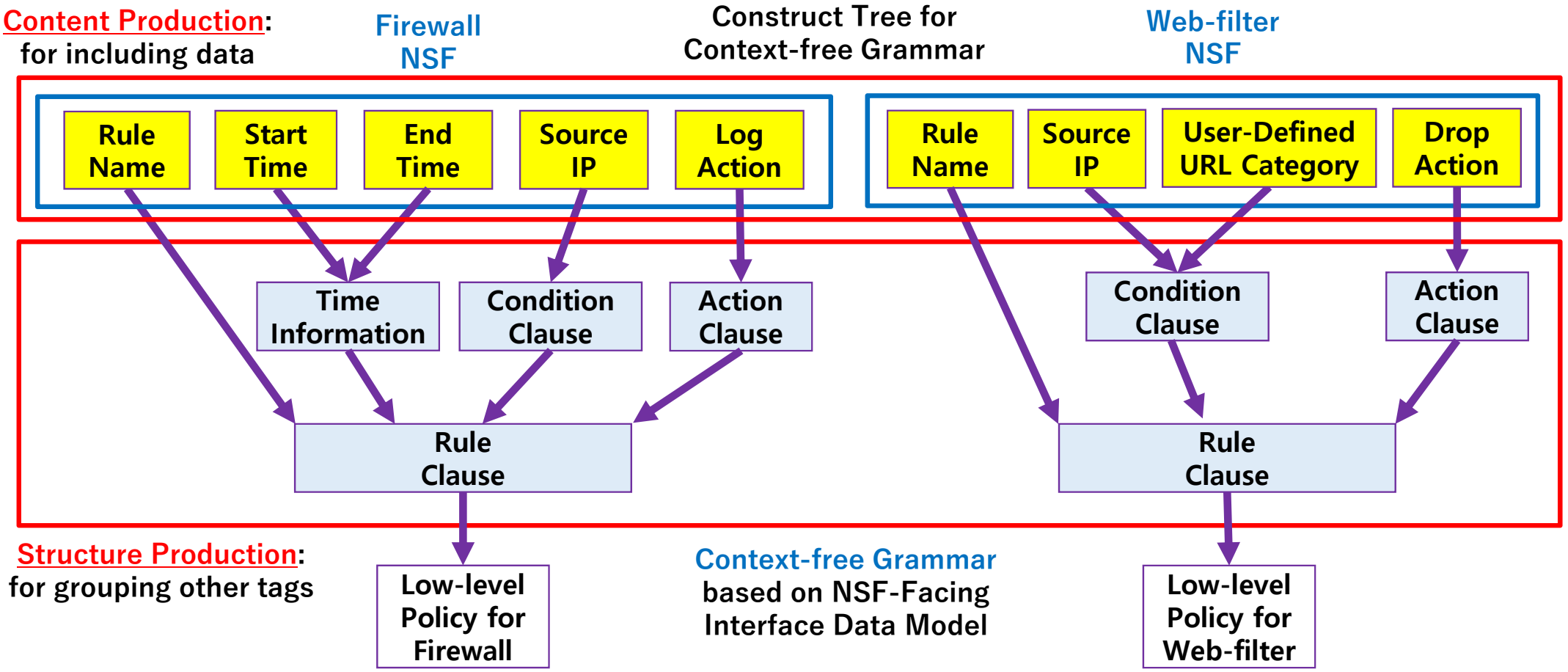


Step 2: Data Converter (2/2)



Select target NSFs which is cover all contents of low-level policy. (NSF Provisioning)

Step 3: Generator



Low-level Policy Construction for NSFs

Next Steps

- **WG Adoption Call** after IETF-102
  - Security Policy Translation is important for I2NSF Implementation.
  - This draft can provide implementers with good guidelines.
  - This draft aims at an Informational RFC.
- We will enhance our draft through IETF-103 Hackathon.
  - We will develop a Tool for Policy Translator management.