OpenKilda

Stream Processing Meets OpenFlow

Jeff Young
Product Architecture, Global Platforms

TELSTRA

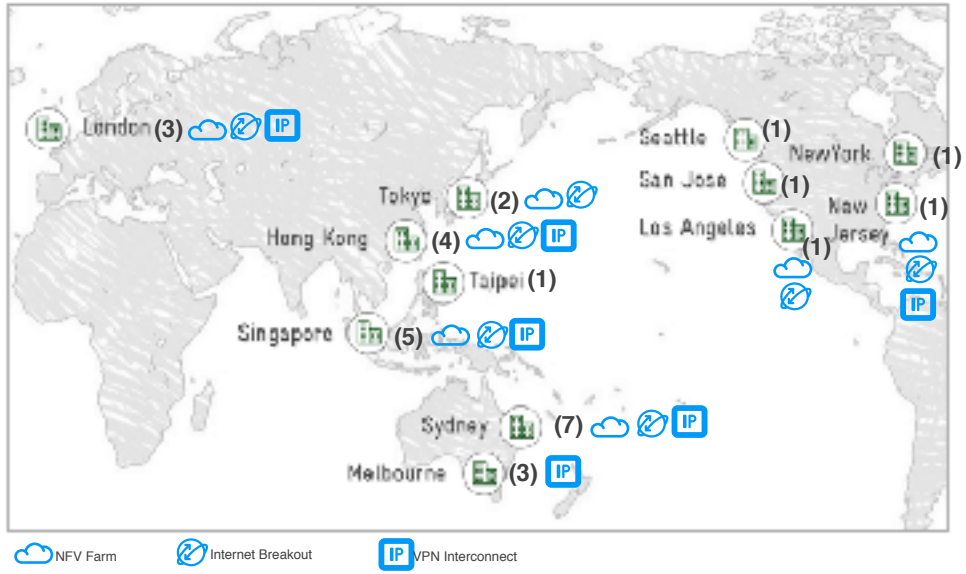
Agenda

- What is the Telstra Programmable Network?
- Why Build on Openflow?
- Why Create (yet another) Openflow Controller?
- Our Solution
- Current State of the Project
- What's Next?
- Get Involved!

TELSTRA

TPN – Telstra Programmable Network

NFV Farm	Sydney, Hong Kong, Tokyo, Singapore, Los Angeles, New Jersey, London
Internet	Sydney, Hong Kong, Tokyo, Singapore, Los Angeles, Secaucus New Jersey, London
Telstra NextIP Interconnect	Sydney, Melbourne
IPVPN/GWAN Interconnect	Hong Kong, Singapore, New Jersey, London
External exchanges	AWS (7), ECX (6), AxonVX (2), Epsilon (2), Westin (1)



35 Programmable Network PoPs in 17 cities and 11 countries and territories

Directly connected to Telstra's network of more than 2,000 points of presence in 200 countries

Connect to leading public cloud providers via various DC Exchange providers

- Cisco CSR1000v

- Fortinet Fortigate NGFW
- Fortinet Fortiweb vAppliance
- Palo Alto VM-series
- Cisco vASA

- Riverbed vSteelhead
- VeloCloud
- Juniper vSRX

- Cisco Primare vNAM
- FortiAnalyzer

TPN Platform

Transform customer experience | Programmable (GUI/ APIs) | Consumption based | Data & Analytics | Global

Marketplace

uCPE

Virtual Network Functions
vFirewall, vRouter, SD-WAN

Global Exchange

Public Cloud

Data Centre

IPVPN/ Network

Multiple use cases

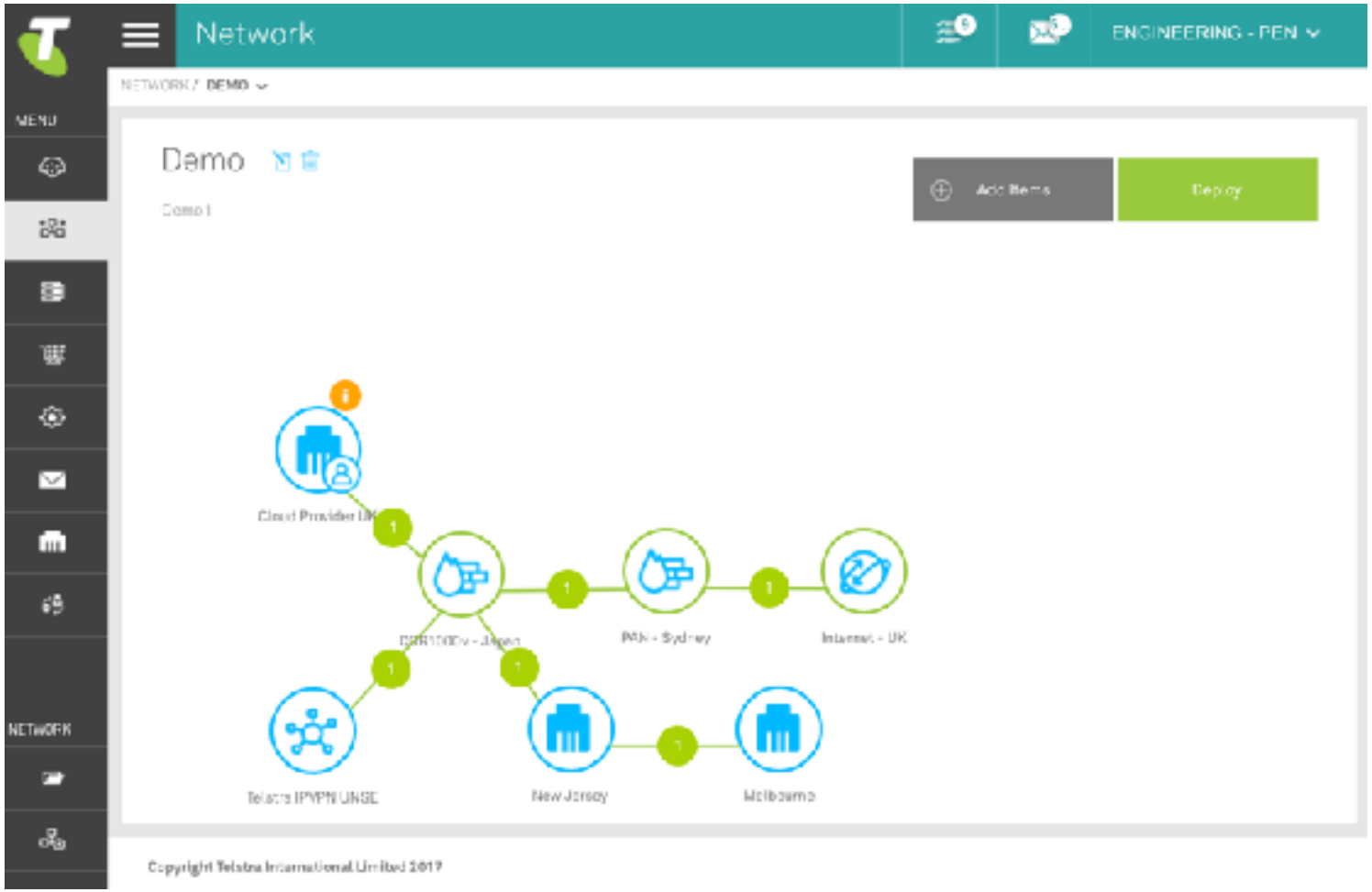
Cloud Connectivity, Data Centre Interconnect, Secure Internet Access, Virtual Branch Networks

Release 1 (April 2017)
R1.0 enables our IPVPN customers to connect to TPN portal and access the TPN capabilities.

Release 2 (July 2017)
R2.0 enables our Next IP customers to connect to TPN portal and access the TPN capabilities.

Release 3 (October 2017)
R3.0 enables our IPVPN and Next IP customers to deploy VNFs on uCPE housed at their premises

TPN Build Blocks



Customer Driven (created)

- Multiple canvases
- Multi-tenant
- Create flows (L2) between building blocks

“Lego” Building Blocks:

- IPVPN’s
- Exchanges
- VNF’s (in server farms)
- Internet
- Switch Ports
- Bandwidth (on demand)

Open Marketplace

- Bring your own ’service’

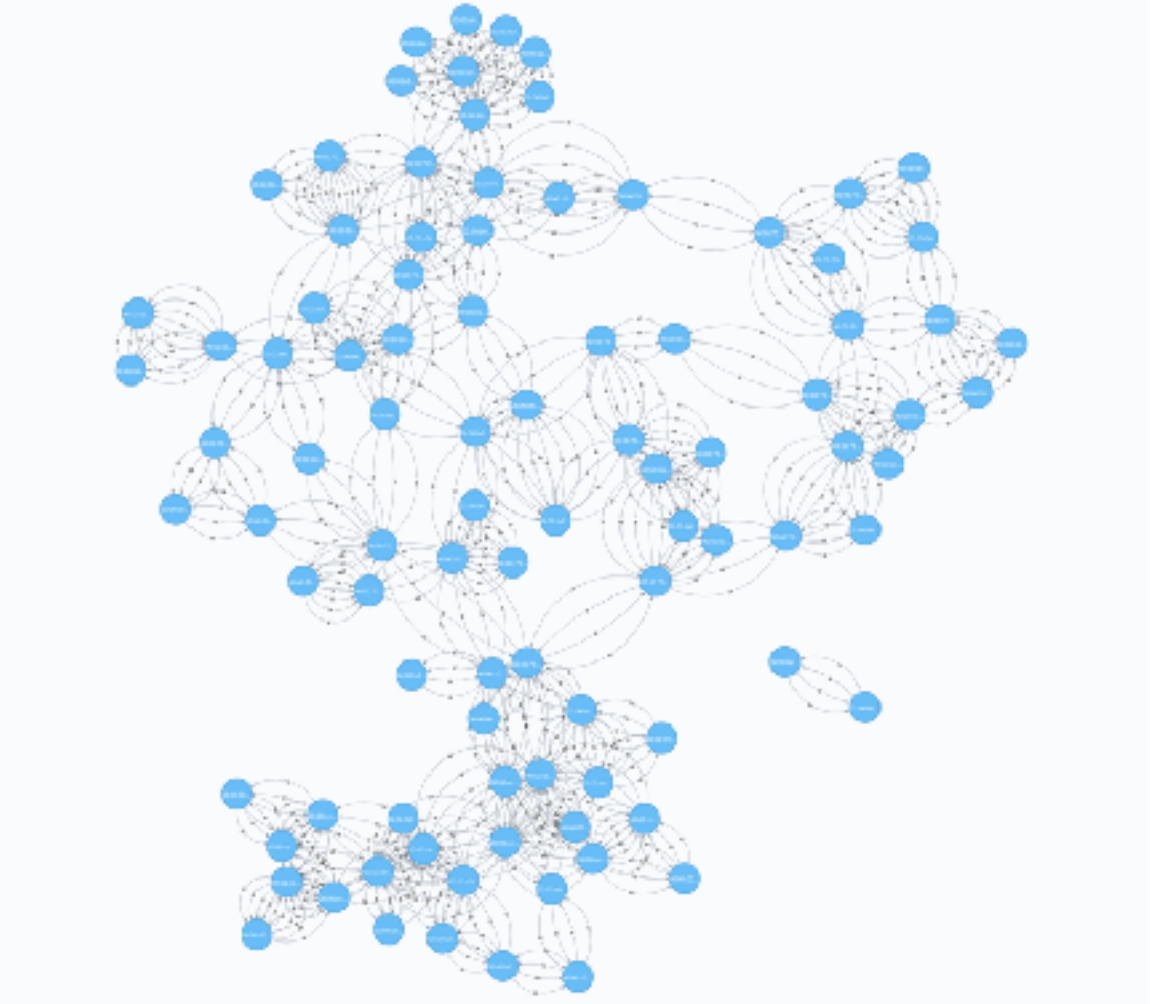
One-Click “Deploy”

- Guaranteed deploy time
- Changes as-needed (self-provision)

Why Build Yet Another OpenFlow Controller?

A few of the existing controllers available today:

Our Challenge Was A Bit Unique
At least we thought it was



- Global network with POPs in Europe, US, Asia, Australia and Middle East
- Control Plane with >300ms of latency
- Controllers located in Hong Kong
- Combination of Dark Fiber and Lit Circuits that don’t all support Link Loss Forwarding
- Guaranteed service, uncontended network

TELSTRA

What We Found

- Constant topology changes
- Network changes increased with network complexity
- Correlation of multiple events

Convergence

- 100K's messages into/out of the controller
- Managing >1M Flows

Events

- LAN based controllers
- High latency in Control Plane

WAN

TELSTRA

Features We Wanted

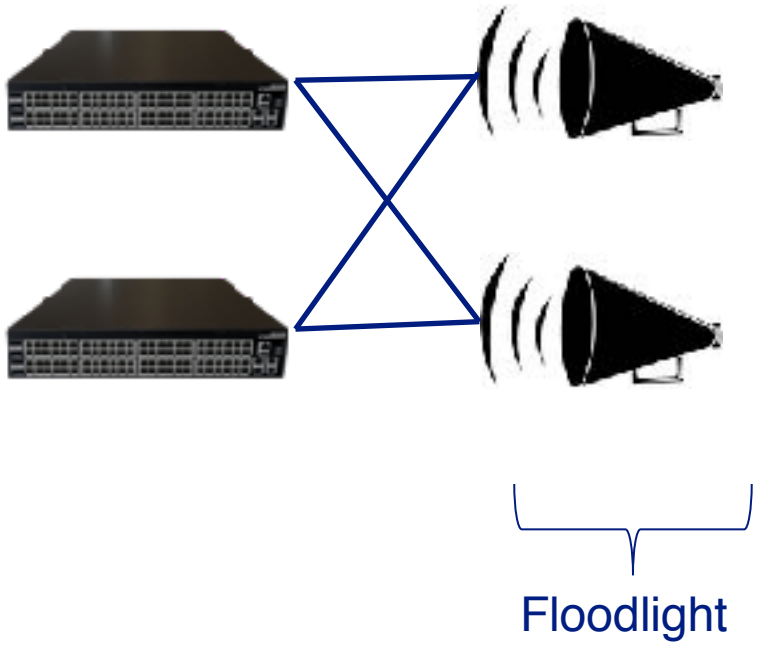
Operations:
- Sub-Second Failover
- Auto-re-route based on real-time latency/packet loss/jitter measurements
- Self Healing/Optimizing Network
- Zero Touch Controller Deployment/Upgrade

Architecture:
- Horizontal scale
  - Number of switches
  - Number of flows
- Negative Affinity In Path Selection
- Path Selection Based on Latency
- Multiple data points for comprehensive end-to-end network state

- Product:
  - Complex match/actions using experimenters
  - Stats collections at 1 second intervals
  - Active Latency Measurement on ISL
  - End-to-End Latency Measurements on every flow

Our Solution

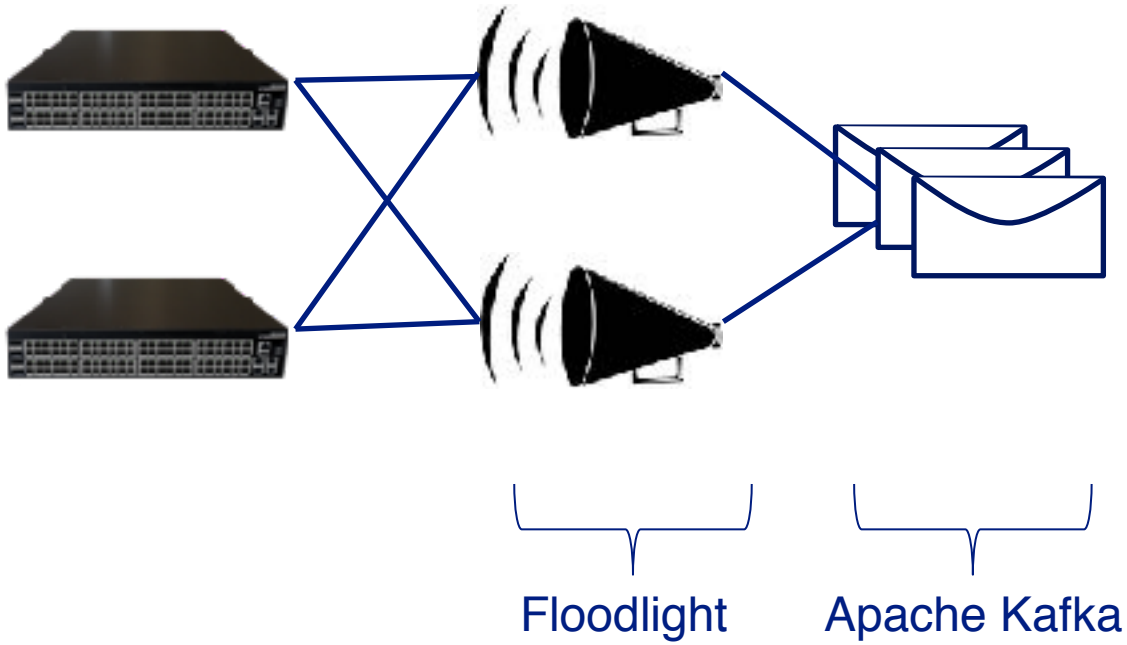
Regionalized OpenFlow Speakers



TELSTRA

Our Solution

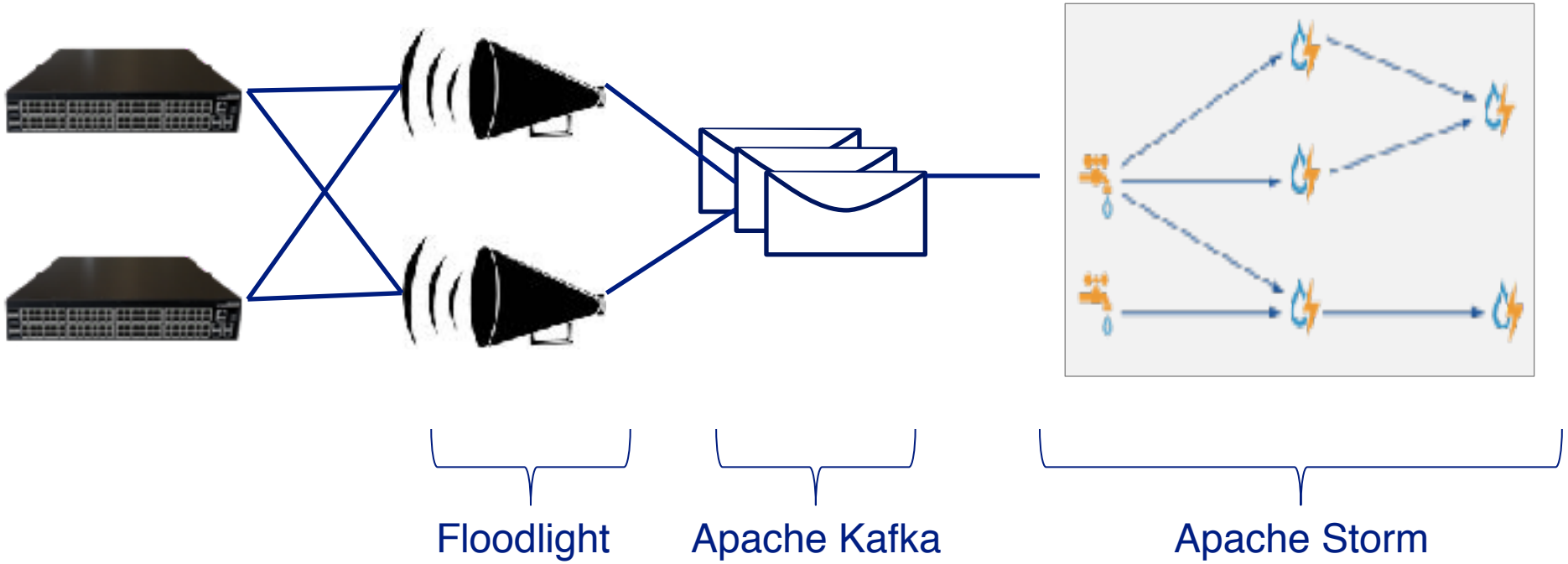
Message Queue as ESP Bus



TELSTRA

Our Solution

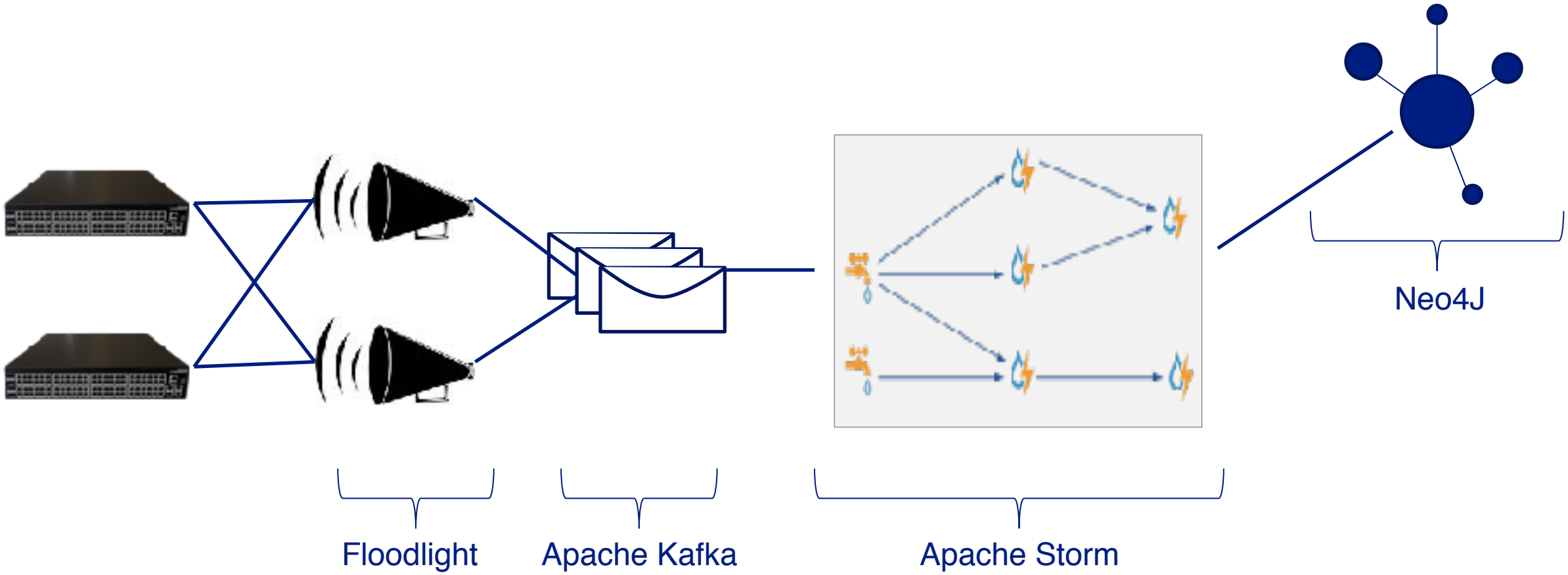
Real-time Stream Processing via Apache Storm



TELSTRA

Our Solution

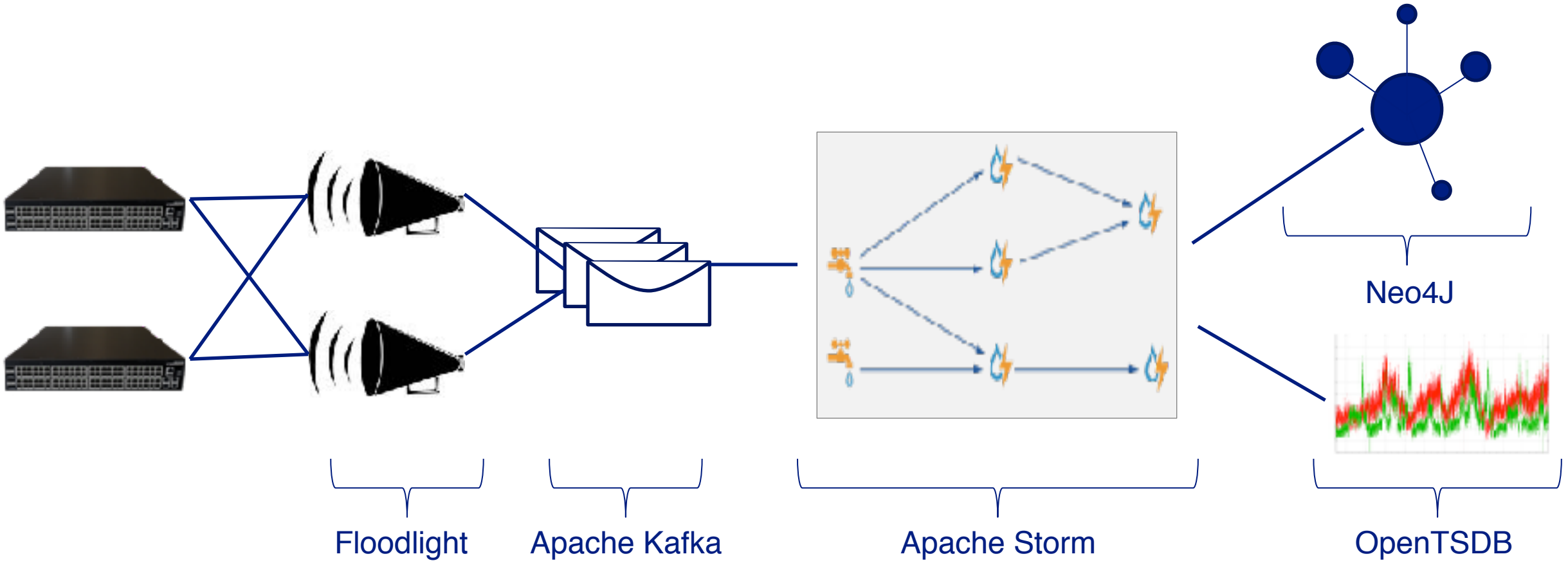
GraphDB - based on Neo4j



TELSTRA

Our Solution

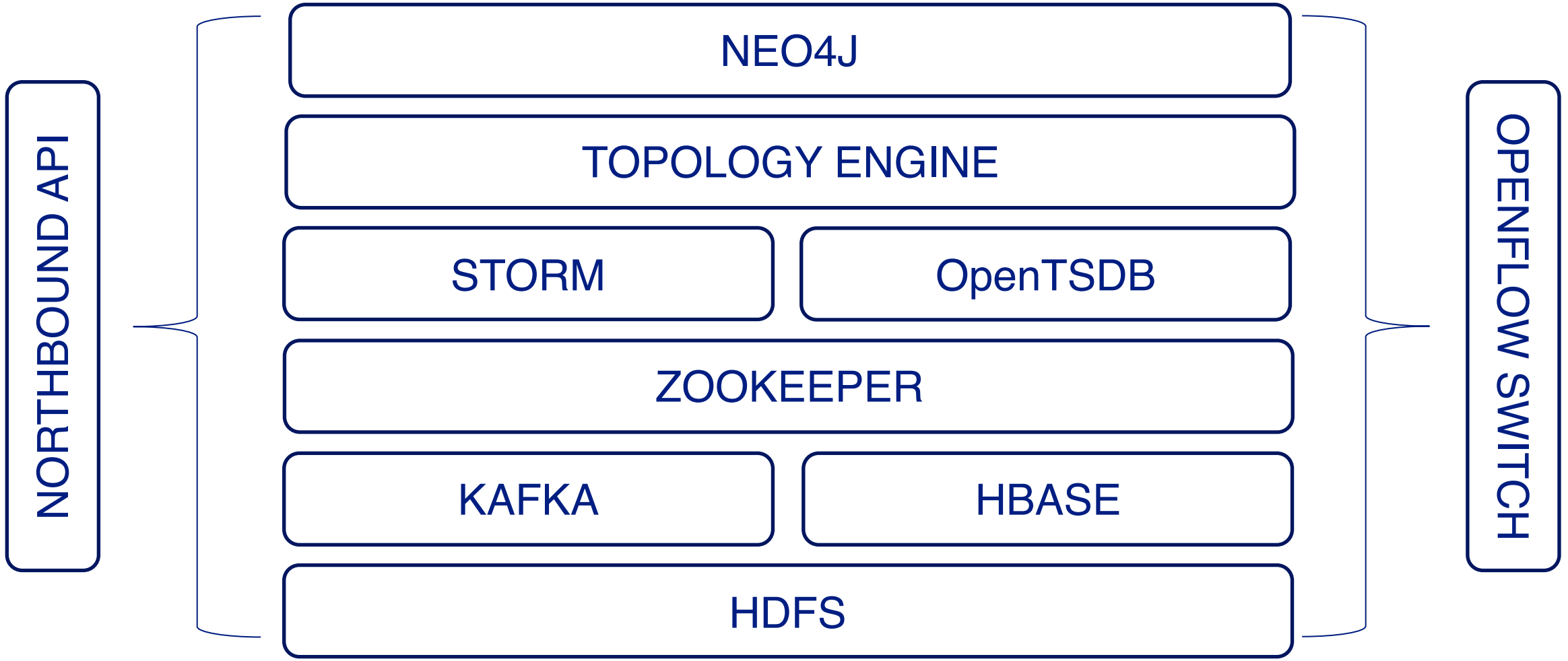
Reporting via OpenTSDB and HBase



TELSTRA

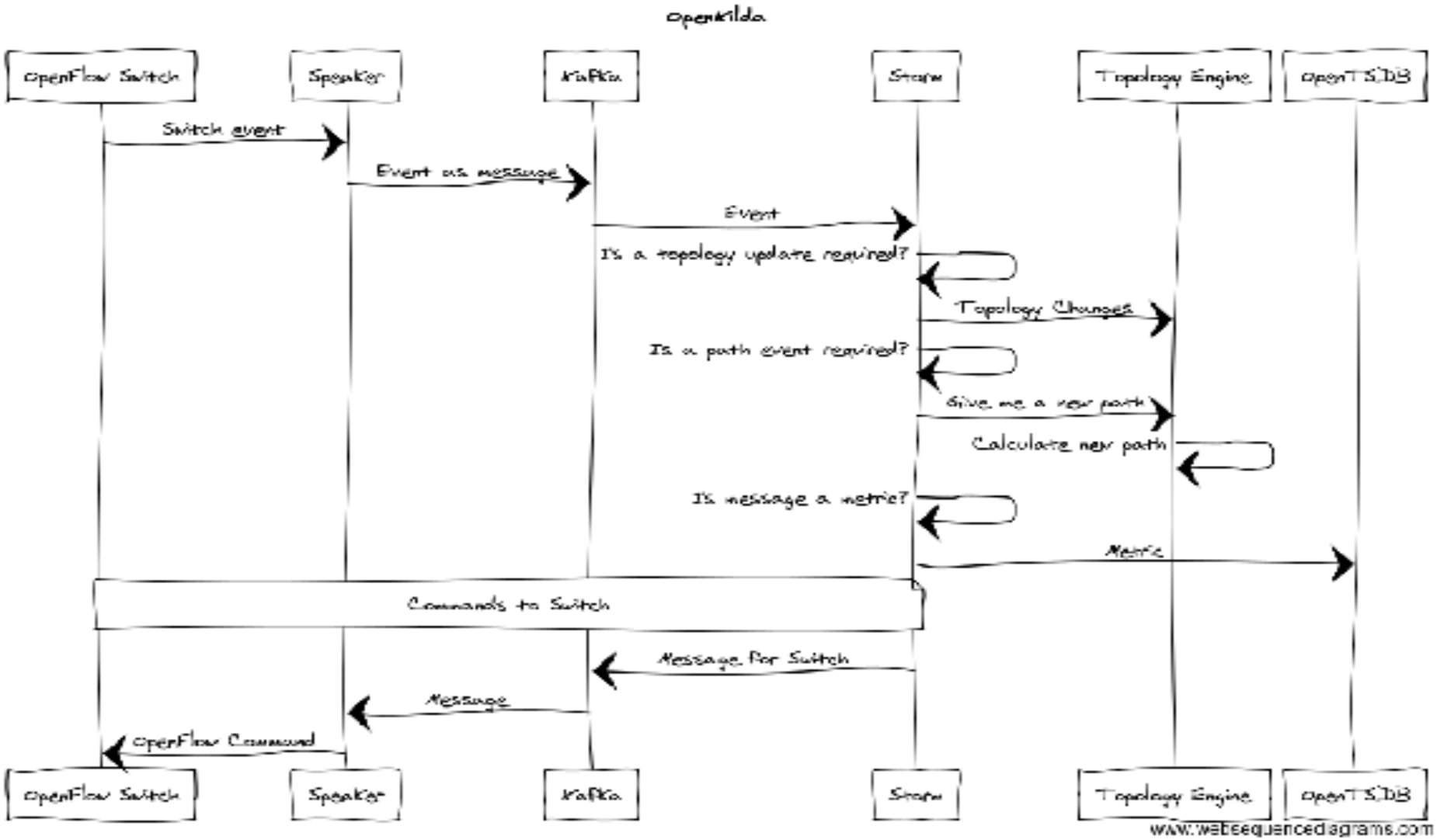
Our Solution

Architecture



TELSTRA

Sequence Diagram



TELSTRA

Current State

Northbound Interface

- Restful
- Create/Modify/Delete Flow
- Push/Pop/Modify VLANs
- List Flows/Switches

Telemetry

- Flow stats
- Port stats
- Switch status

Operational

- Auto-discover network
- Active monitor of ISL with Latency
- Re-Flow when topology change occurs

TELSTRA

How'd We Do?
Based On The Original Objectives

Sub-Second Failover – **NOT YET**

~~Negative Affinity In Path Selection~~

~~Active Latency Measurement on ISL~~

~~End-to-End Latency Measurement on Flow~~

~~Path Selection Based on Latency~~

~~Auto-re-route based on real-time latency/packet loss/jitter measurements~~

Multiple data points for comprehensive end-to-end network state – HALF DONE

Horizontal scale *(achieved in testing)*

~~Number of switches~~ - 10K Switches

~~Number of flows~~ – 16M Flows

Complex match/actions using experimenters – **NOT YET**

~~Stats collections at 1 second intervals~~

~~Self Healing/Optimizing Network~~

~~Zero Touch Controller Deployment/Upgrade~~

Whats Next for Kilda?

Features

- GUI
- Consolidated Northbound API
- Lightweight Speaker
- Documentation

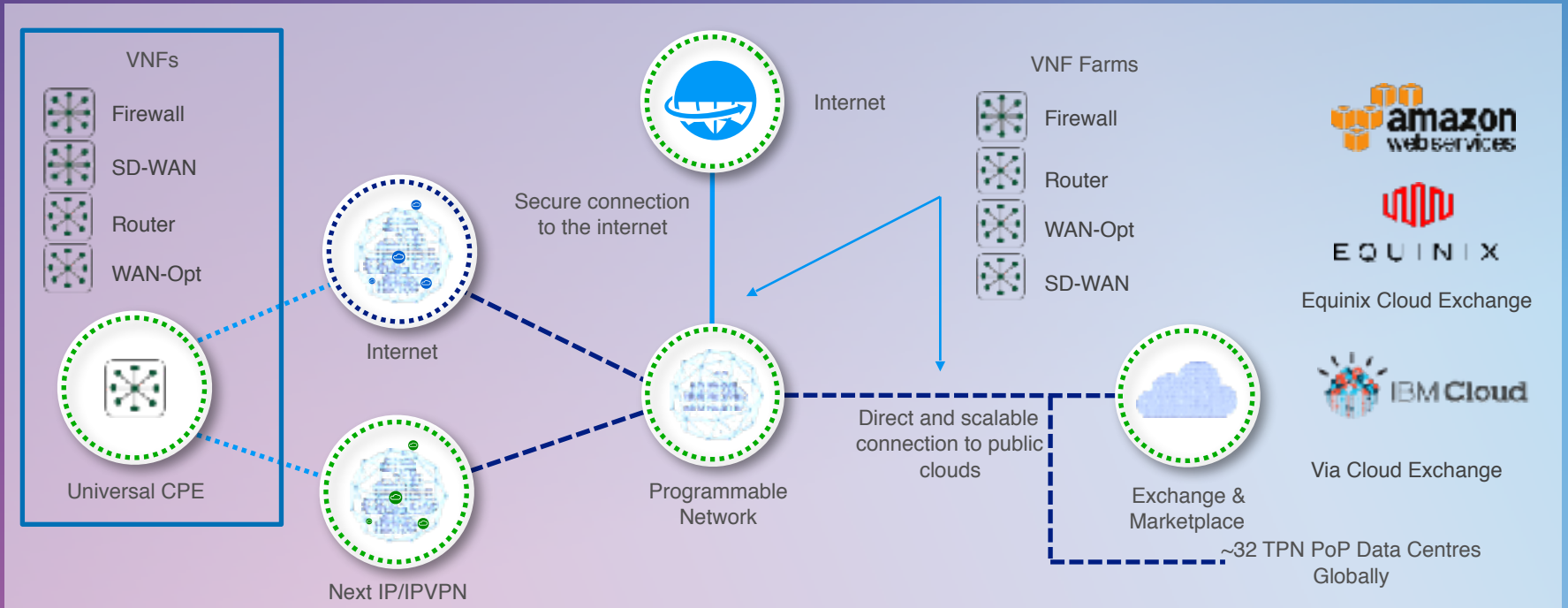
Functionality

- Extend topology event logic
- Complex Match/ Action
- BFD for ISL status
- Fast re-route
- Pre-emptive re-route

Build

- Shorten build time
- Extend build pipeline
- Test in sandbox

What's next for TPN?



Release 1 (April 2017)

R1.0 enables our IPVPN customers to connect to TPN portal and access the TPN capabilities.

Release 2 (July 2017)

R2.0 enables our Next IP customers to connect to TPN portal and access the TPN capabilities.

Release 3 (October 2017)

R3.0 enables our IPVPN and Next IP customers to deploy VNFs on uCPE housed at their premises

New Capability

uCPE hardware device

- Customer can deploy VNFs from marketplace on a uCPE in their branch (Juniper NFX 250)

New virtual network functions

- Juniper vSRX
- VeloCloud SD-WAN
- Riverbed vSteeklhead

Portal enhancements

- Online on-boarding for existing Telstra customers
- Automated retrieval of customer's Telstra VPN and Internet service for information display

Get Involved!

Homepage - `https://github.com/telstra/open-kilda`

(git clone https://github.com/telstra/open-kilda.git)

Native Development Environment

```
# clone your GitHub fork
> make build-latest
> docker-compose up
```

Linux Based Environment

```
> vagrant up
> vagrant ssh
> ssh-keygen -t rsa -C your_email@example.com
# update your GitHub fork with ssh key
# clone your GitHub fork
> make build-latest
> docker-compose up
```


TELSTRA

Thank you