Reformo NETWORKS

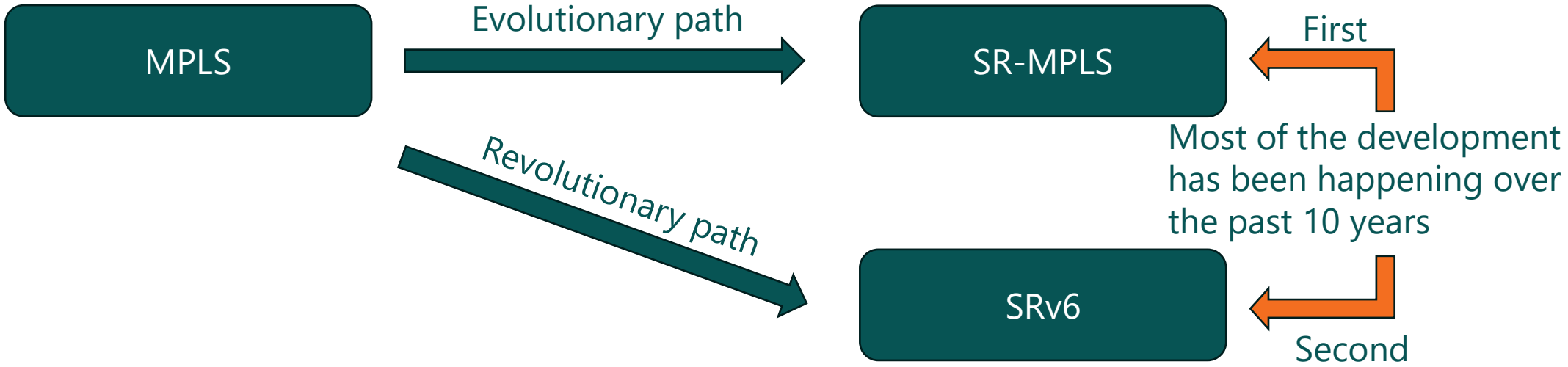
# SRv6 – a Short Introduction

Aki Anttila (@reformo.fi)

# What is SRv6?

# SRv6 background

- MPLS is good, well-known, well-supported, interoperable…
  - …and also complex, inadequate TE capabilities, depth of the label stack might be an issue, (is old and dusty)



# Current state of SRv6 globally

100's of millions of subscribers in SRv6 networks

Heavy usage in Asia (which is IPv6 anyway)

>100 operator deployments

Vendor support is strong

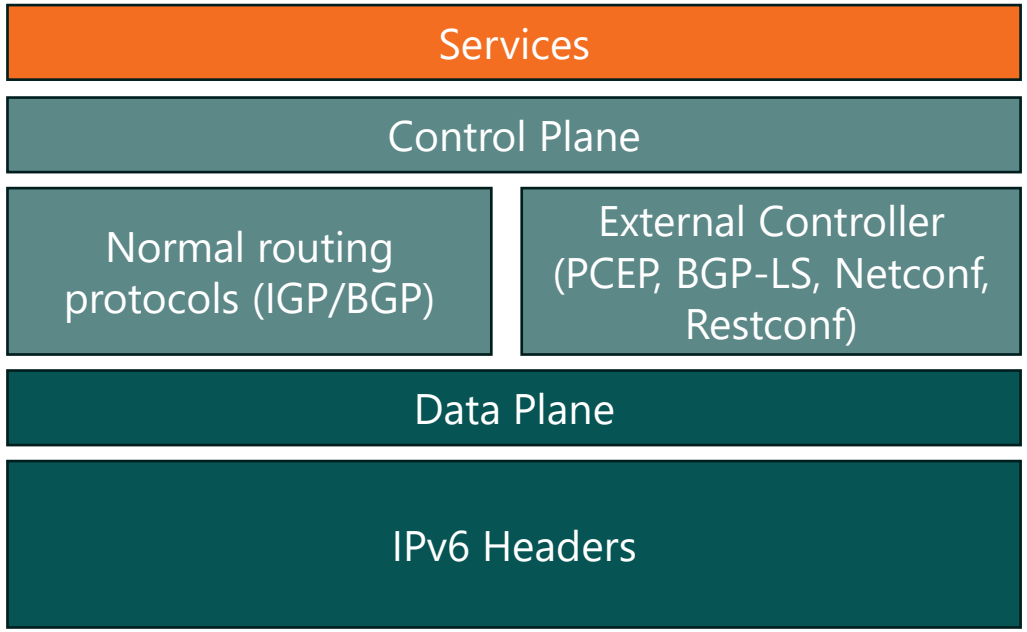

IETF standards are ready or about ready

Open source community is strong

# Main RFC documents for SRv6

# So…how does it work?

- The underlay network is pure IPv6
  - "Standard" routing mechanisms, but let's stick to IS-IS (OSPFv3 is quite ok)
  - External sources of policies can be used
  - Aggregation is available and recommended to minimize the amount of control information and path calculations
- The overlay network is whatever is needed
  - BGP L3 VPNs (VRFs)
  - BGP-EVPN L2s or L3s
  - Service node insertion



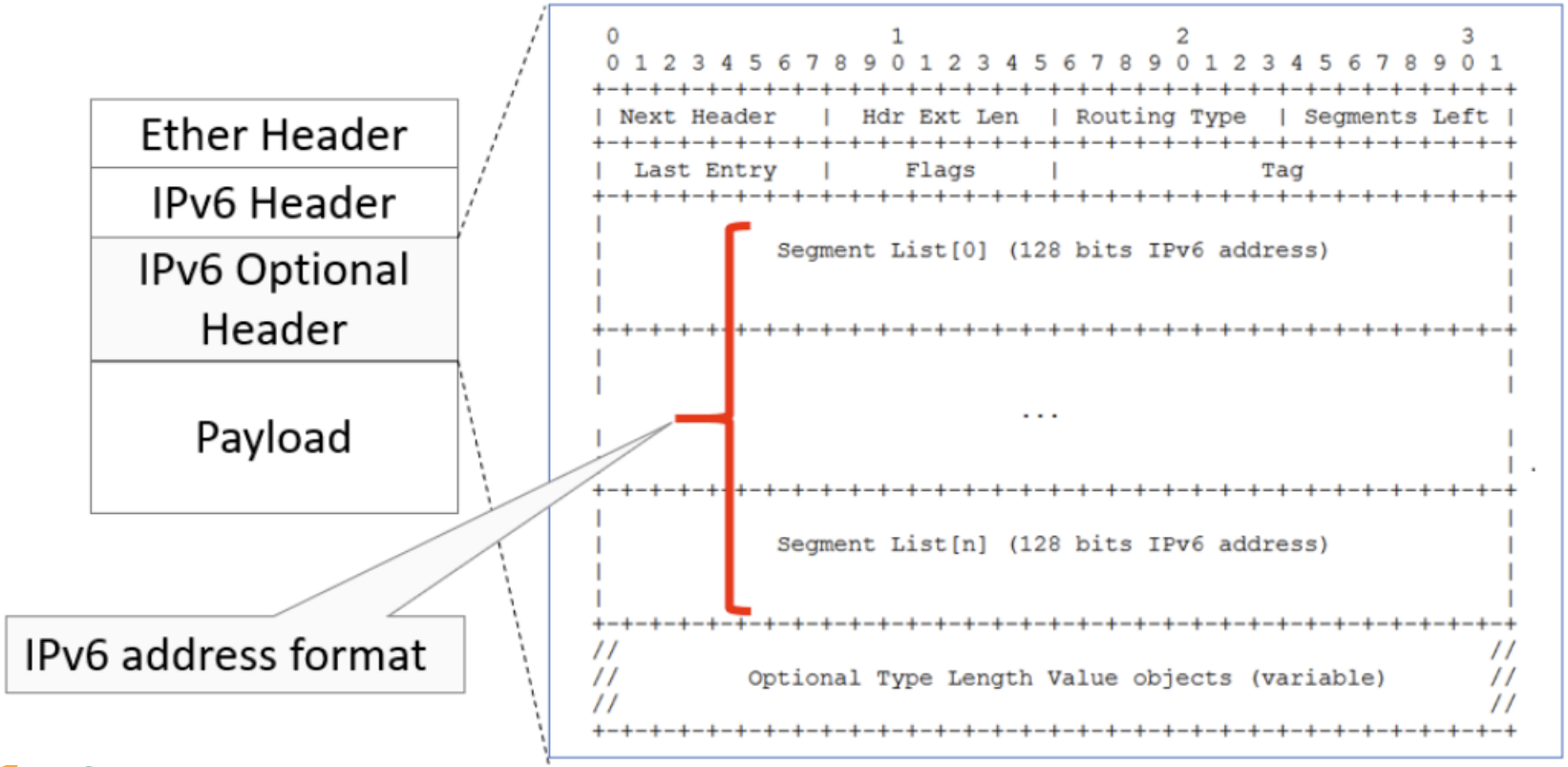

IPv6 provides the capability to do *network programming* (i.e., which nodes should be traversed, which services should be used) through SIDs (Segment Identifiers) that are presented through IPv6 extension headers.

# SID? SID-Prefix? Locator?


- SID-Prefix is the IPv6 address space that is used for SRv6 purposes
  - Unique Local Addresses (ULA) are recommended to prevent leaking
- Each node is allocated one or more locators depending on the needed services
  - These locators are “normal” IPv6 addresses and hence routable and aggregetable within the IGP domain
- Segment Identifier contains the instruction (function) within the locator to tell the device what to do with the traffic
  - E.g. SID <locator>::000A means “forward via VRF-table 1”
  - Note that the first instruction is the normal IPv6 header DA
    - And this is often sufficient for services to work!

# IPv6 header

- The official name is “Segment Routing Extension Header, SRH”




Each SID (Segment List) is 128bits = 16bytes

In case of 6 segments, overhead is 6*16+8 bytes = 104 bytes

Within iMix traffic, this is about 30% additional overhead!

Therefore uSID!

# uSID saves space!

- uSID is a compressed SID that does not include locator bits
  - “default size” is 16 bits, other sizes can be used
- Normal IPv6 header DA holds up to 6 uSIDs
  - if more is needed, then SRH must be used

Outer DA: 2001:db8:0001:0002:0003:0004:0005:0006

uSID1 uSID2 uSID3 uSID4 uSID5 uSID6

Outer SRH: 2001:db8:0007:0008:0009:0010:0011:0012

uSID7 uSID8 uSID9 uSID10 uSID11 uSID12

# What can be done with SIDs?

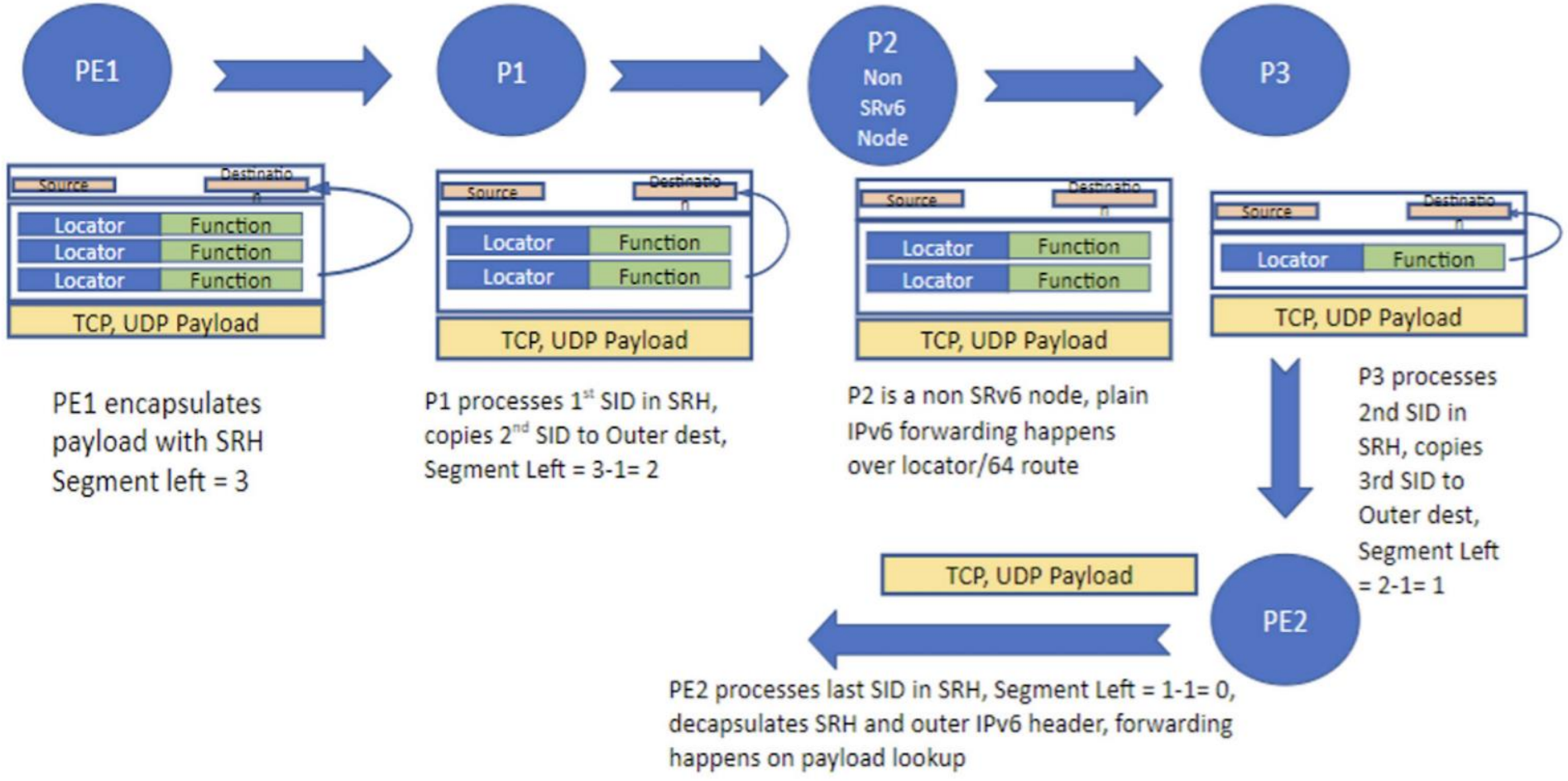
Since SID is a key to *network programmability*, everything can be programmed with it.

(However, do not mix with Netconf/Restconf/JSON etc. programmability)

Network level functions

Customer facing functions

# An example of using SIDs



# SRv6 policy behaviors

- SRv6 has two different behaviors
  - Headend (H.X) behaviour that happens on the *source* node
  - Endpoint (End.X) behaviour that happens on the *end* node
- Headend behaviors can be used to provide backbone services
  - E.g. H.Encaps (SR Headend with encapsulation in an SR policy), used with TI-LFA
- Endpoint behaviors are used to steer traffic in the edge of the SRv6 domain
- “MPLS-like” additional behaviors are also present – Penultimate Segment Pop (PSP), Ultimate Segment Pop (USP) and Ultimate Segment Decapsulation (USD)

# Examples of End behaviors

| Behavior | Explanation (Per RFC 8986) |
|---|---|
| End | Endpoint. The SRv6 instantiation of a Prefix-SID. |
| End.X | Endpoint with L3 cross-connect. The SRv6 instantiation of an Adj-SID. |
| End.T | Endpoint with specific (IPv6) table lookup. |
| End.DT4 | Endpoint with decapsulation and specific IPv4 table lookup, e.g., IPv4-L3VPN (equivalent to per-VRF VPN label). |
| End.DT6 | Endpoint with decapsulation and specific IPv6 table lookup, e.g., IPv6-L3VPN (equivalent to per-VRF VPN label). |
| End.DX4 | Endpoint with decapsulation and IPv4 cross-connect, e.g., IPv4-L3VPN (equivalent to per-CE VPN label). |
| End.DX6 | Endpoint with decapsulation and IPv6 cross-connect, e.g., IPv6-L3VPN (equivalent to per-CE VPN label). |
| End.DX2 | Endpoint with decapsulation and L2 cross-connect, e.g., L2VPN use case |

# SRv6 services in the backbone

# Examples of the network-level possibilities

Topology Independent Loop Free Alternate (TI-LFA)

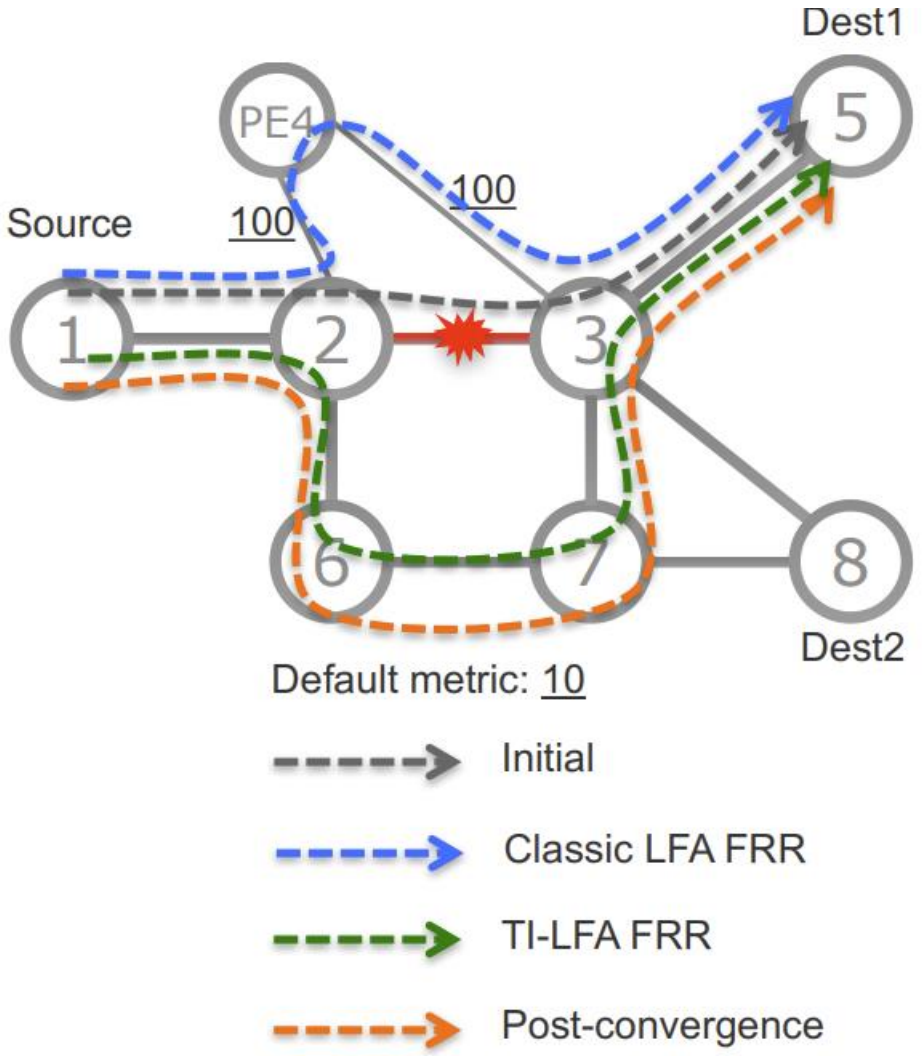
Flexible Algorithm (FlexAlgo)

Microloop Avoidance

Performance monitoring for link latency

# Example – TI-LFA

- TI-LFA provides 100% coverage with 50 msec link and node protection through the usage of post-convergence path
  - Prevents suboptimal routing
  - Prevents transient congestion
  - Is computed automatically by IGP
  - Can be incrementally deployed
- Why now?
  - SR(v6) provides network programmability to steer traffic through post-convergence path LOOP FREE



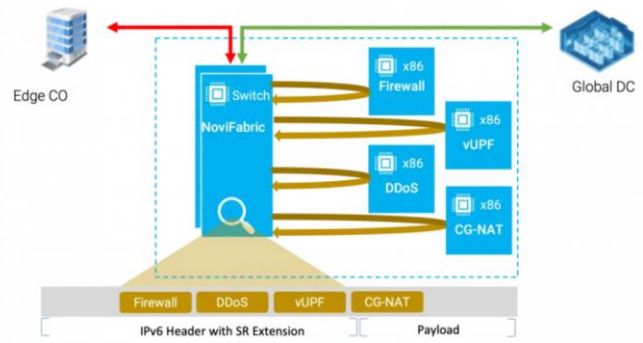
# Example – FlexAlgo (i.e. “slicing” the network)

- FlexAlgo can be used to provide paths with different properties
  - Minimal delay, high BW, avoiding certain links etc.
  - Each constraint forms own topology from the base network
- There are 256 FlexAlgos of which
  - 0 is the traditional IGP/SPF
  - 1-127 are reserved by IANA
  - 128-255 can be used by operator-defined constraints
- RFC 9350 defines extensions for IS-IS and OSPF for calculating constraint-based paths

# SRv6 services for the customers

# Flexible selection of services

- Traditional network services can be implemented on top of SRv6
  - BGP-based L3 VPNs (traditional RFC 2547bis) constructs
  - BGP-based EVPNs (as defined in RFC 7432) constructs
  - These are defined in RFC 9252 (BGP Overlay Services Based on Segment Routing over IPv6 (SRv6))
- In addition, service elements can be inserted (service-chaining)
  - Based on IETF's Service Function Chaining architecture (RFC 7665)
  - SRv6 SID is used as the Network Service Header (NSH) to steer traffic to service functions



Reformo NETWORKS

# That's it!
# Q?
# Email: aki.anttila@reformo.fi OR
# Call: 040-7591631
# Or sit with me at the dinner table!