Xantaro

SERVICE INTEGRATION

# TRAFFIC ENGINEERING SR/SPRING NETWORKS USING OPEN SOURCE TOOLS

## SR/SPRING SAGA PART II – RETURN OF THE LABELS

# Who Am I?

- Tobias Heister
- Solutions Architect
  - Technical Pre-Sales
  - Vendor MGMT for Focus Vendors
  - Solutions Engineering and Partner Evaluation
  - XT3LAB

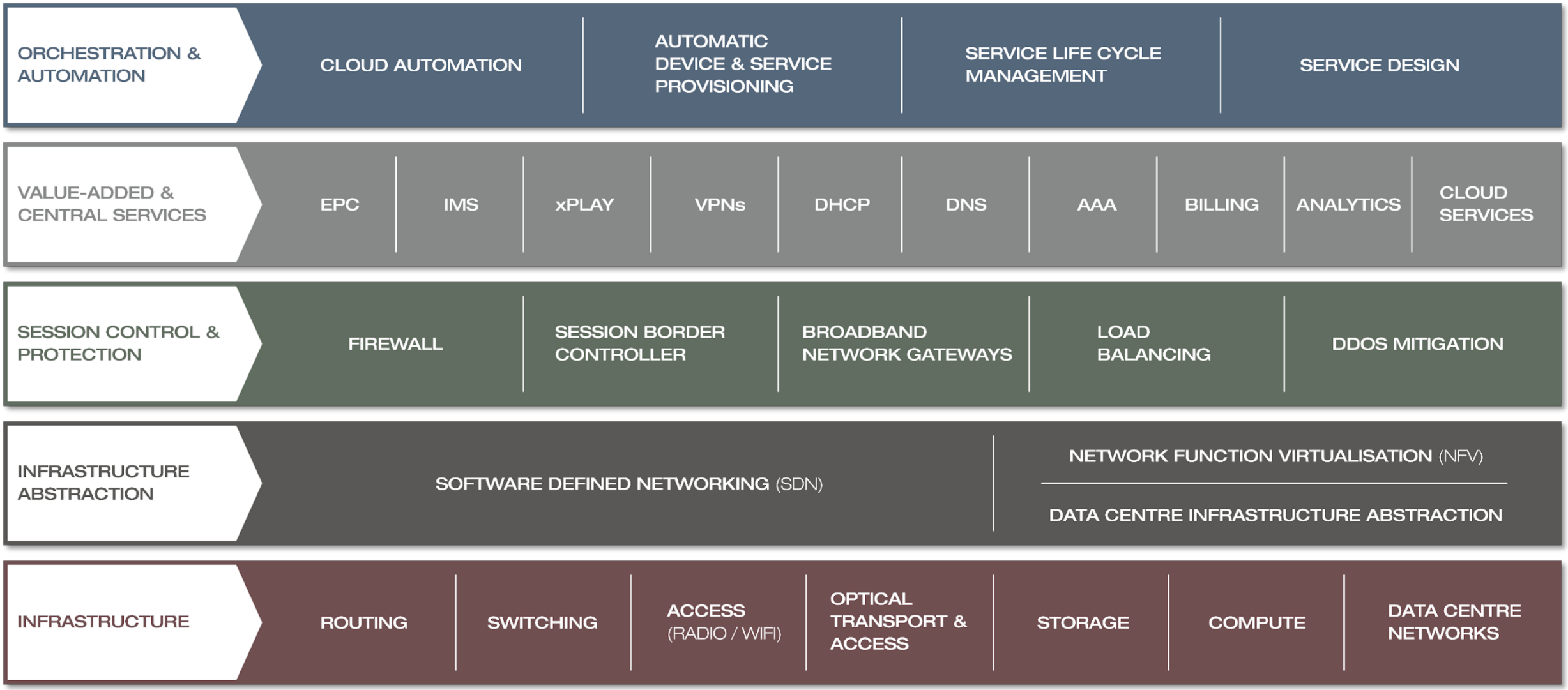
We are here today

- 3y Deutsche Telekom – Carrier and DC
- 8y Host Europe - Hosting Provider, Network centric Roles (Engineering, Managed Services)
- 4y Xantaro – Professional Services for 1y then SOLAR

# The Obligatory Marketing Slide –

## Covering All Network Layers

Xantaro SERVICE INTEGRATION



# Agenda

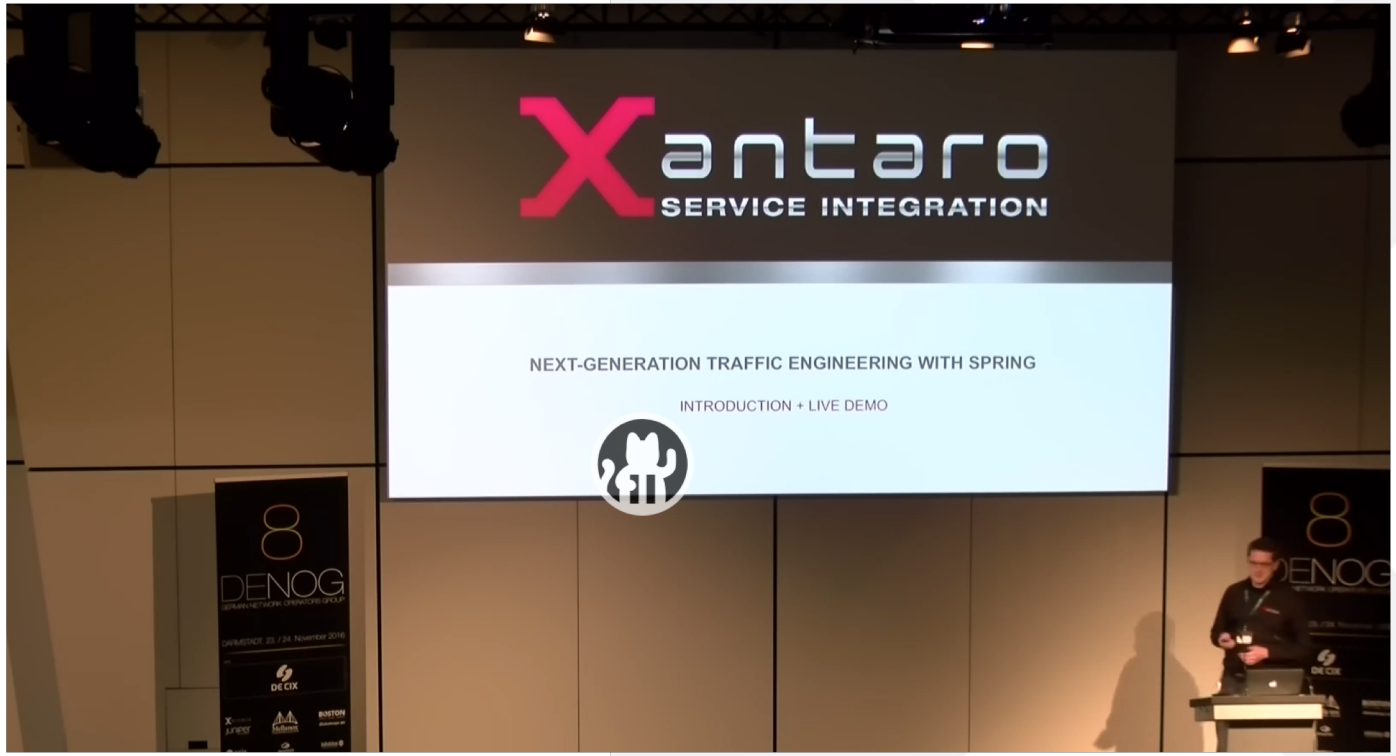
- Short Recap on SR/Spring
- SR Controller
  - Overview
  - Architecture Details
- High Level Walkthrough
- Building Blocks
- „Roadmap“

# A NOT SO LONG TIME AGO IN AN AUDITORIUM NOT SO FAR AWAY

NEXT-GENERATION TRAFFIC ENGINEERING WITH SPRING – DENOG 8

# DENOG 8 - SR - The Label Menace

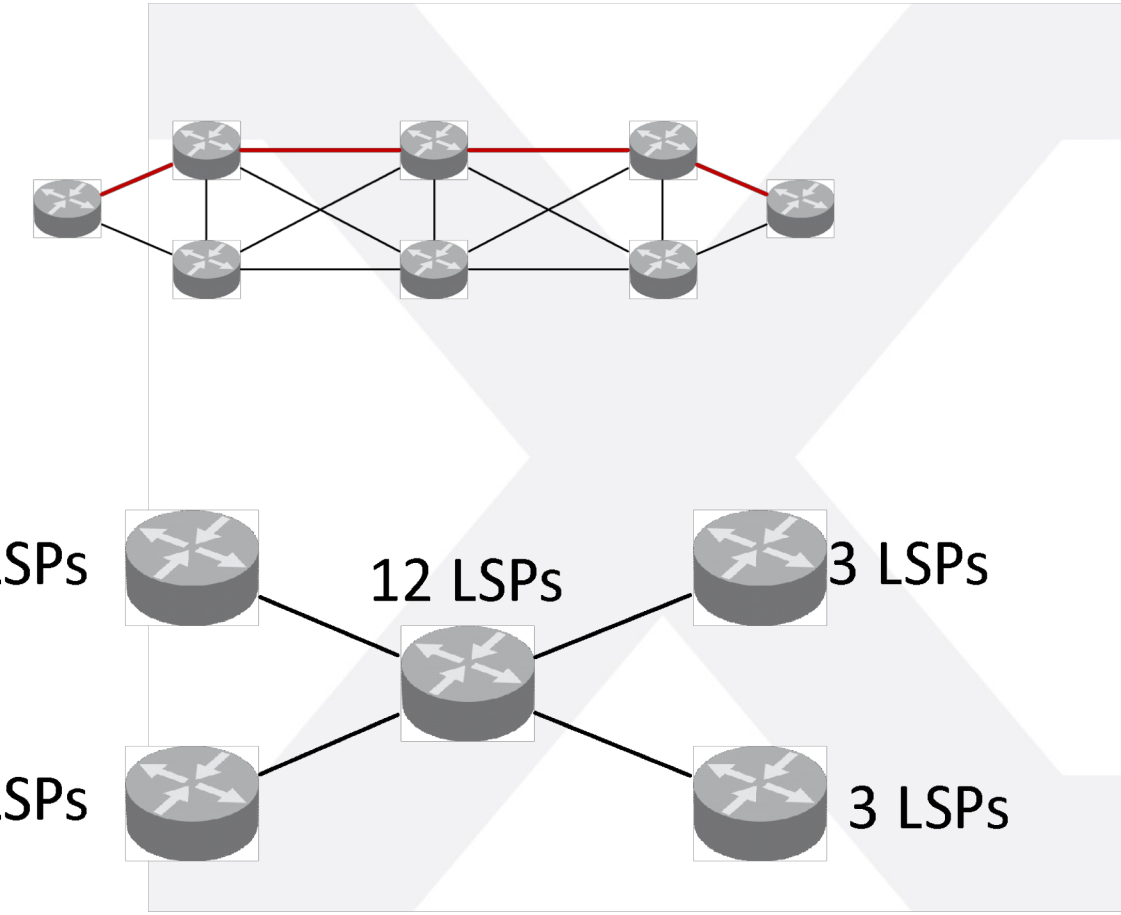
- This Talk builds on top of Sebastians talk from DENOG 8
- The next slides are a very brief summary of key topics
- Watch the recording for more details




https://media.ccc.de/v/denog16-4008-next_generation_traffic_engineering_with_spring

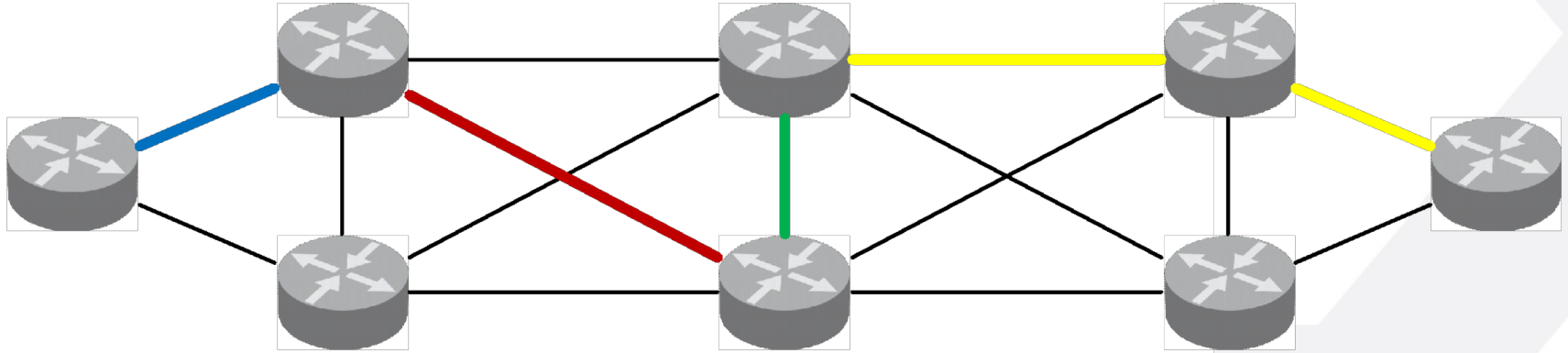
# Label Distribution – The Old World

- Basically two choices within an autonomous system
- LDP
  - provides any-to-any connectivity
  - less control plane overhead
  - no traffic engineering capabilities, shortest path only
  - Can provide fast failover via IP-LFA
- RSVP
  - explicit signalization of LSPs
  - can provide any-to-any connectivity if configured
  - provides traffic engineering capabilities
  - creates states in the network for each LSP
    - especially problematic on P Routers
  - FRR was big selling point -> 50ms claim



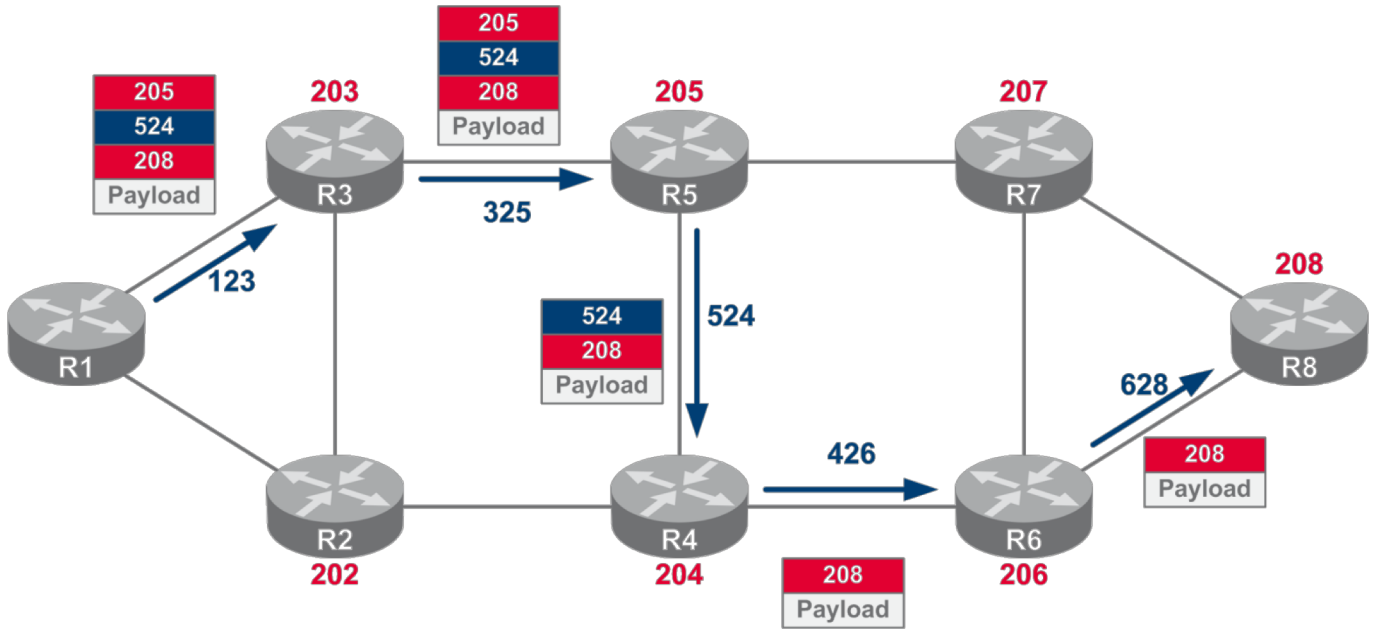
# SPRING – A new Hope

- New approach standardized in the IETF
  - **S**ource **P**acket **R**outing **i**n **N**etworkin**G**
- Formerly known as Segment Routing (SR)
- SPRING envisions the network as a collection of topological sub-paths – also called segments.
- Create any kind of desired path by stitching one or more individual segments
- All of that by making use of existing IGP IS-IS or OSPF (BGP Draft exist as well)



# Traffic Engineering - NG

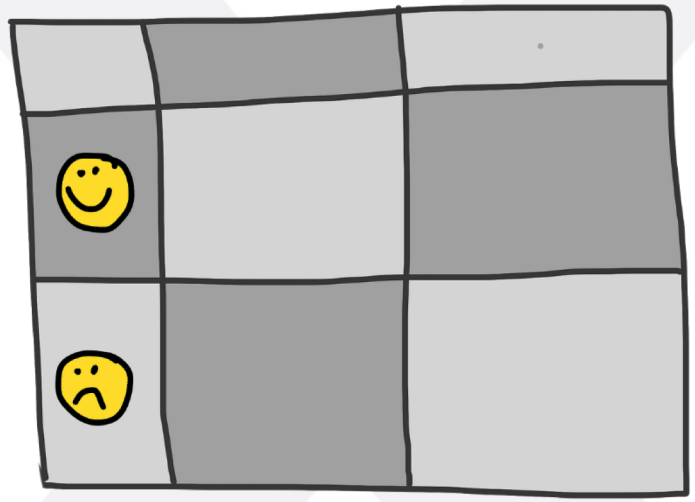
- Any path can be expressed using a combination of IGP prefix (node) segments and adjacency segments at the ingress Router




- Somebody has to know which labels to use where and which labels to push in ingress

# There is no free lunch

- Advantages
  - Policy state is in the packet header, not on the transit routers
  - There is no midpoint state (n^2 scale in RSVP-TE)
  - There is no extra protocol (RSVP-TE, LDP) – integrated in IS-IS, OSPF or BGP
  - Can be drop in replacement for LDP
  - Only few announced segments are required to achieve TE
  - Segment-routing traffic engineering supports distributed or centralized computation
- Disadvantages
  - Next to no policy support on current NOS
    - Some stuff in very recent JunOS and IOS-XR
  - TE needs Policy
    - Distributed approach is a challenge
    - Central/Control Based approaches are usually proprietary and costly



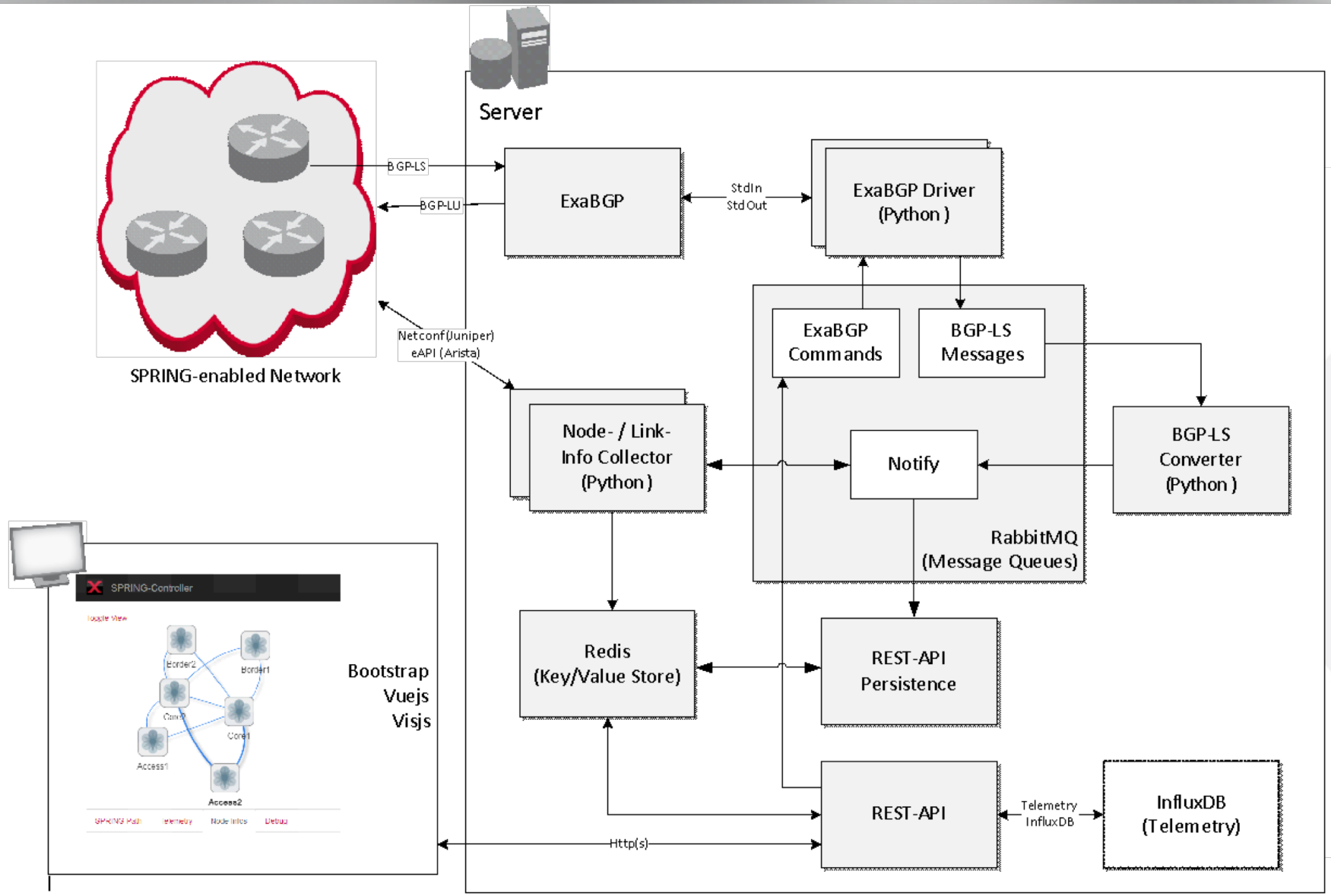
# OPEN SOURCE TO THE RESCUE!

LETS SEE HOW FAR DUCT TAPE CAN TAKE US – DIY SR/SPRING CONTROLLER

# Shoutout to the Creator

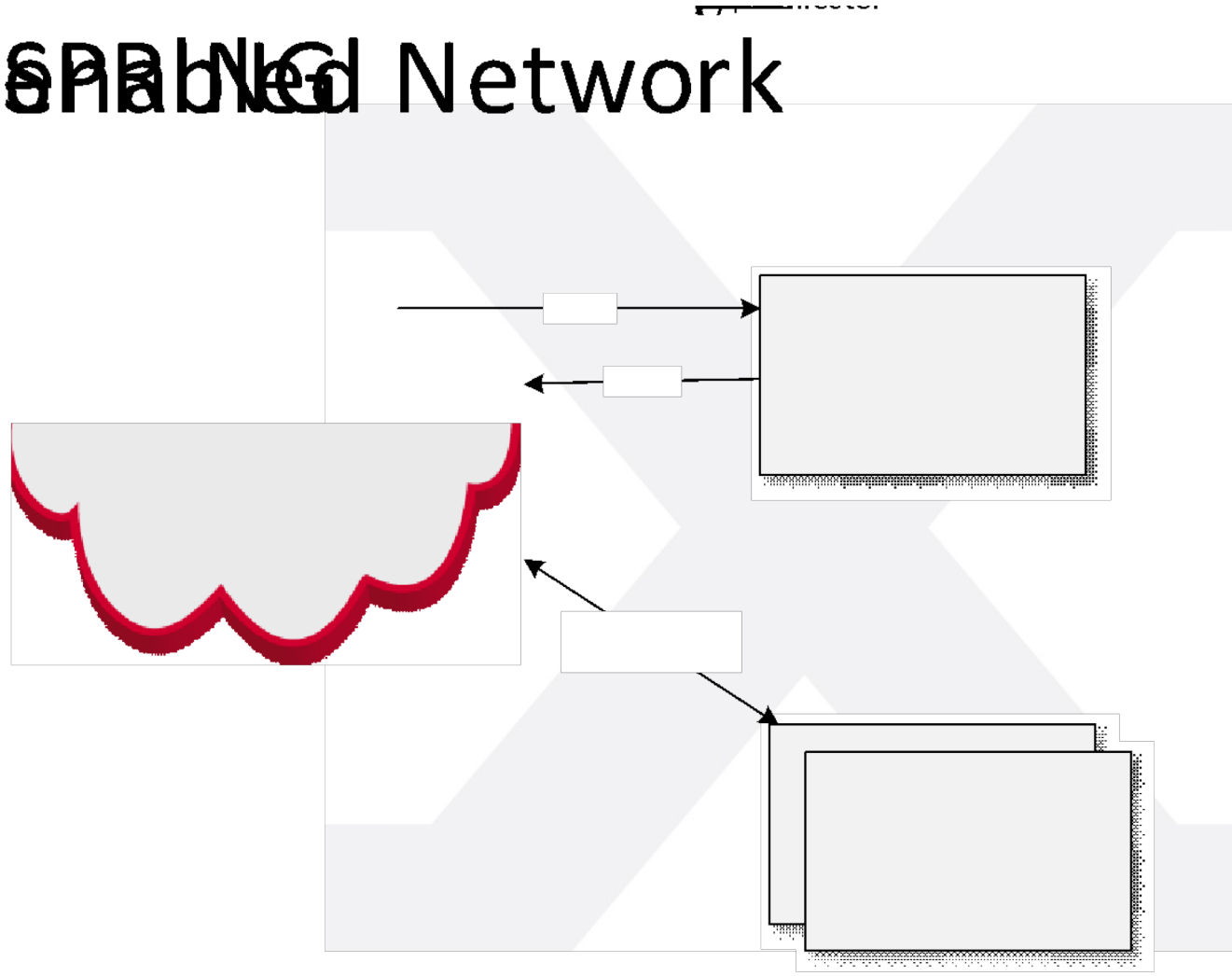
- All of the implementation done by Xantaro Consultant - Markus Vahlenkamp
- Got inspired by our Spring/SR Talk two years ago
- Was surprised by the lack of existing Open Source Solutions for the problem
- Does DC and IP/MPLS Projects as well as Automation stuff during his normal working hours
- Wrote most of the stuff in his free time

# The Scary All in One Architecture Picture



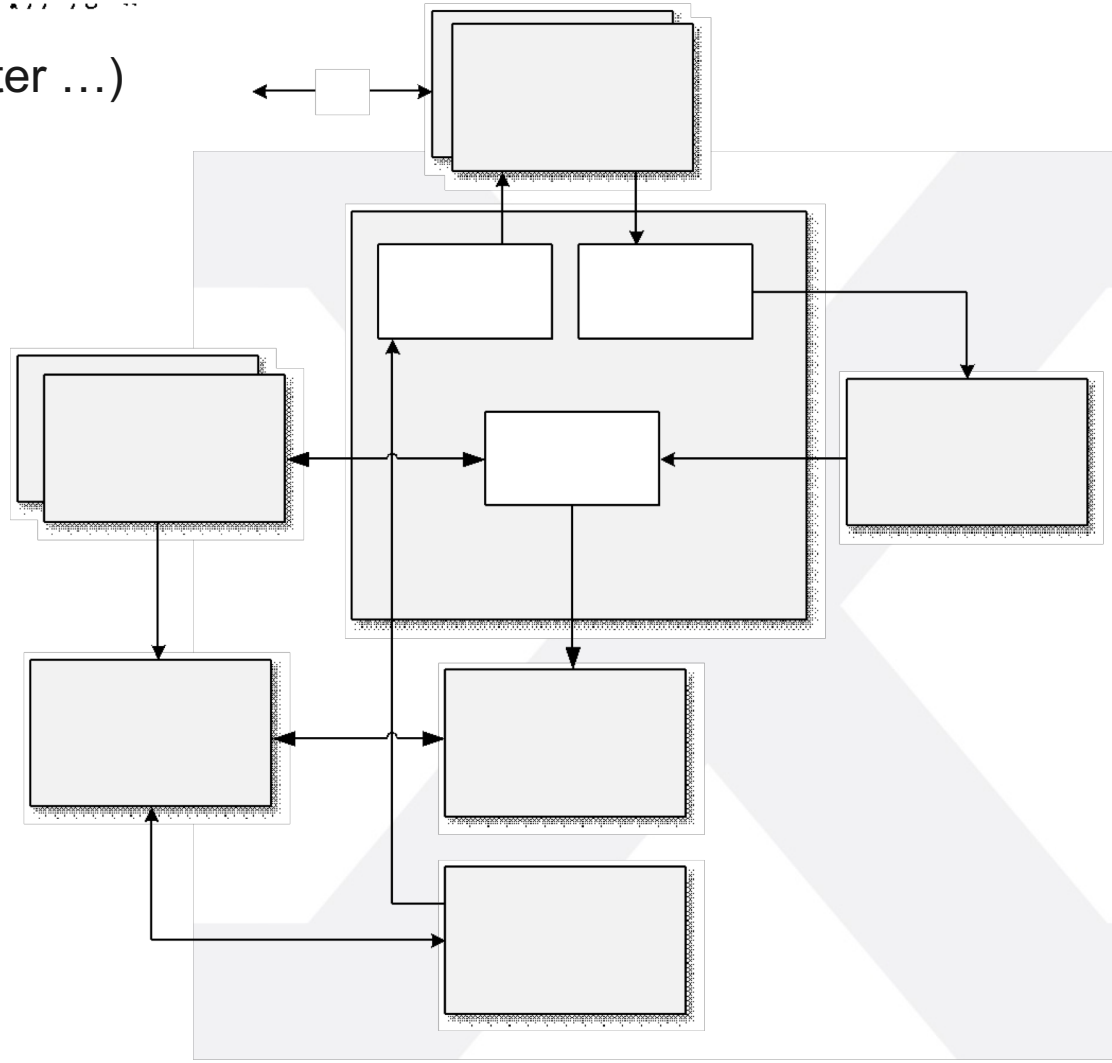
# Lets zoom in – Network Interaction

- BGP-LS with one Network Element to extract
  - Link State DB
  - Label Information
  - Topology
- Netconf or vendor API (e.g. eAPI) for
  - Additional Data
  - Adding some flavour to visualization
- BGP-LU to program prefixes with Label stack



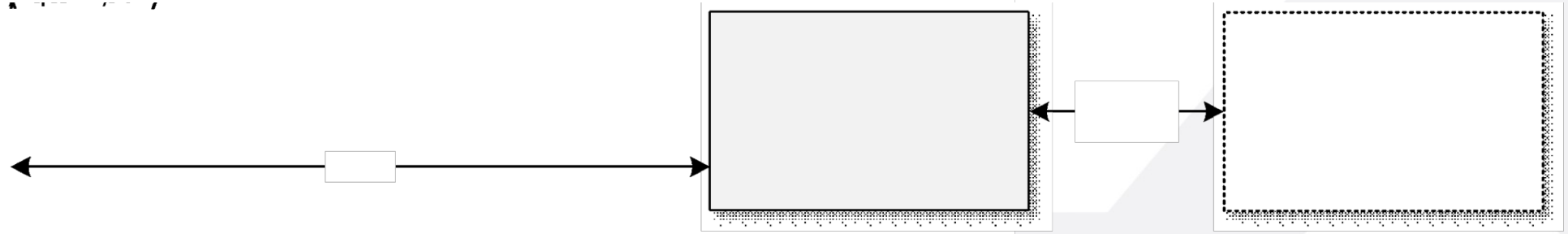
# Lets zoom in – Core Infras

- Various python modules (Exa-BGP Driver, BGP-LS Converter …)
  - Do their job
  - Publish to Message Q(s)
  - Subscribe to Message Q(s)
- Driver converts
  - clear text to JSON
  - ingest commands from API -> BGP-LU
- BGP-LS Converter parses JSON Messages
- Node/Link Collector gathers additional info
- Storage in Redis Instance
- Exposed via REST-API



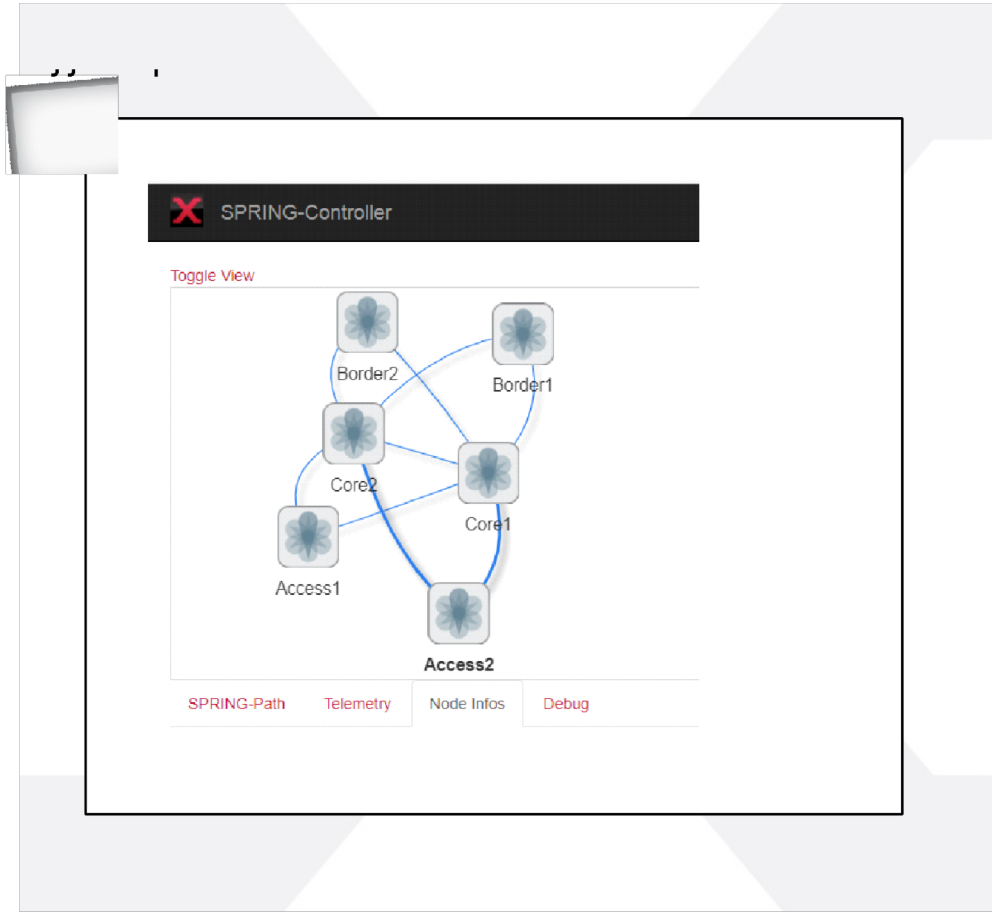
# Lets zoom in – external Interaction

- All Interaction is done via a REST API
- We integrated a TSDB for Telemetry storage and visulization
  - The REST API is used to query the TSDB and integrate the data in to the Web UI
  - Telemetry is not part of this talk

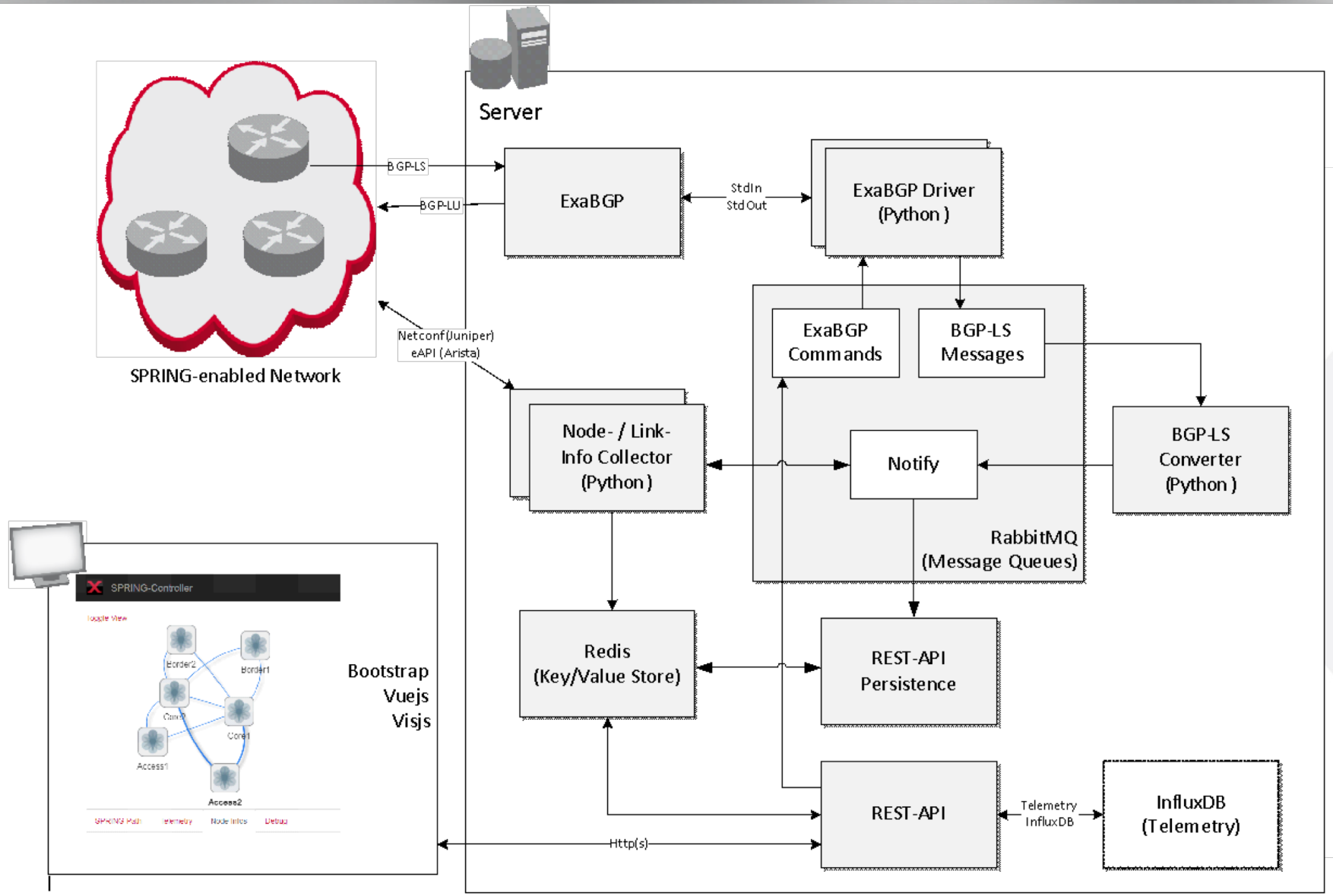


# Lets zoom in – Web UI

- Simple Streamlined Web-UI
- Visualization of the Network based on Topology Info
- Visualization of network utilization based on Telemetry Data
- Interface to program a TE network path
- For the display part of the Telemetry data we rely on Grafana

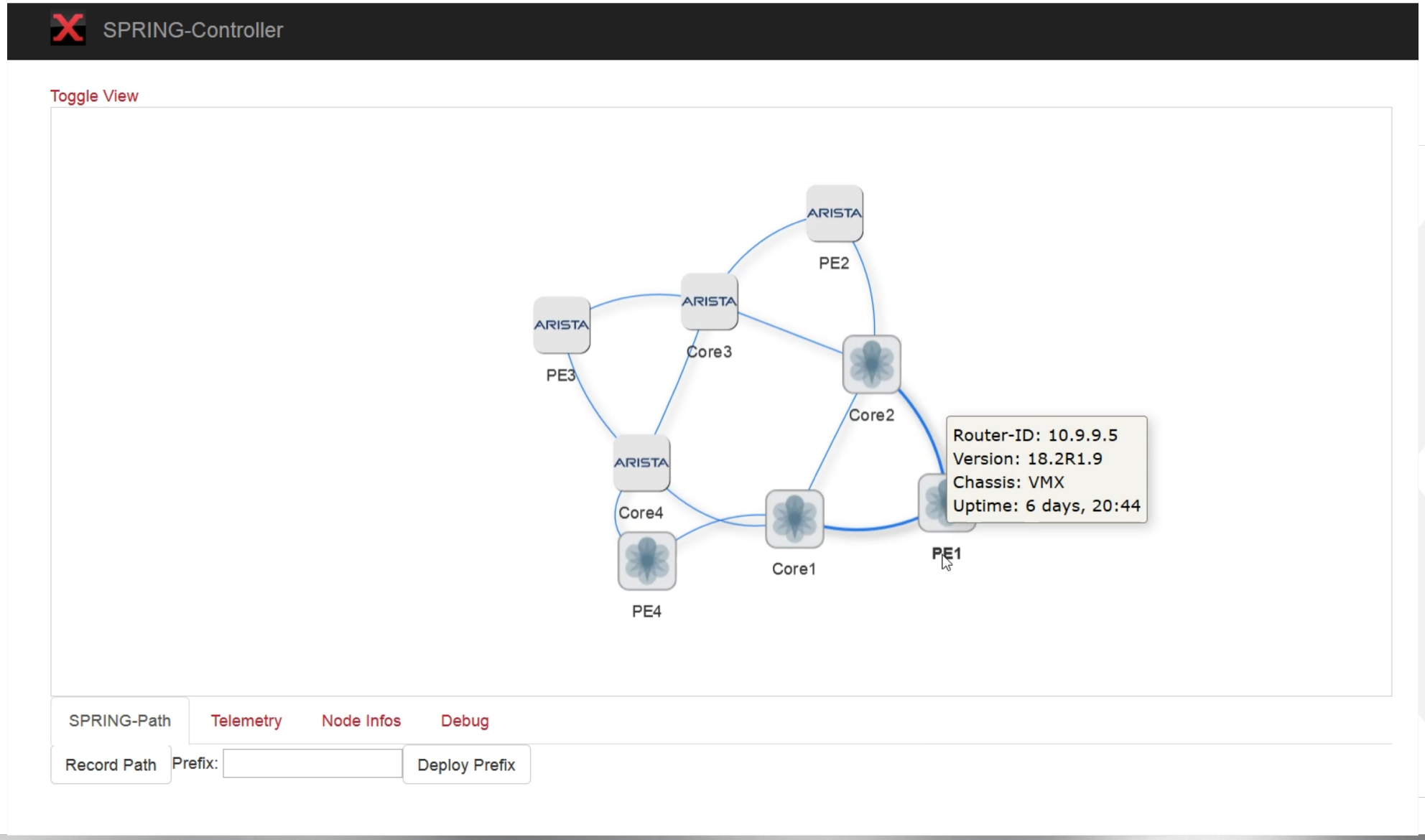


# The Scary Architecture Picture - Revisited

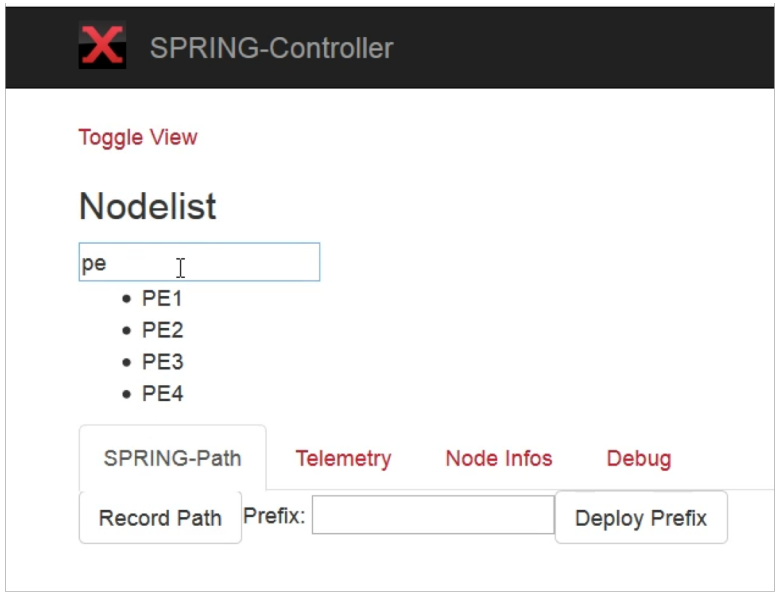


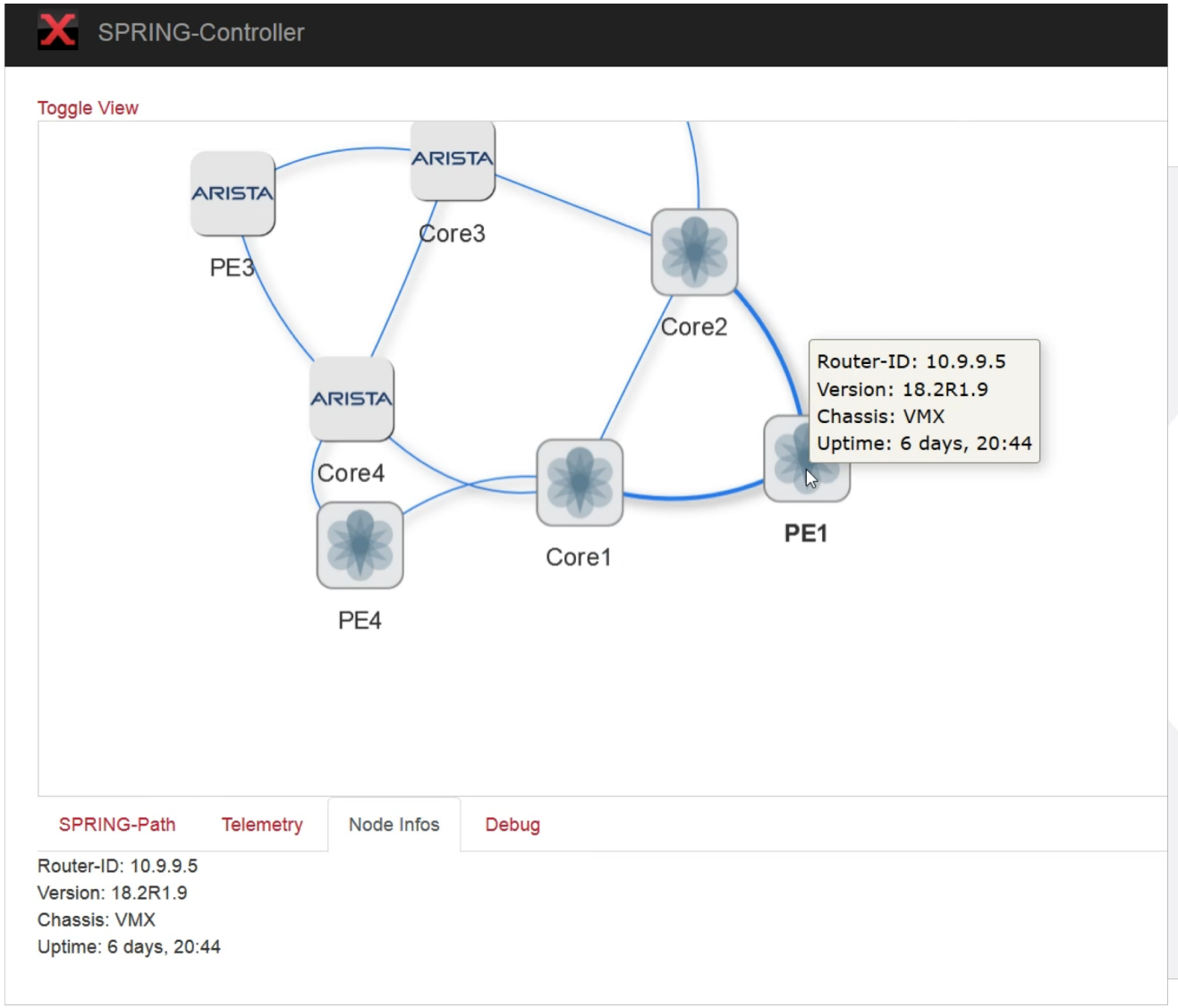
# HIGH LEVEL - WALKTHROUGH

# Topology Overview

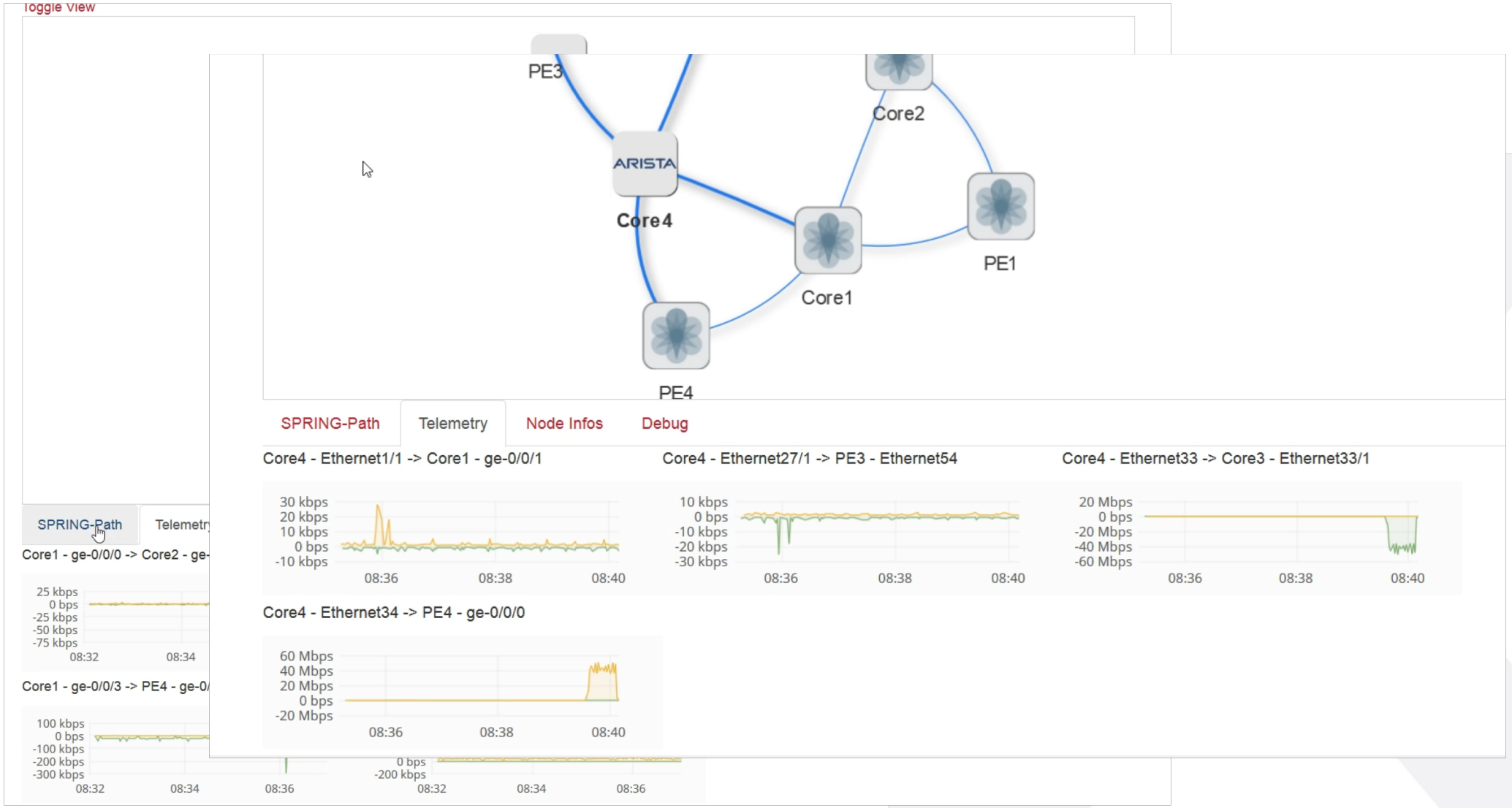


# List view and filtering

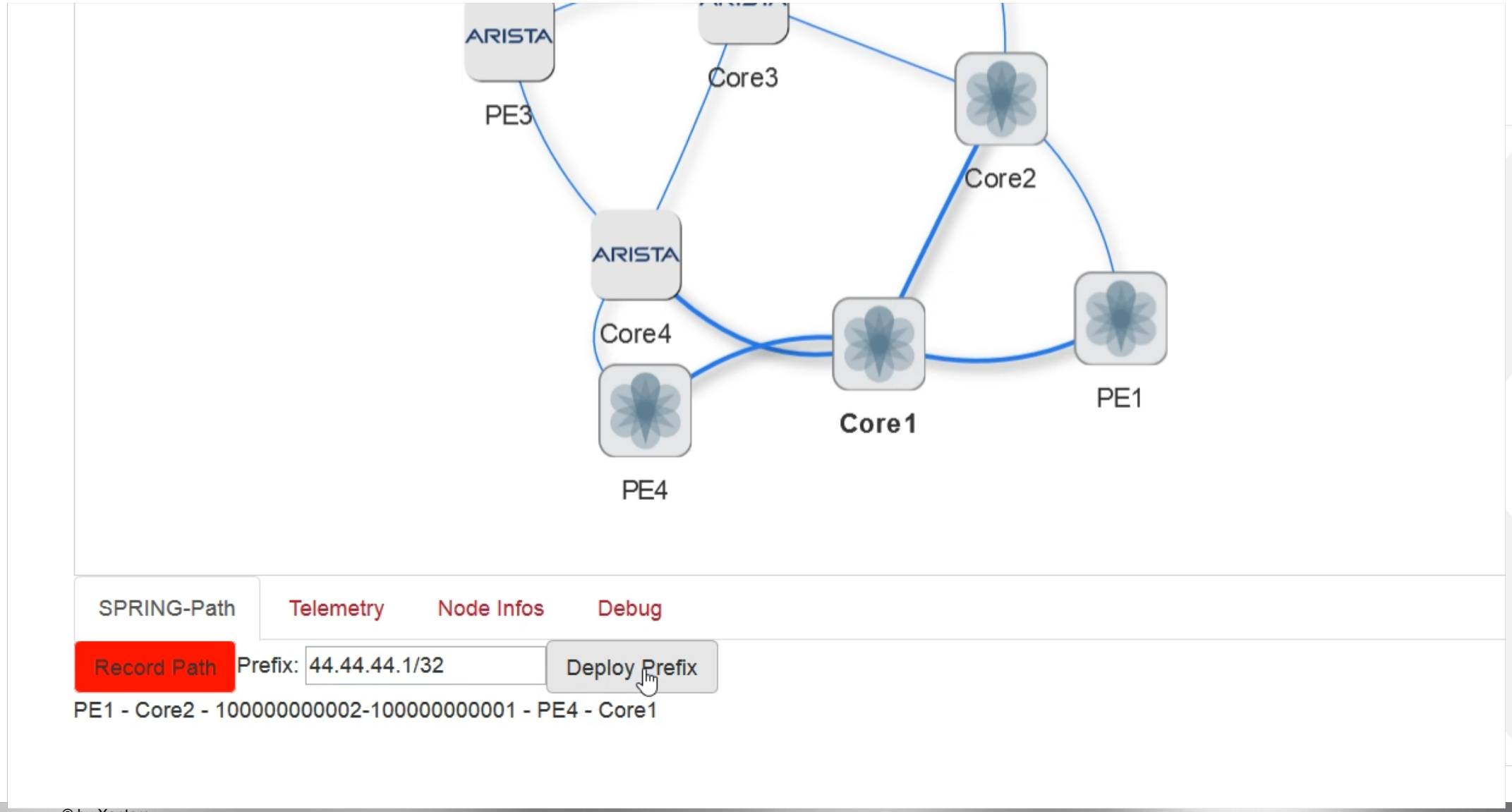


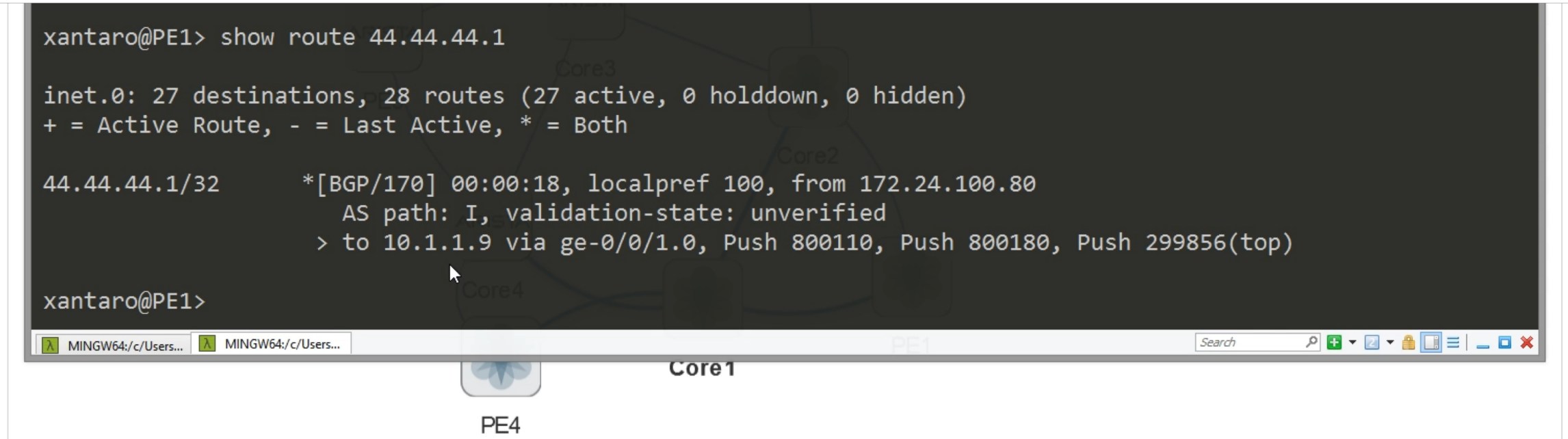
# Telemetry Integration



# Deploy TE path



# TE Path - Live View




SPRING-Path | Telemetry | Node Infos | Debug

Record Path | Prefix: 44.44.44.1/32 | Deploy Prefix

Route: neighbor 10.9.9.5 announce route 44.44.44.1/32 next-hop 10.9.9.2 label [ 299856 800180 800110 ];

Status: success

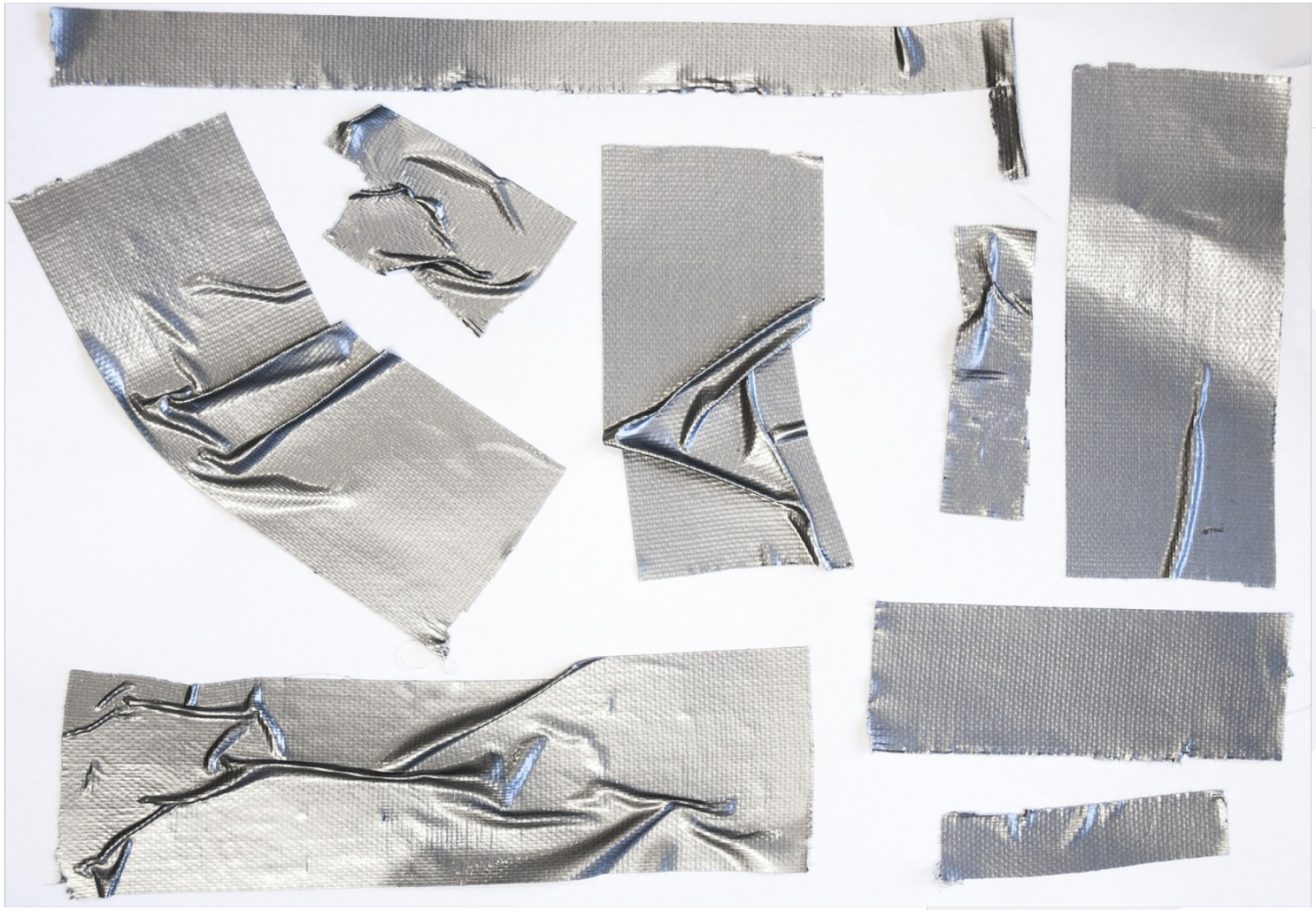
# BUILDING BLOCKS

## SOME ASSEMBLY REQUIRED

# Components

- Docker
  - Containerization of components
- ExaBGP
  - BGP daemon – BGP-LS, BGP-LU
- RabbitMQ
  - Message Queuing, RPC and notifications
- Redis
  - Shared Data Storage
- Python Flask
  - Webserver UI / REST-API
- Junos-pyez
  - Python Netconf for Juniper JunOS devices
- eAPI
  - REST-API for Arista EOS devices
- OpenConfig / Juniper Telementry Interface
  - Telemetry interfaces
- InfluxDB
  - Telemetry database
- Grafana
  - Telemetry visualization
- Javascript Frameworks
  - Vuejs
    - Javascript UI framework
  - Visjs
    - Graph framework
  - JQuery
    - Transition effects and AJAX calls

# And of course lots of Duct Tape



# Features

- **Implemented**
  - BGP-LS – Link, Node and Prefix LSAs
    - announcements and withdrawals
  - SPRING Path Deployment
    - multi-peer BGP-LU
  - Telemetry integration
    - Junos Telemetry Interface (JTI)
    - OpenConfig Telemetry for Arista
  - Multi-vendor link and node information retrieval
    - Interface descriptions, Chassis type
    - Juniper + Arista supported, more to come

- **“Roadmap”**
  - Persistent SPRING Paths
    - Paths will be redeployed after a restart
  - Server-Side Events
    - Streaming of topology events towards UI
  - Redundant ExaBGP instances
  - Per link telemetry visualization

# But Why?

- Because We Can! Also nobody else did it so far (at least when we started the internal project)
- There is no real customer project or customer interest …. Yet!
- SR/Spring will happen at one point – stay ahead of the curve

# Questions?