Walkthrough of P4-based demo

Panagiotis Famelis, Georgios P. Katsikas

ETSI TFS – Hackfest #3, October 16, 2023

TeraFlowSDN Installation

```
cd ~/controller
```

2

my_deploy.sh

- my_deploy.sh contains all the information needed for the installation of TFS
  - `TFS_COMPONENTS` defines what components we want to deploy
    - We want: Basic + `monitoring` + `policy`
  - `TFS_SKIP_BUILD` defines whether to skip build and use the already built images
    - VM has already built images, in order to save time set it to `YES`
  - `{CRDB, NATS, QDB}_REDEPLOY` defines whether to redeploy the databases
    - It is best to (almost) always redeploy databases, so set it to `YES`

Install TFS

◉ **Run** `source my_deploy.sh`

◉ **Run** `./deploy/all.sh`

◉ The installation should begin and at the end we should have TFS installed

   ◉ **Check with** `kubectl get pods -A`

◉ If some component is not running you can check for any errors with
`kubectl logs <Pod> -n=tfs -c=server`

Instantiate the Mininet topology

```
cd ~/ngsdn-tutorial
```

Mininet topology

◎ The topology we are using can be found here: `mininet/4switch2path.py`

◎ **Run** `make start` to initiate the topology

◎ **Run** `make mn-cli` to check that mininet is up

◎ `MN:nodes` should return 2 hosts (client, server) and 4 switches

◎ `MN: client ping server` should not ping!

The dockerised mininet is taken from ONF's ngsdn-tutorial:
https://github.com/opennetworkinglab/ngsdn-tutorial

Topology
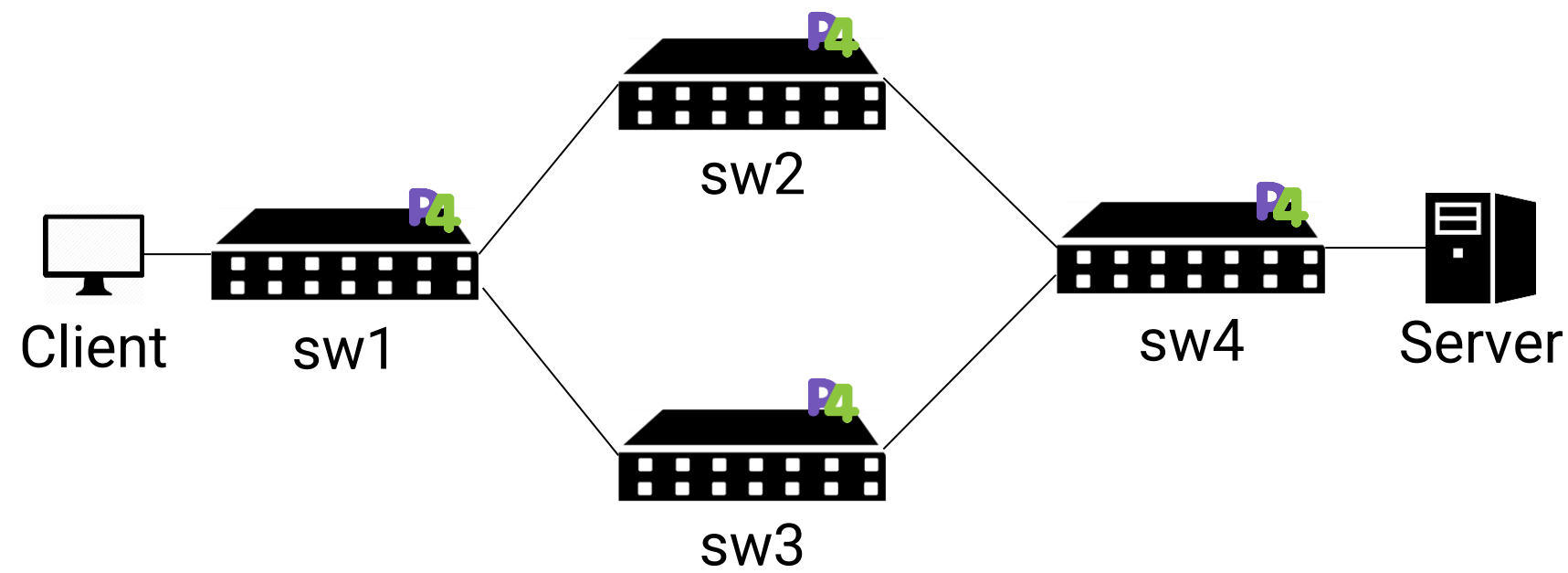
Register devices/service to TFS

```
cd ~/controller
```

Objects.py

◉ The information about the devices, links and services that we are going to upload to TFS is defined in
`./src/tests/hackfest3/test/Objects.py`

◉ Make sure that you change the IP to the correct IP of your VM!

   ◉ `DEVICE_SW{1,2,3,4}_IP_ADDR`

Scripts

◉ There are 5 scripts that we use to register the devices and services to TFS

1. `./src/tests/hackfest3/setup`

   • Copies the p4 compiled code to the sbi pod

2. `./src/tests/hackfest3/run_test_01_bootstrap.sh`

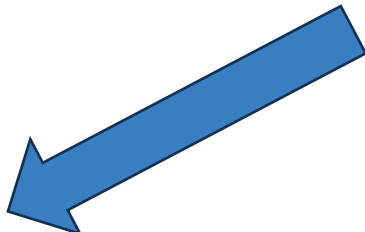   • Registers the devices and links to TFS

3. `./src/tests/hackfest3/run_test_02_create_service.sh`
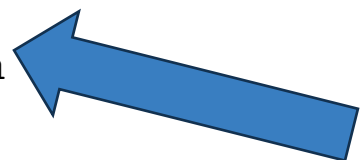
   • Creates a service from client to server

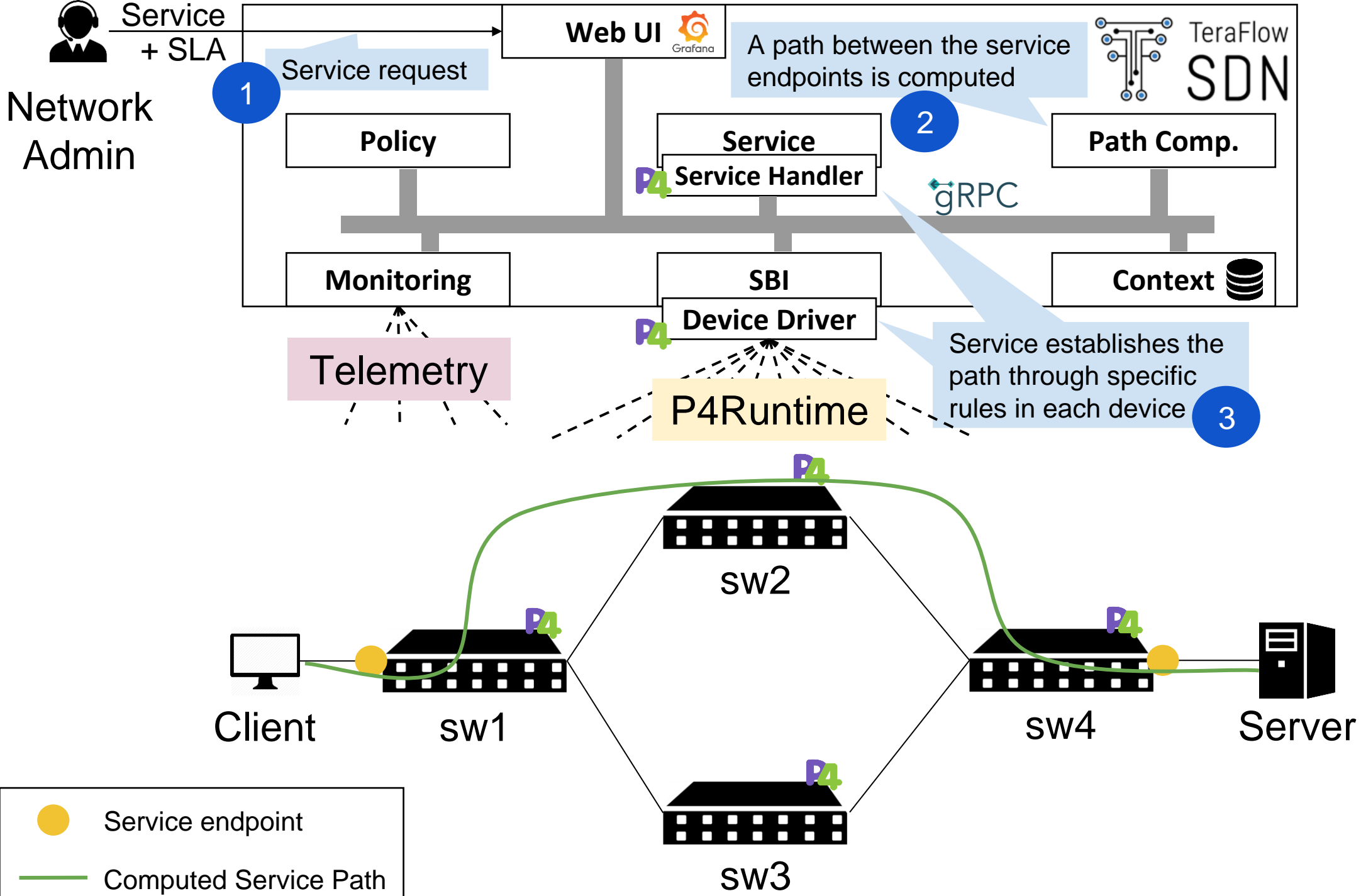4. `./src/tests/hackfest3/run_test_03_delete_service.sh`

5. `./src/tests/hackfest3/run_test_04_cleanup.sh`

Run the scripts until here and check in mininet if we can ping.

Delete the service before continuing! (don't cleanup)

Monitoring

Use the probe to register KPI(s)

Probe

- ◉ **Before moving on make sure that we have the correct** `tfs_environment_varialbes.`
  - ◉ `source ~/controller/tfs_environment_varialbes`

- ◉ `cd src/tests/hackfest3/probe`

- ◉ **Copy the probe to mininet container**
  - ◉ `probe-tfs/deploy.sh`

- ◉ **Connect to the mininet container**
  - ◉ `probe-tfs/connect-to-mininet.sh`

- ◉ **Run** `tfsagent.sh`
  - ◉ `./tfsagent.sh`

Probe (part 2)

◉ In another terminal establish the service

   ◉ `./src/tests/hackfest3/ run_test_02_create_service.sh`

◉ From inside mininet run the `tfsping.sh`

   ◉ We are going to use screen in order to run it in the background

      ◉ `MN: client screen -S ping`

      ◉ `./tfsping.sh`

      ◉ `Ctrl+A` **and** `d` returns to the main mininet cli

Monitoring Visualization

Grafana setup

Import Dashboard

- Import the `grafa_dashboard.json` to Grafana to show the kpi measurements

16

Policy setup

Policy

◉ Let's look at the Policy message

```
"serviceId": { },
"policyRuleBasic": {
    "booleanOperator": "POLICYRULE_CONDITION_BOOLEAN_OR",
    "actionList": [
        {"action": "POLICY_RULE_ACTION_RECALCULATE_PATH"}
    ],
    "conditionList": [
        {"numericalOperator": "POLICYRULE_CONDITION_NUMERICAL_GREATER_THAN",
         "kpiValue": {"floatVal": 100},
         "kpiId": {}
        }
    ]
```
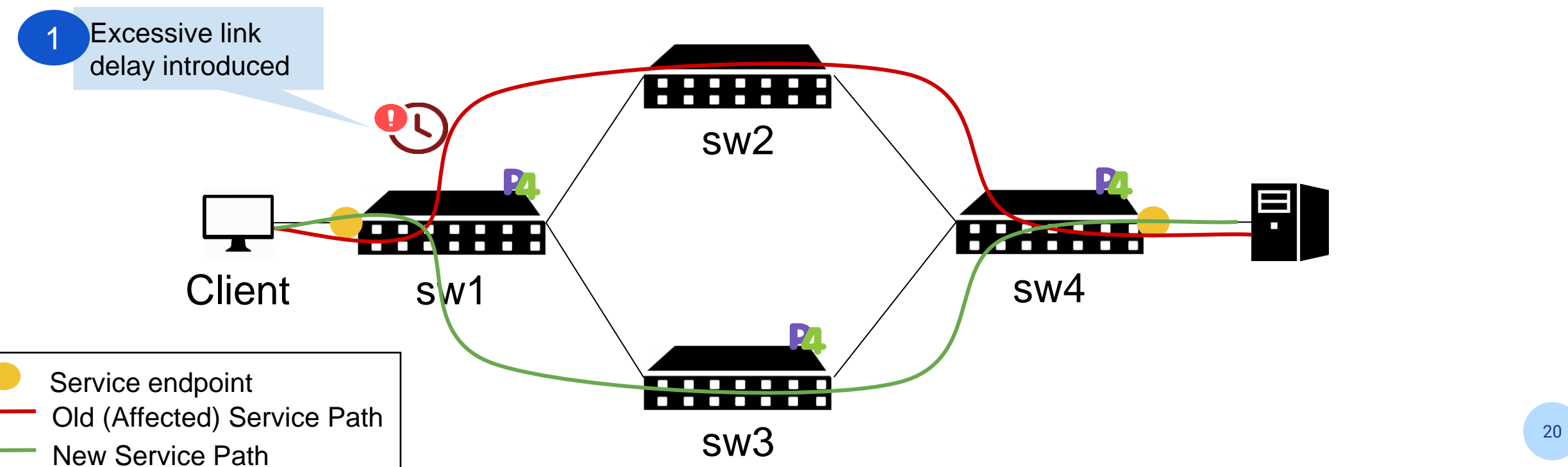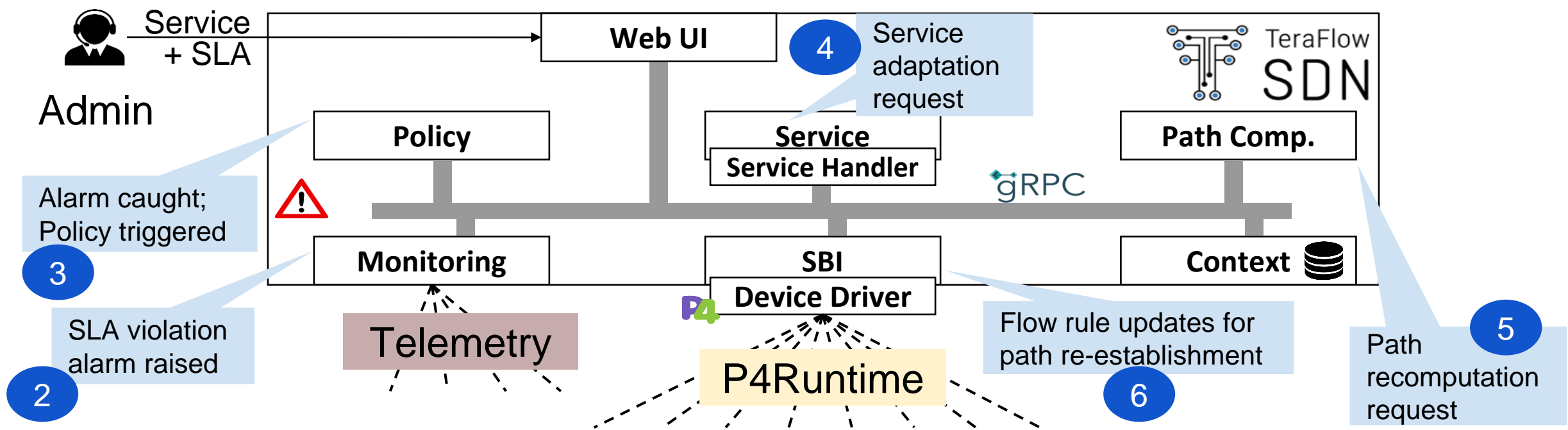
◉ Use a gRPC client (Kreya/Postman) to insert this policy

Add delay

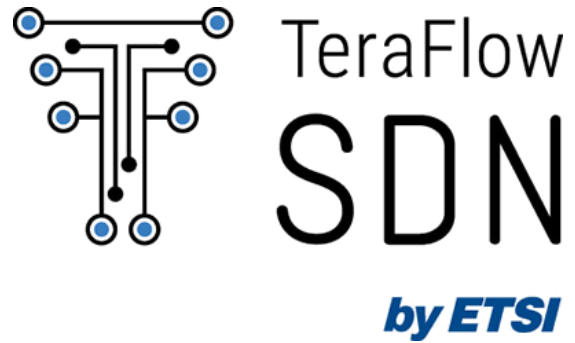◉ To add delay on a link, we use the following:

    ◉ `tc qdisc add dev <switch-iface> root netem delay ` <mark>`<x>`</mark>`ms`

◉ Policy should catch this change and change path

Closing

◉ Before closing make sure:

  ◉ Remove policy

  ◉ Remove Service

  ◉ Stop probe

  ◉ Stop `client screen`

  ◉ Run cleanup

Thank you!

TFSsupport@etsi.org