The TEAS Network Slicing Design Team

Status Report at IETF-106

Design Team Setup

Task: ”Develop a framework for providing Network Slicing using IETF TE technologies. The expectation is for IETF technologies such as IP, MPLS and GMPLS, to be used to create specific, isolated, and managed network instances.”

Main challenge: Ability to focus on the TEAS-relevant part of a very broad & diffuse topic (incl. Marketing Words)

Mode of operation:

- Team preparing proposals for the WG
- Open calls, archives & results
- Draft authorship based on contributions, not membership

Members

- Aijun Wang
- Sergio Belo
- Dong Jie
- Eric Gray
- Geng Xuesong
- Jari Arkko (Lead)
- Jeff Tantsura
- John E Drake
- Luis M. Contreras
- Rakesh Gandhi
- Ran Chen
- Ricard Vilalta
- Rokui, Reza
- Ron Bonica
- Tomonobu Niwa
- Xufeng Liu
- + others who participate

Design Team Timeline & Status

IETF 106

- ✓ Plans
- ✓ Scope
- ✓ Early individual contributions (e.g., definitions)

IETF 107

- Initial framework draft from the design team

IETF 108

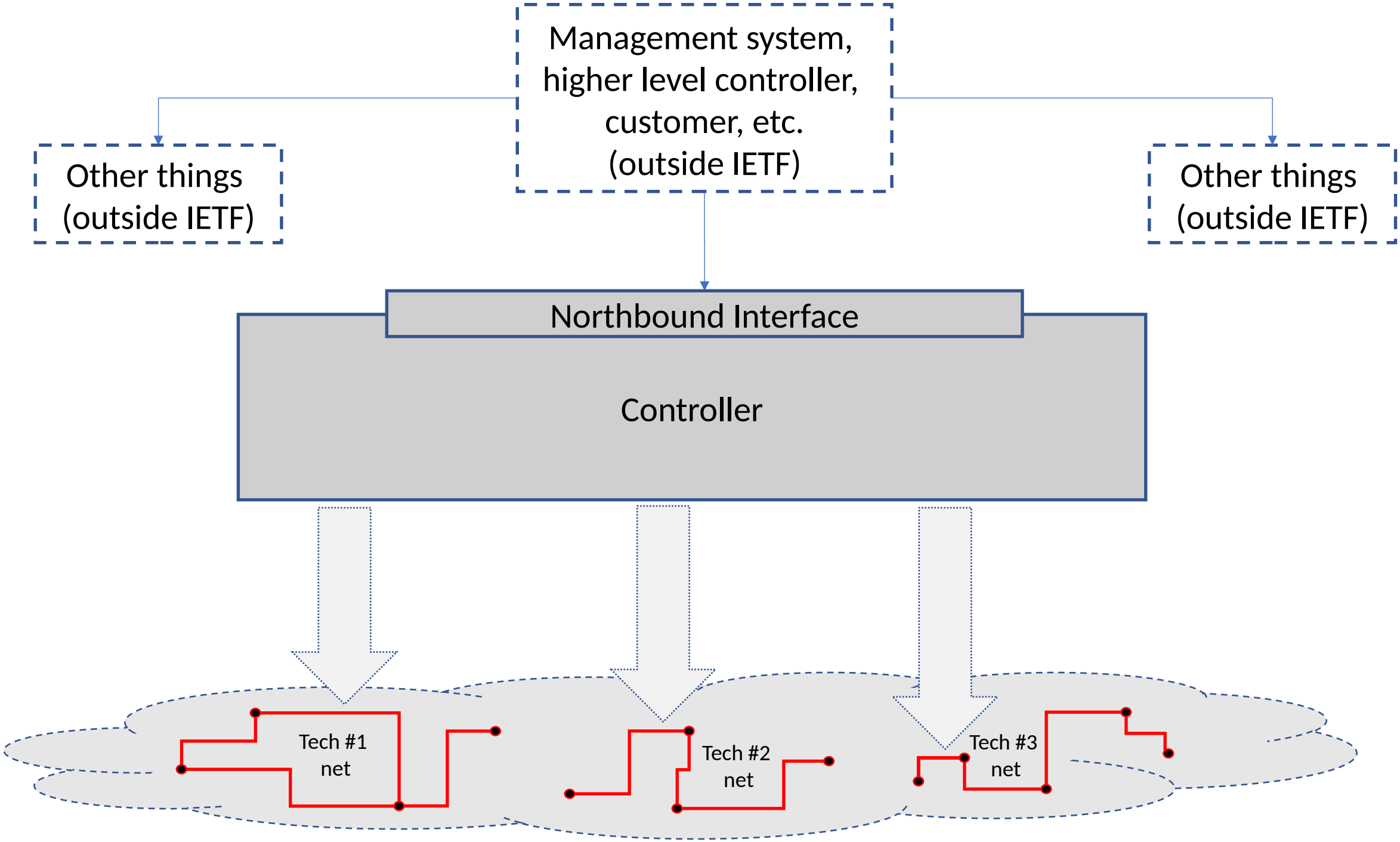
- Stable draft from the design team

Design Team Scope & Plan

”Back to basics” – explain how to use existing IETF transport technologies:

- Definitions of transport connections or slices
- Framework that describes the overall system, and its requirements
- Employ existing IETF TE tech for the necessary components and interfaces
  - Or extend as needed
  - A northbound interface for requesting connections with specific characteristics
  - How to map northbound requests to underlying IETF tech(s)
  - Underlying IETF tech, e.g., IP, MPLS, GMPLS, VPNs, etc
- Provide some use cases (as examples)

The design team plan is to publish documents on the above topics


Management system, higher level controller, customer, etc. (outside IETF)
Other things (outside IETF)
Other things (outside IETF)
Northbound Interface
Controller
Tech #1 net
Tech #2 net
Tech #3 net

Design Team Scope

But note also:

- Not an overall definition of all virtualization or all network functions (but can be used as a component in one)
- Not our role to define 5G slicing (even if they may use our results)
- Not overtaking any other SDO's roles, but rather working together
- Not about picking any single implementation technology
- Might be just better definition & explanation of existing tech
  - Extensions for underlying protocols and encaps not needed initially
    - It is possible that extensions prove useful later (but to be developed by relevant WGs)

Definitions – Example minimal definition

A Transport Slice is an abstract network topology connecting a number of endpoints, with an expected network service specified as an SLO

Notes:

- Similar to TE definitions, but technology agnostic and maybe wider
- This covers both p2p and other topologies (e.g., p2mp, hierarchical)
- The SLO specifies characteristics such as guaranteed bandwidth or latency, requirements to use of physical separation, etc.
- A slice connects to only to endpoints, which can be various network functions, interfaces, or end hosts. Endpoints may stitch slices.
- Covers only transport; firewalls, 5G nodes, etc. are not part of the model, but transport can connect to them – broader definitions possible

Definitions – Example minima[illegible]efinition

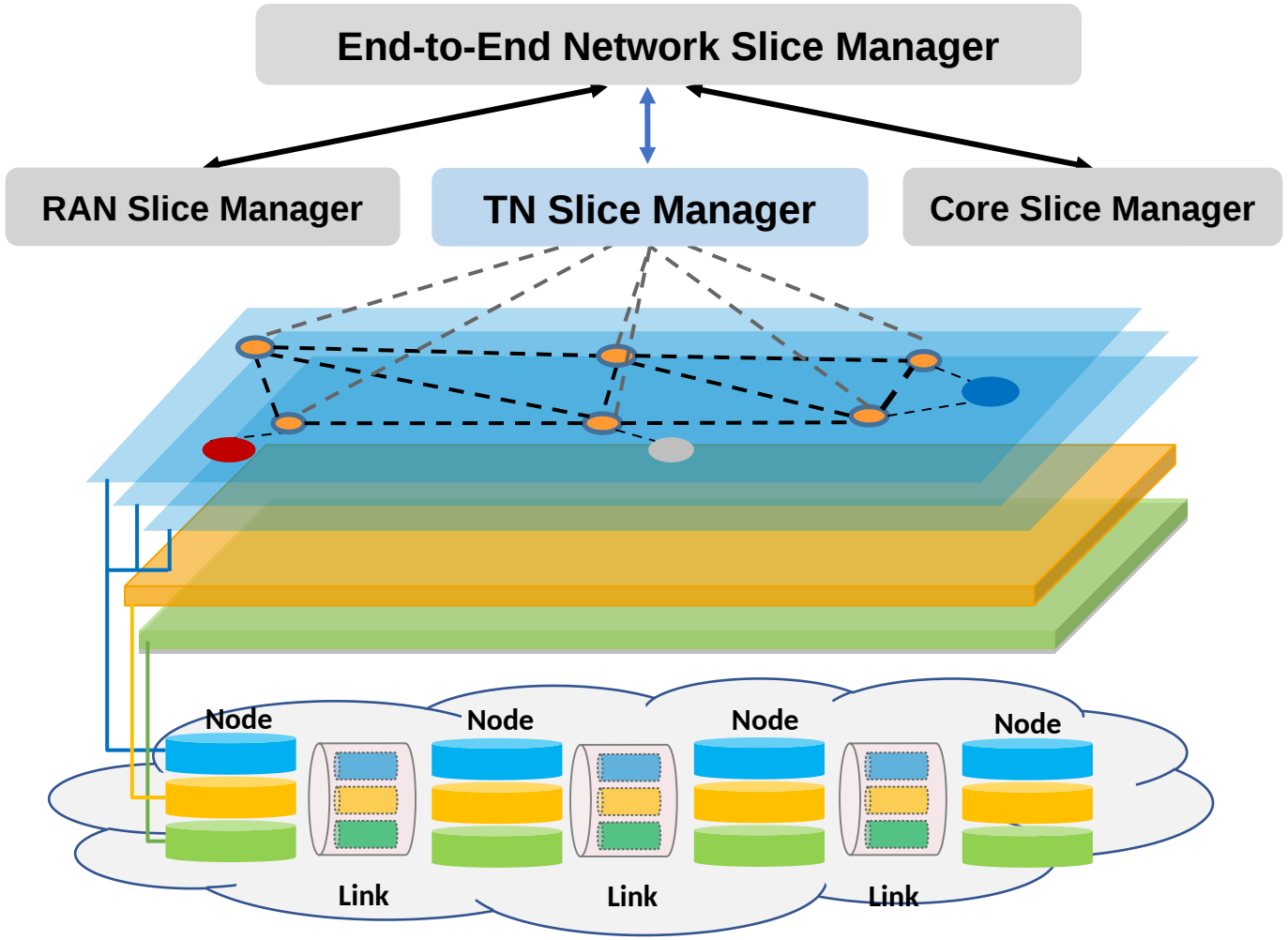
A <u>Transport Slice</u> is an abstract network to[illegible] a number of endpoints, with an expected networ[illegible]as an SLO

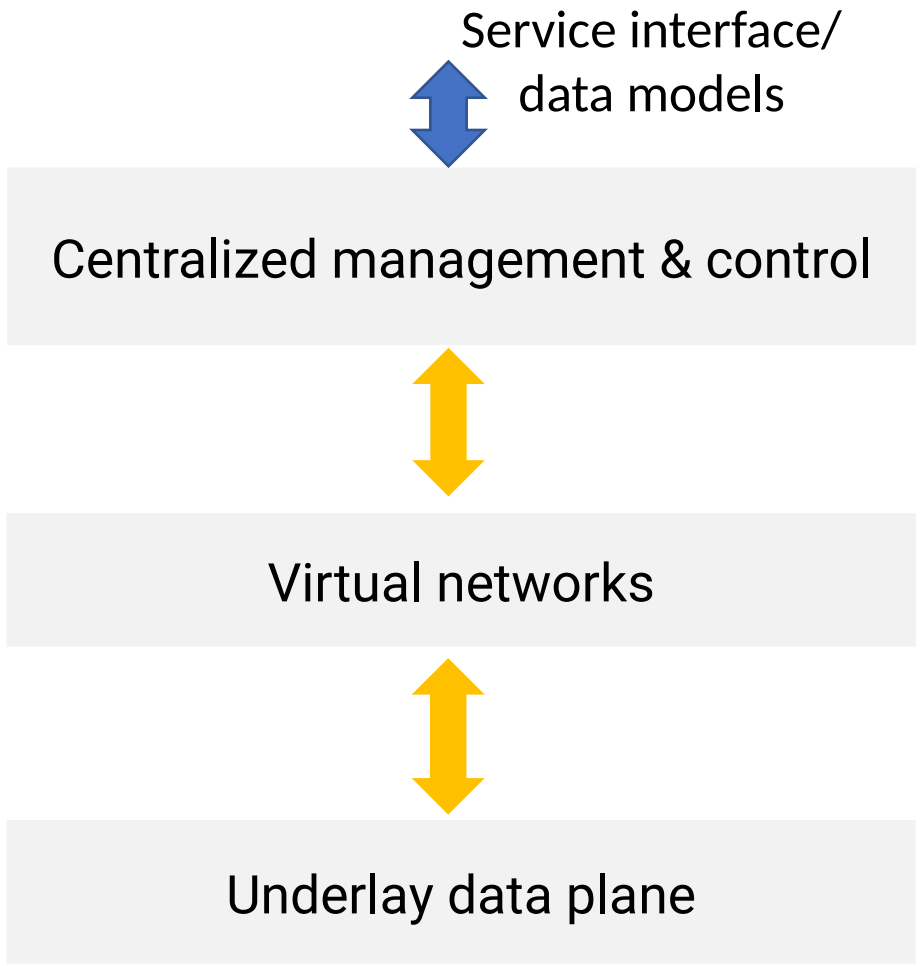
Notes:

- Similar to TE definitions, but tec[illegible]nd maybe wider
- This covers both p2p and ot[illegible]., p2mp, hierarchical)
- The SLO specifies charac[illegible]aranteed bandwidth or latency, requirements[illegible]separation, etc.
- A slice connects t[illegible], which can be various network functions, interfa[illegible]. Endpoints may stitch slices.
- Covers only transpor[illegible], 5G nodes, etc. are not part of the model, but transport ca[illegible]nnect to them – broader definitions possible

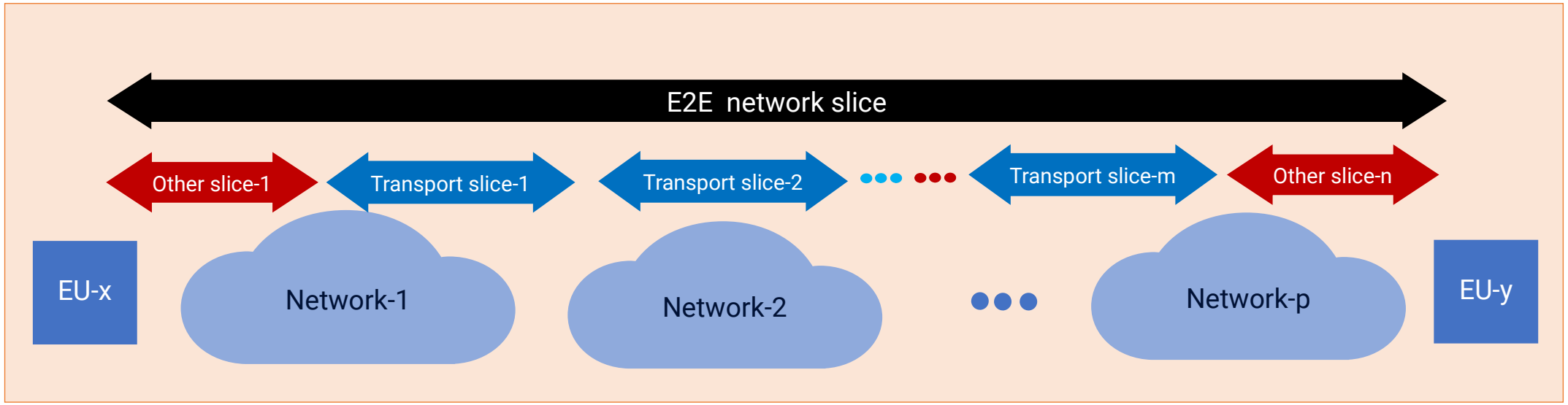
Note: Discussion still ongoing; no final agreement on name or the term SLO/SLA

Transport Slices in Networks – 5G Example





Transport Slices in Networks – 5G E2E Slices



Observation:

- The E2E network slice is different from Transport Slice
- An E2E network slice might contains multiple transport slice
- An E2E network slice might contains “Other type of slices”

Pointers for further reading

Announcement: https://mailarchive.ietf.org/arch/msg/teas/jiHWXU_i5kK5BzjRffFbYnbZJfs

List archive: https://www.ietf.org/mailman/listinfo/teas-ns-dt

Notes from calls: https://github.com/teas-wg/teas-ns-dt/tree/master/notes

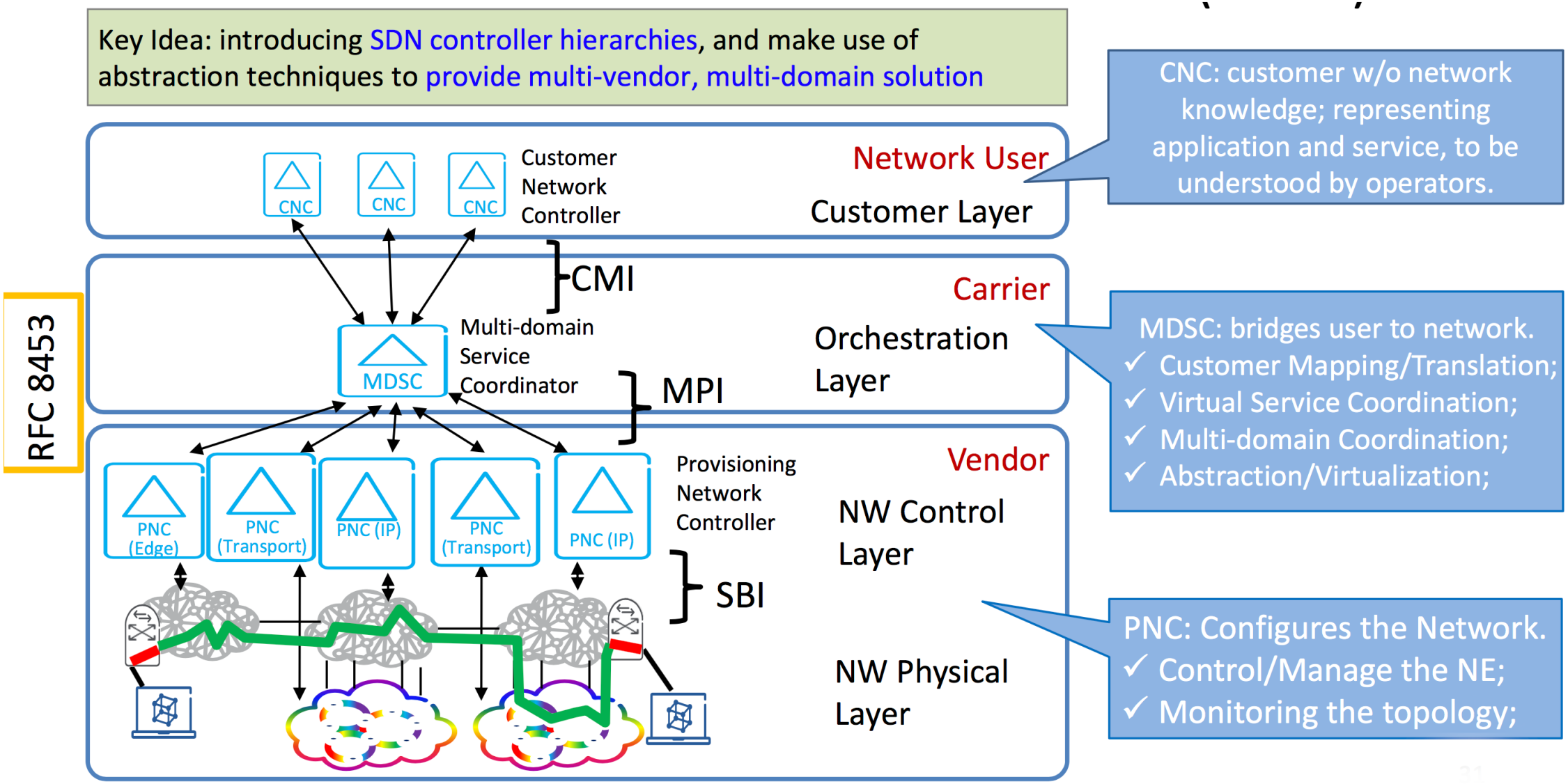
Some currently discussed documents:

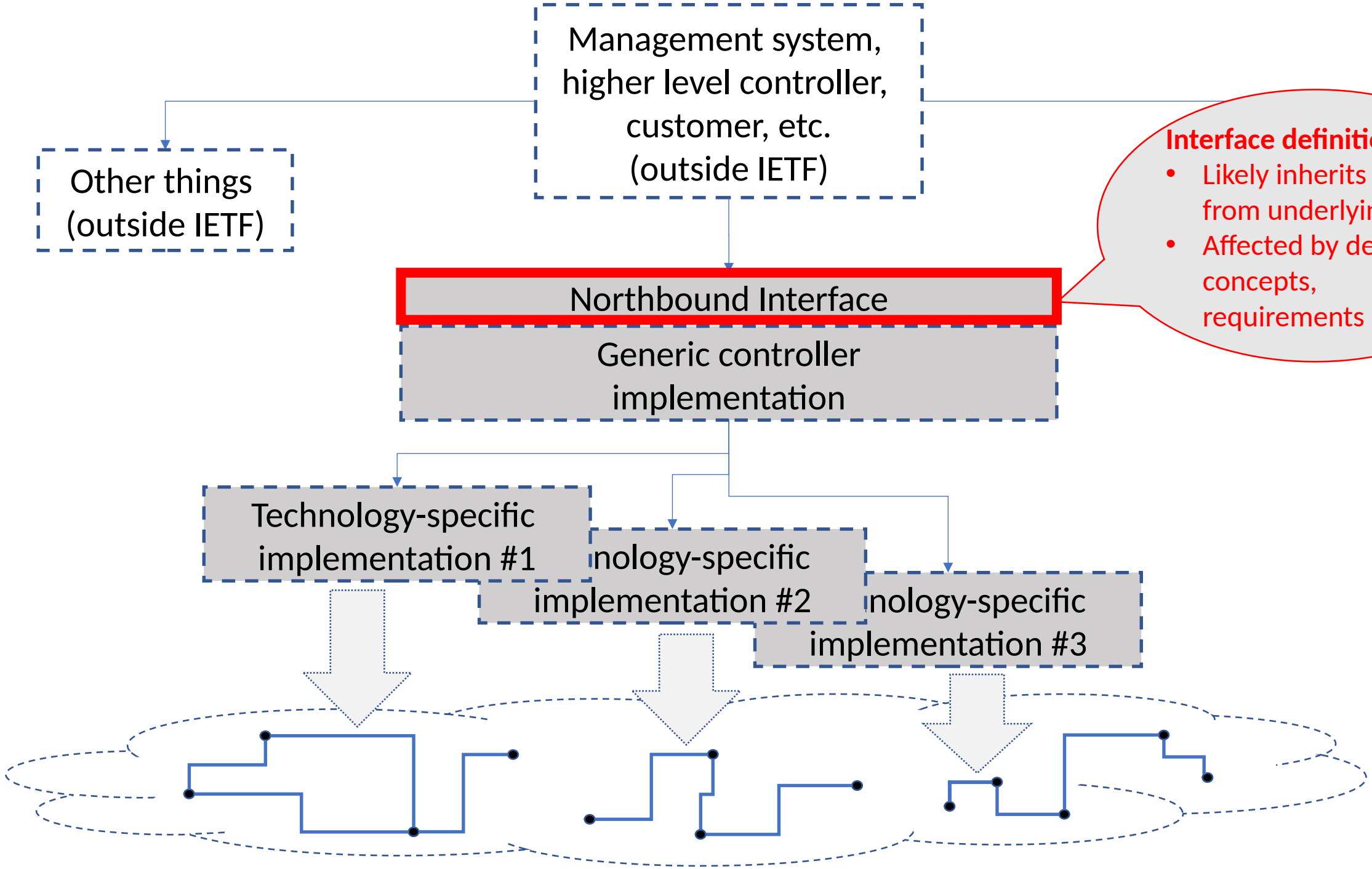
- Definitions https://tools.ietf.org/html/draft-nsdt-teas-transport-slice-definition-00
- Data model and APIs https://tools.ietf.org/html/draft-liu-teas-transport-network-slice-yang-00 and https://datatracker.ietf.org/doc/draft-contreras-teas-slice-nbi/
- Generic comments https://tools.ietf.org/html/rfc8345 and https://tools.ietf.org/html/draft-ietf-teas-actn-yang

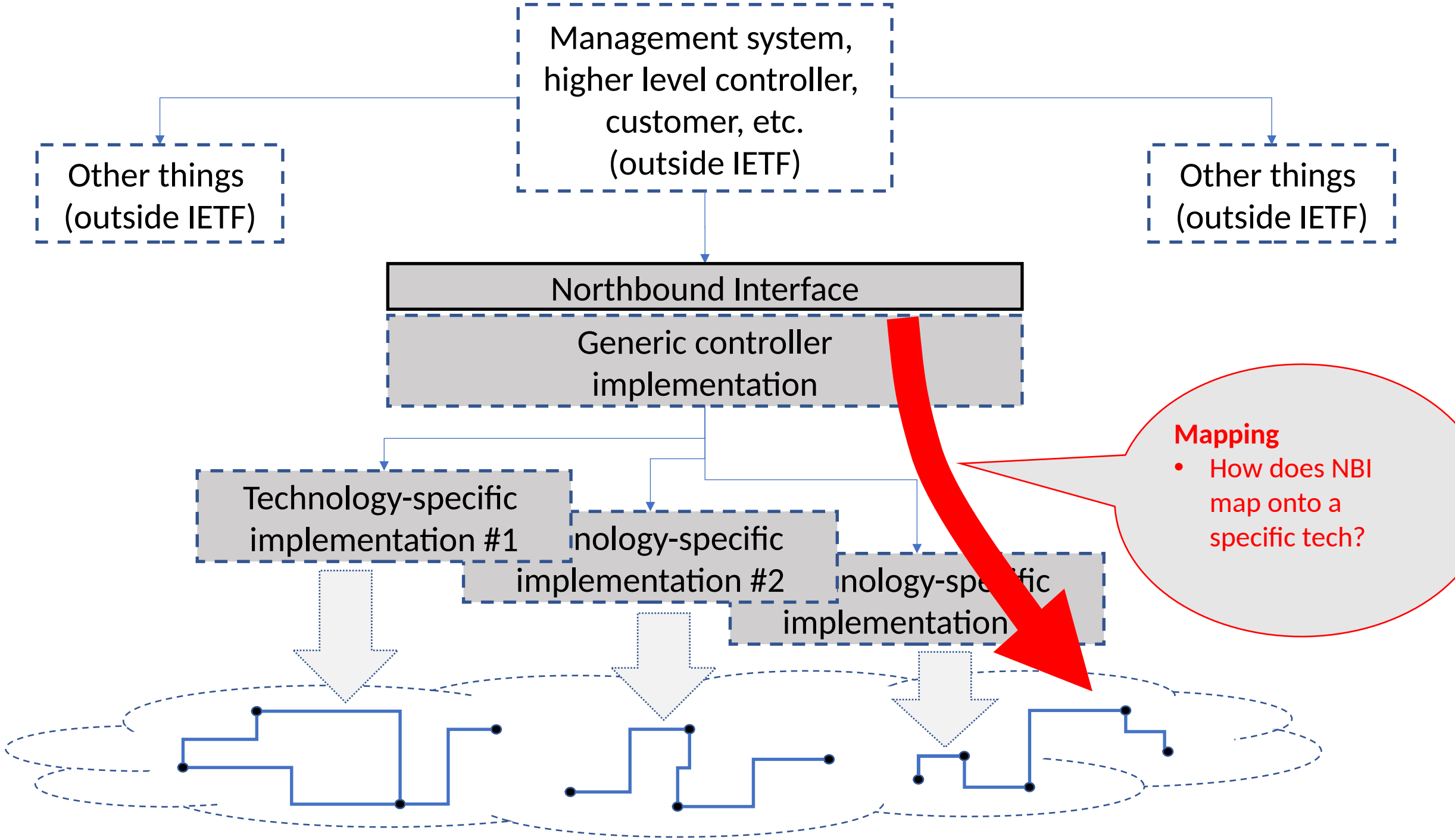
Questions? Comments?

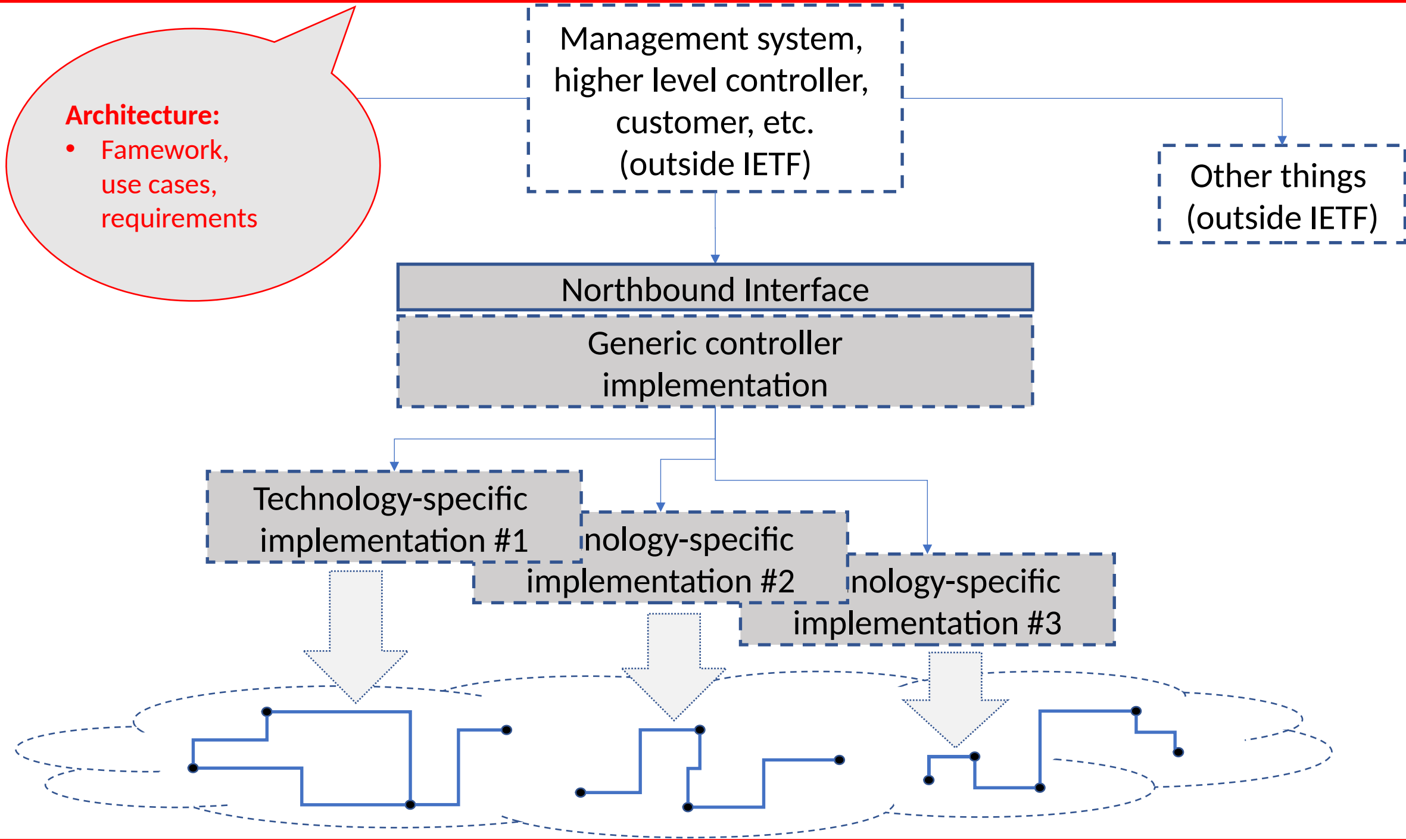
Backup Slides

Transport Slices in Networks – ACTN Example




Management system, higher level controller, customer, etc. (outside IETF)
Other things (outside IETF)
Interface definition
• Likely inherits much from underlying tech
• Affected by definition concepts, requirements
Northbound Interface
Generic controller implementation
Technology-specific implementation #1
nology-specific implementation #2
nology-specific implementation #3


Management system, higher level controller, customer, etc. (outside IETF)
Other things (outside IETF)
Other things (outside IETF)
Northbound Interface
Generic controller implementation
Technology-specific implementation #1
nology-specific implementation #2
nology-specific implementation
Mapping
How does NBI map onto a specific tech?


Architecture:
• Famework, use cases, requirements
Management system, higher level controller, customer, etc. (outside IETF)
Other things (outside IETF)
Northbound Interface
Generic controller implementation
Technology-specific implementation #1
nology-specific implementation #2
nology-specific implementation #3