Sustainable Network Automation

From Minimum Viable Product to Mature , Valued Platform

Allan Eising
Network automation architect
Telia Company

MVP

“Minimum Viable Product”

Not

“Most Valued Player”

Minimum Viable Product

“A minimum viable product (MVP) is a version of a product with just enough features to be usable by early customers who can then provide feedback for future product development.”

Pitch

When setting out to do network automation, you often start small with a Minimum Viable Product.

But what is that really, and what do from there?

What have I learned?

The all or nothing MVP

A tumultuous start

Or when agile is not agile

It all started in 2018…

The catalyst

— In 2018 Telia bought TDC Norway and their B2C subsidiary GET

— In 2019 a project was started to build a new unified datacom stack

— The project had a strict "big bang" deadline

— Failure was not an option, nor was partial delivery

Telia kjøper Get-eier

Telia har lenge jaktet etter en åpning i det norske bredbåndsmarkedet. Etter at Broadnet glapp har den svenske telegiganten nå fått kloa i Get-eier TDC Norge. Prislappen er 21 milliarder kroner.

KJØPER GET: Telia-sjef Johan Dennelind (t.v.) har tidligere hintet om oppkjøp av en norsk bredbåndsaktør. Nå slår selskapet kloa i Get-eier TDC Norge. Her er han i en samtale med Norgessjef Abraham Foss. Foto: Marius Lorentzen E24

Av André Haugen og Marius Lorentzen

Oppdatert 20. juli 2018

OSS system for Service Order Management

Here processes should be made into automated workflows.

This should give us automated customer deliveries

A modern inventory

Here we could store everything from logical resources Hardware, topology, IP addresses, service data, etc.

A network activation platform

Built on NSO, we would get intent-based service activation with transactions, rollbacks and so on

BSS systems with support for unified products

Misc

Replace a large number of smaller supporting systems with modern equivalents

Minimum Viable Product?

Or everything and the kitchen sink?

The beginning was exciting

- Lots of exciting discussions, white board sessions
- Designing new systems from the ground is fun

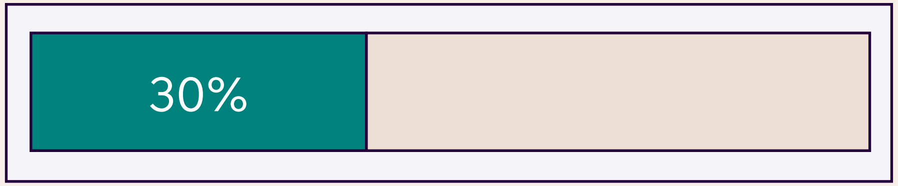
Scope completion

5%

The Network Automation

- Our project team was a cross-functional team of engineers and developers
- We had forked the existing NSO codebase, refactored and extended
- We had also inherited the automation architecture ideas
- We tried our best to make sense of it all
- We had a basic non-declarative demo ready after a few months

Scope completion



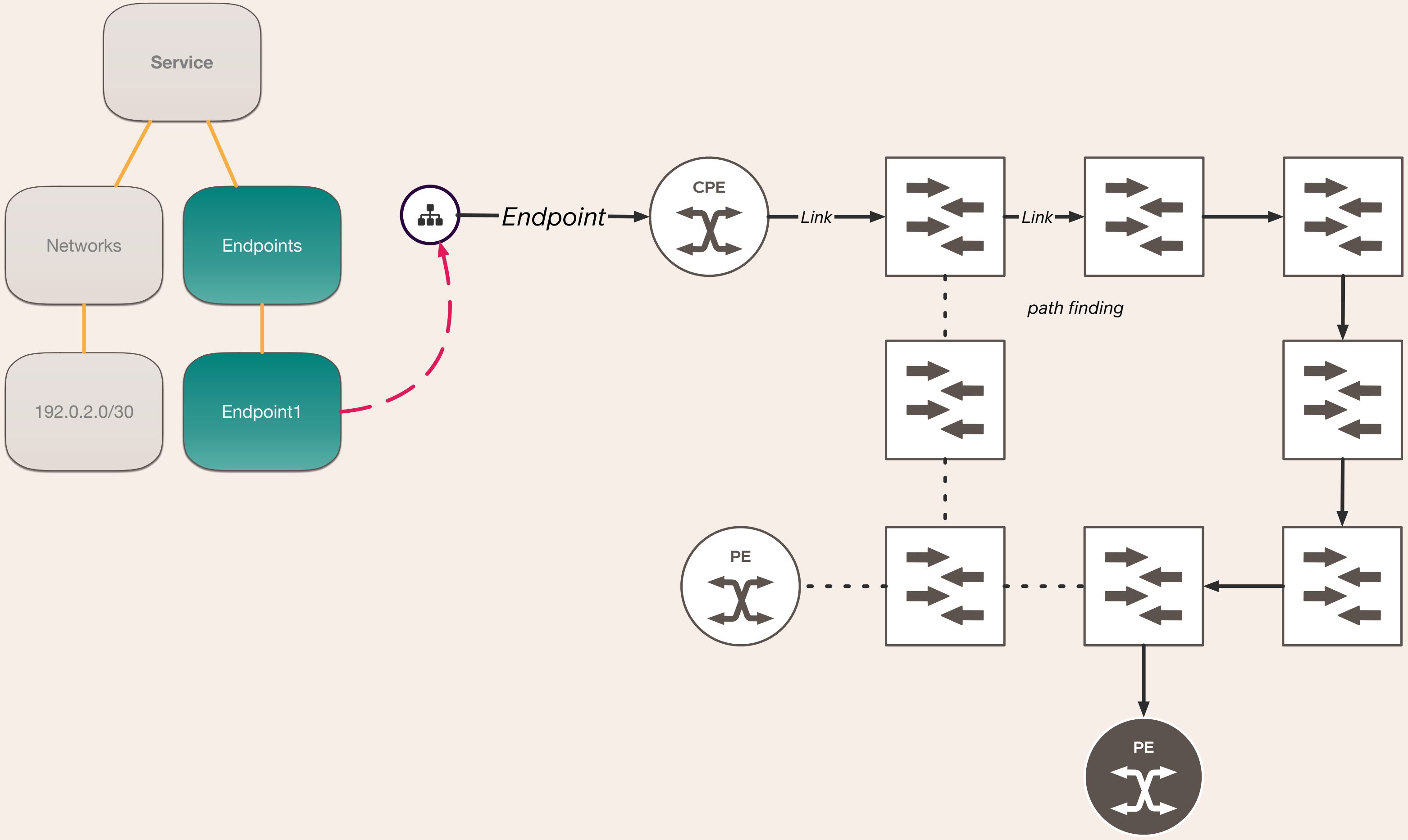
Endpoint

- The service would only contain customer intent and where to apply it (endpoint)

- An endpoint would usually be on a CPE

- The endpoints point to devices with links that can be followed

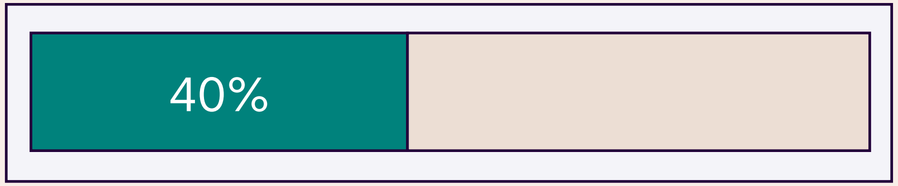
- This required accurate integration between systems



Setbacks

- Nobody predicted the first setback: Corona
- Achieving coherent system architecture requires extensive collaboration
- We lacked workshops, whiteboards and coffee machine talk
- Further setbacks came as we approached the deadlines

Scope completion



Halfway through 2021, the deadline arrived

Scope completion

?%

Reducing into a

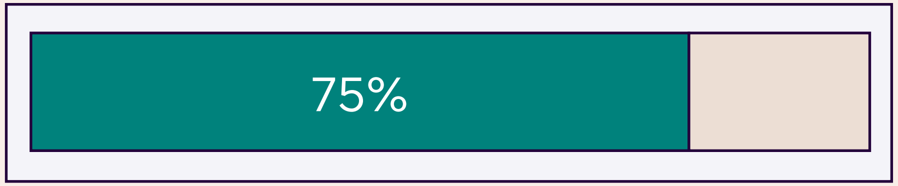
Minimum Viable Product

Finding out what the smallest acceptable delivery was

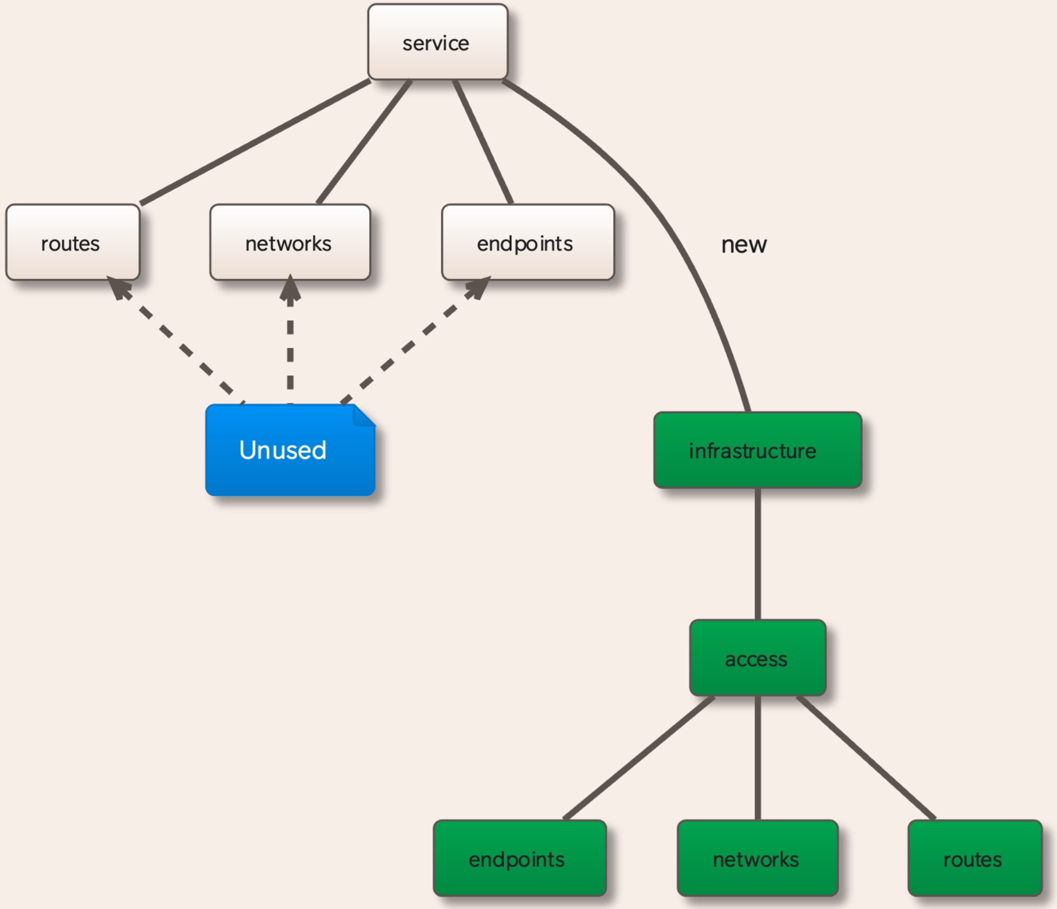
Minimum delivery out of necessity

- We had to create a new solution based on what was ready enough
- The solution was named “hybrid”
- It mixed new components with homegrown/legacy components
- We managed to make something that would just barely be able to do the job
- A actual MVP

Scope completion



We re-designed our automation



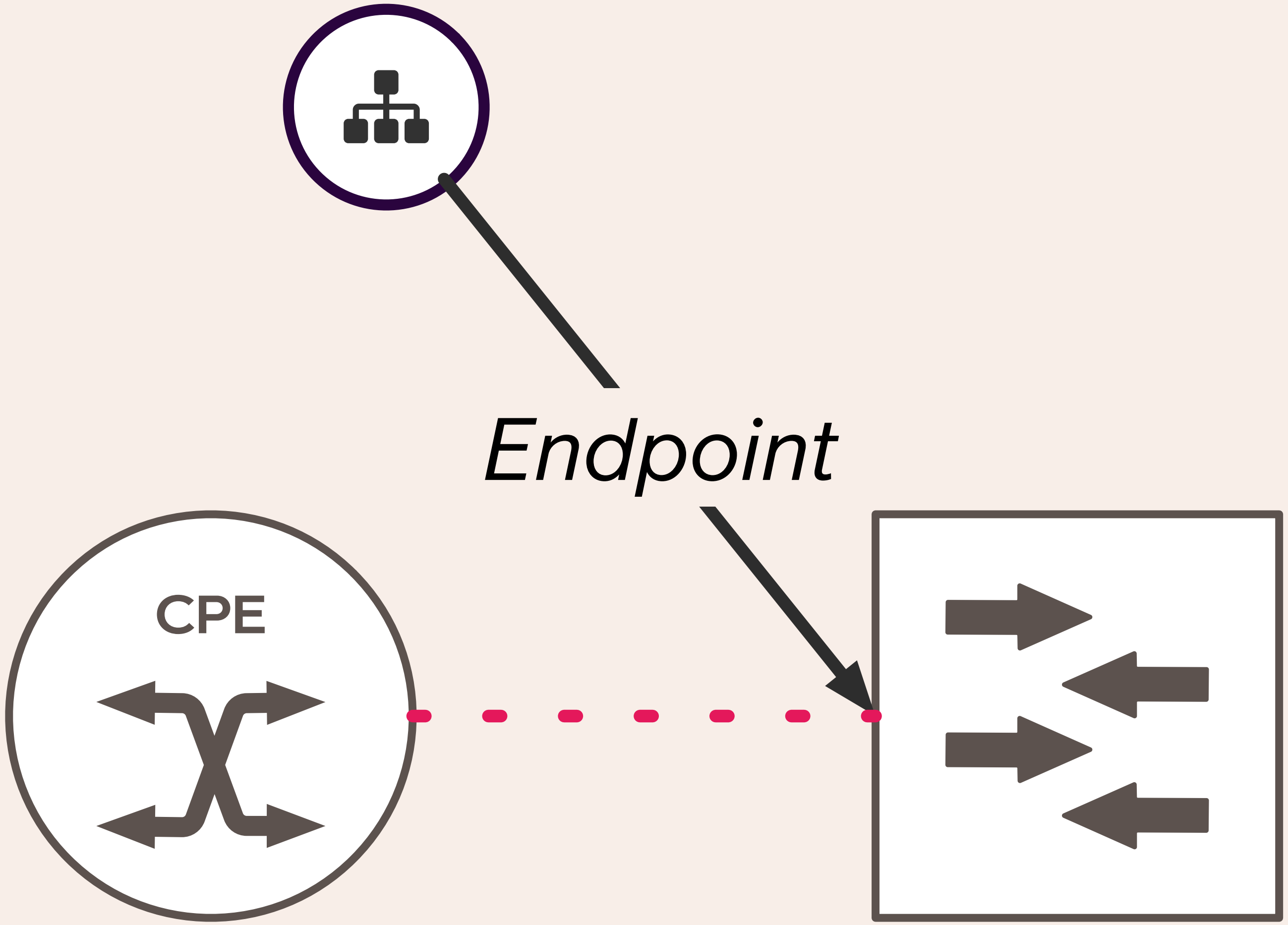
- We made a sub-section in our services to fit the network side
- We left the CPE-oriented side blank for later

Scope completion

80%

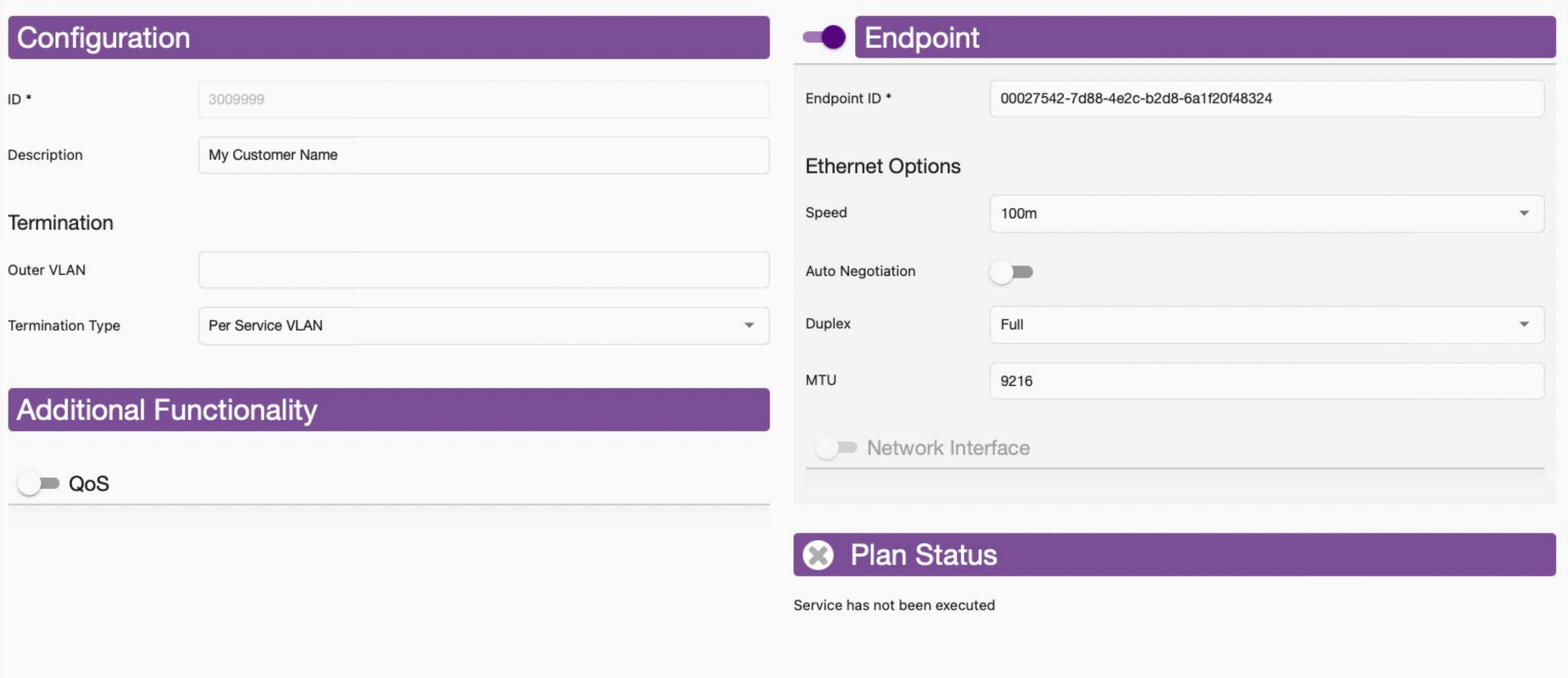
Network endpoints

- The endpoint was on the network now
- We had much higher data quality



New UI Frontend

— New frontend was developed to make it easier to get going



Configuration

ID * 3009999

Description My Customer Name

Termination

Outer VLAN

Termination Type Per Service VLAN

Additional Functionality

QoS

Endpoint

Endpoint ID * 00027542-7d88-4e2c-b2d8-6a1f20f48324

Ethernet Options

Speed 100m

Auto Negotiation

Duplex Full

MTU 9216

Network Interface

Plan Status

Service has not been executed

Scope completion

90%

Preparing for go-live

- We ran daily discrepancy reporting
- We compared the output of our automation with existing services
- This was an iterative process where we fixed code and identified non-harmfull changes
- We knew we could go live once we hit a high enough percentage

Scope completion

95%

Sustainability and reusable components

— We managed to re-design thanks to a modular codebase

— We benefitted from a layered design

— Everything was consistently modelled

— It was relatively simple to add UI frontends to all our services, since they all followed a similar pattern

Scope completion

100%

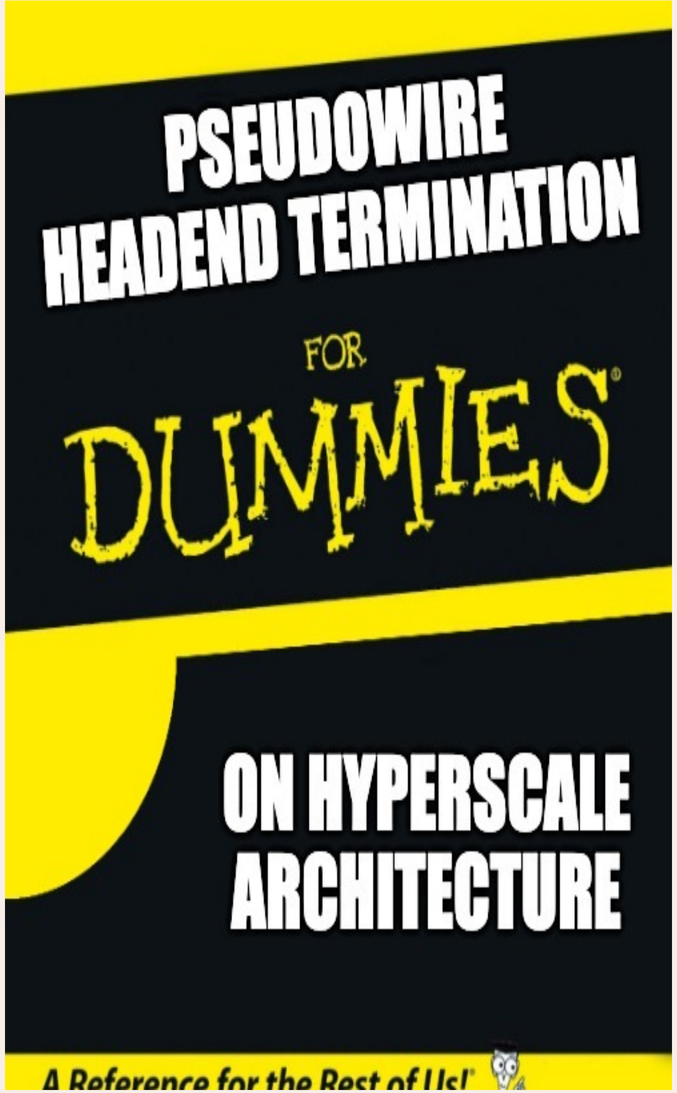
We went into production and everything was great

The end!

Just kidding...

Next MVP: Shoe-horning in new platform support

Automating new telia services and network designs


PSEUDOWIRE HEADEND TERMINATION
FOR
DUMMIES
ON HYPERSCALE ARCHITECTURE
A Reference for the Rest of Us!

New transport network and common services

- First major addition to the platform was a new service on a new network
- This was the ”common” l3vpn service called Datanet
- This was going to built on top of a completely different network architecture
- The service was designed for another project originally

Reusing as much as you can

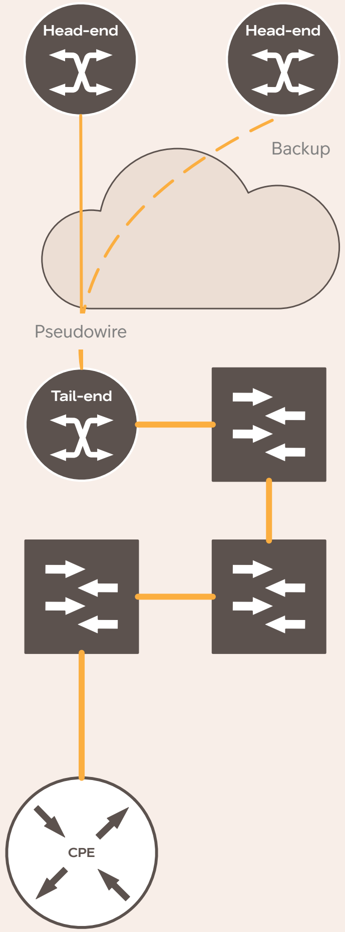
— The model was different, and much of the logic was too

— It had to activate new Cisco ASR9000 routers as well as existing Cisco switches and Juniper PE routers

— We could re-use our existing foundation

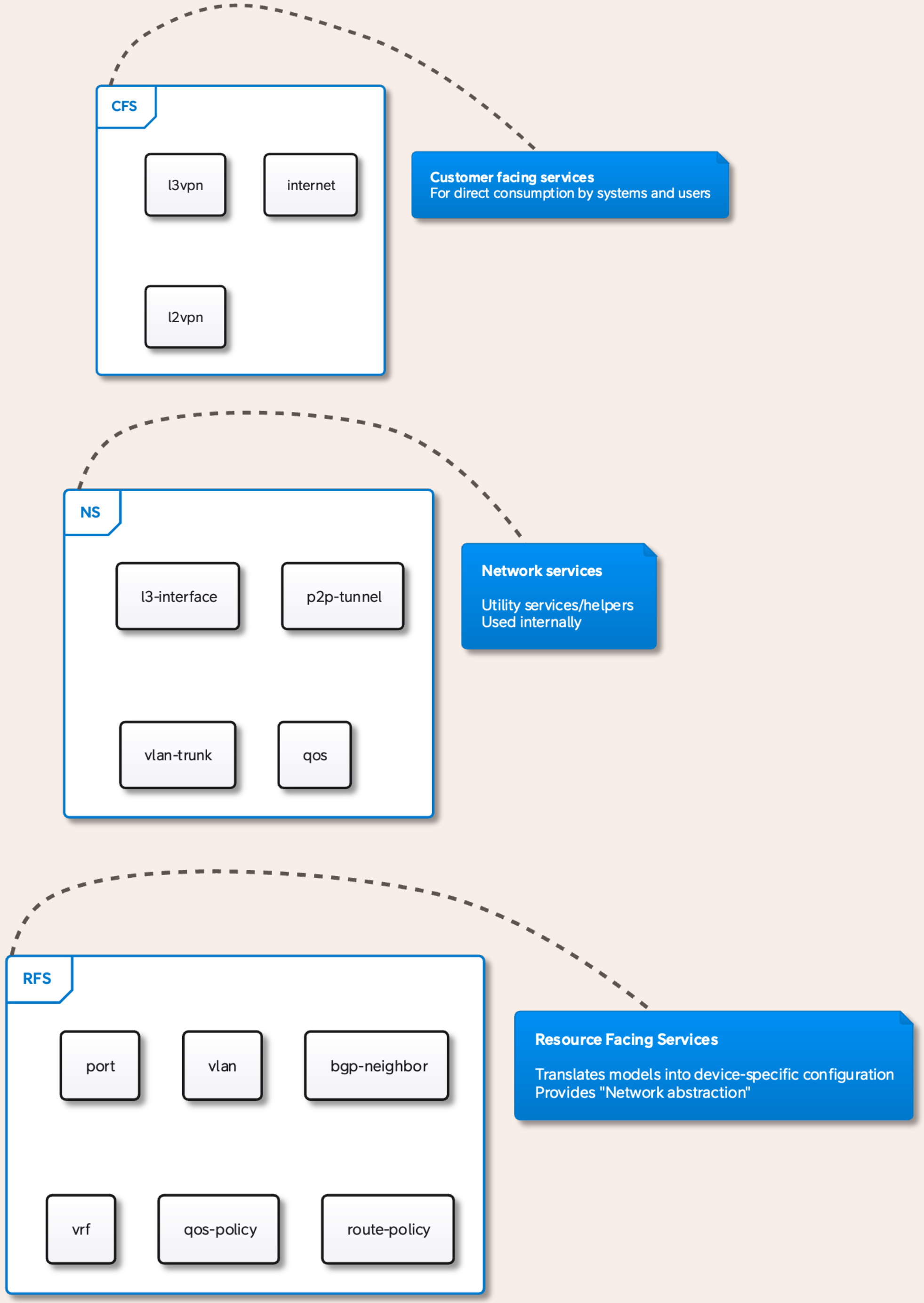
— We could save time by sending the IOS-XR configuration directly from the service

— Minimum Viable Product: IOS-XR templates, reuse components for the rest



Layered services

- Our foundation was a layered architecture
- This model allowed us a lot of flexibility but gave us some overhead as well
- Writing directly from service to a device was also possible

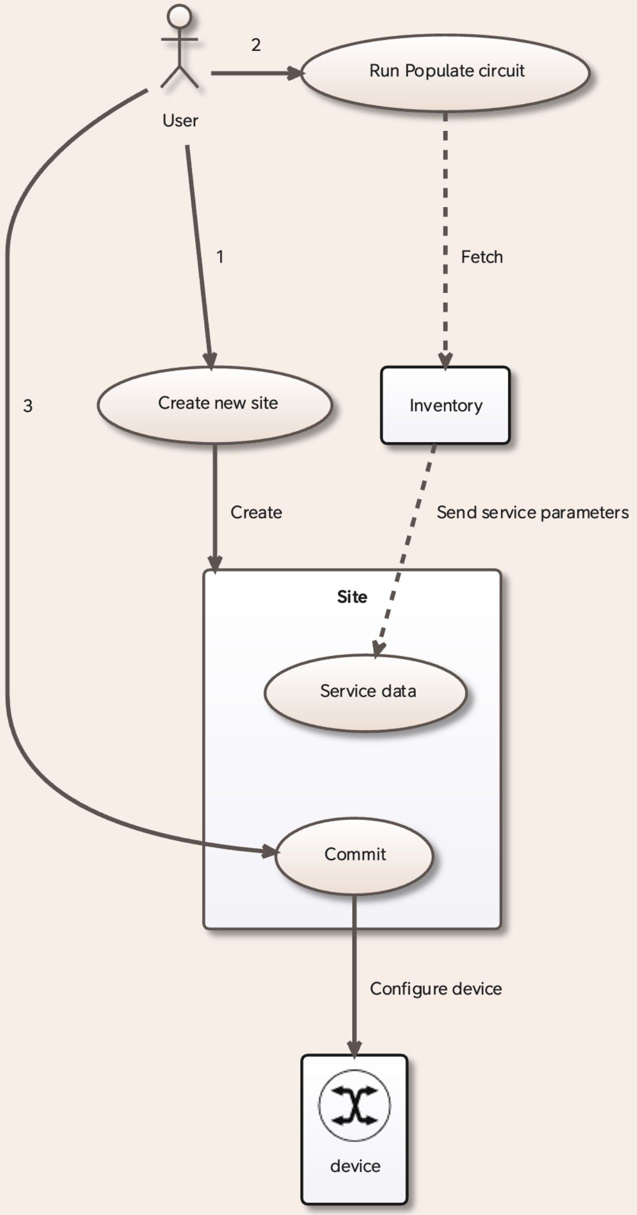


Result

- The MVP got into production quickly thanks to our flexible foundation
- We did not have time to improve our layers to support IOS-XR
- But the project managers were happy and we reached our deadline
- The new services kept coming…

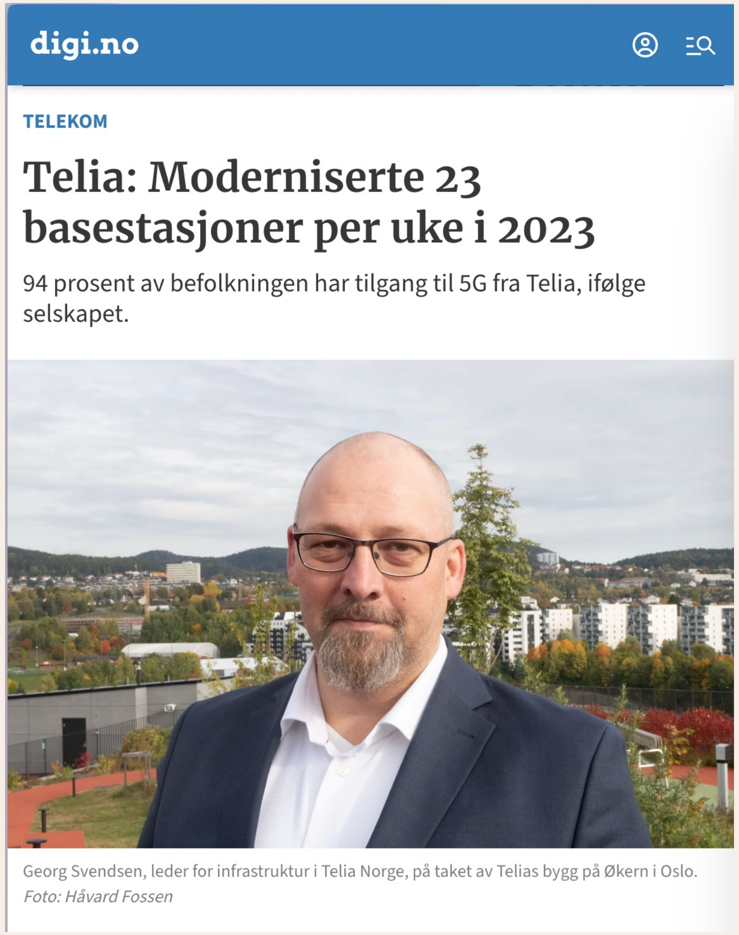
Next service: Mobile backhaul

- This service existed in other countries already
- We improved it with support for fetching data from inventory
- With a good user experience in our webapp, you could create new sites with very few clicks of a mouse



Results hit the news

- Simple provisioning flow means more people can do it
- No need to use network experts
- The milestone was achieved ahead of schedule because of this
- But still no IOS-XR support in our layers
- More tasks were added to the backlog



digi.no

TELEKOM

Telia: Moderniserte 23 basestasjoner per uke i 2023

94 prosent av befolkningen har tilgang til 5G fra Telia, ifølge selskapet.

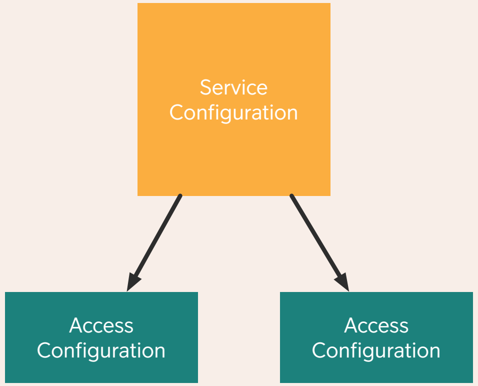
Georg Svendsen, leder for infrastruktur i Telia Norge, på taket av Telias bygg på Økern i Oslo.
Foto: Håvard Fossen

A better MVP: Getting more right from the start?

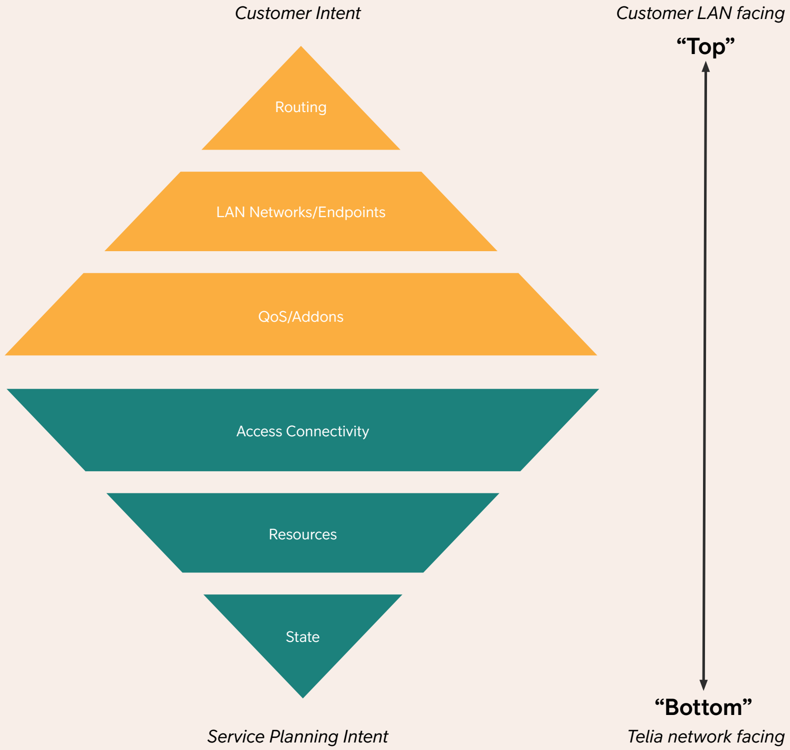
- I got the chance to write a new service from scratch
- Trying to learn from past mistakes
- If we didn’t get time to update our services with our learnings, could we make it right from the beginning?
- Here are the cliff-notes from my design

Smarter multi-access services

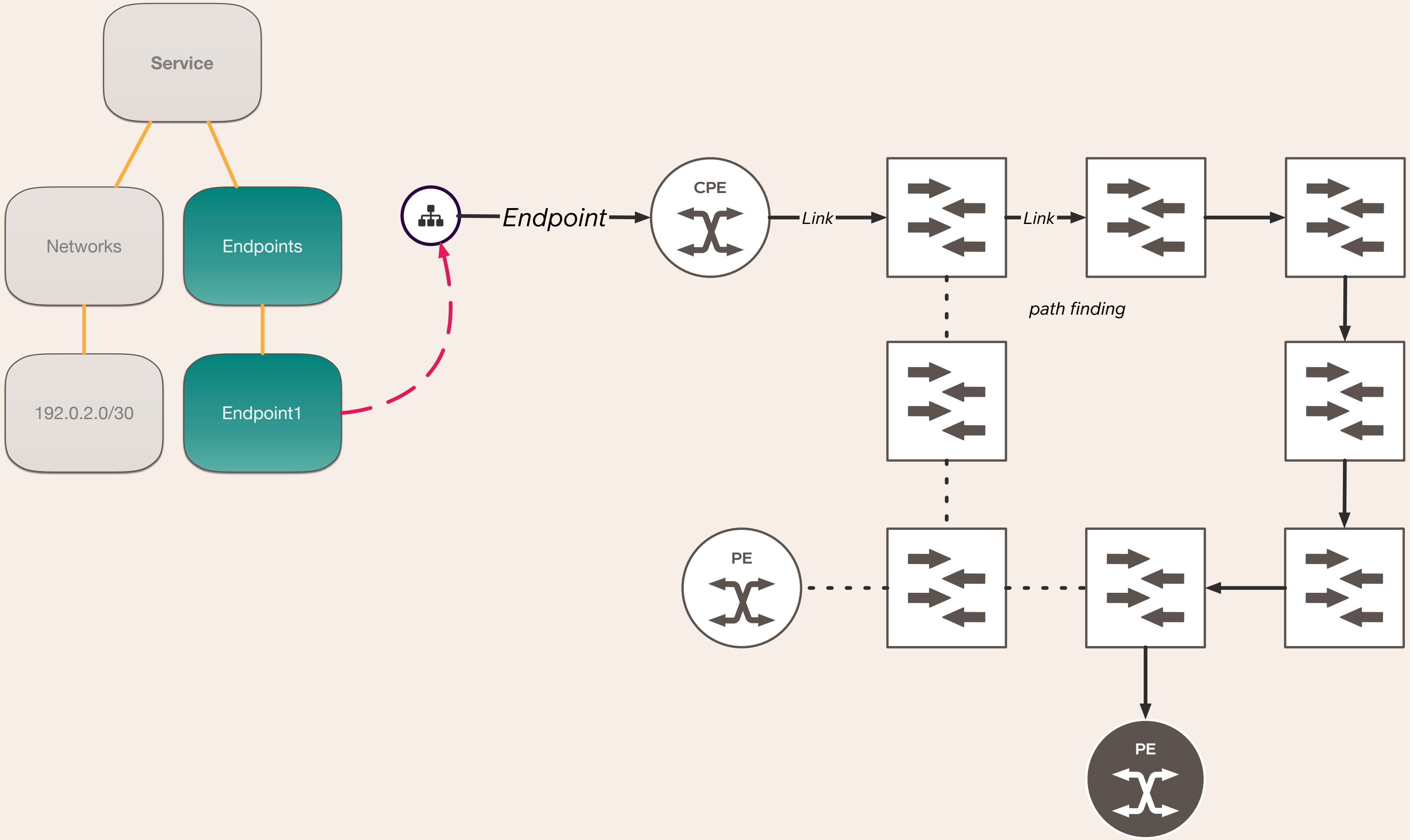
- I tried to put our learnings into the next service
- Our current model handled dual-homed CPEs poorly
- We knew it was going to be needed, so why not build it right from the start?

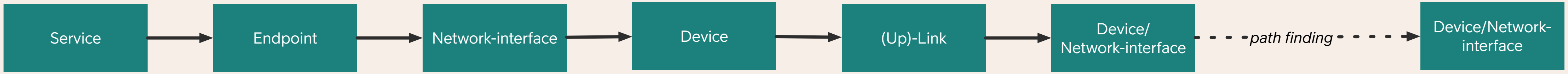


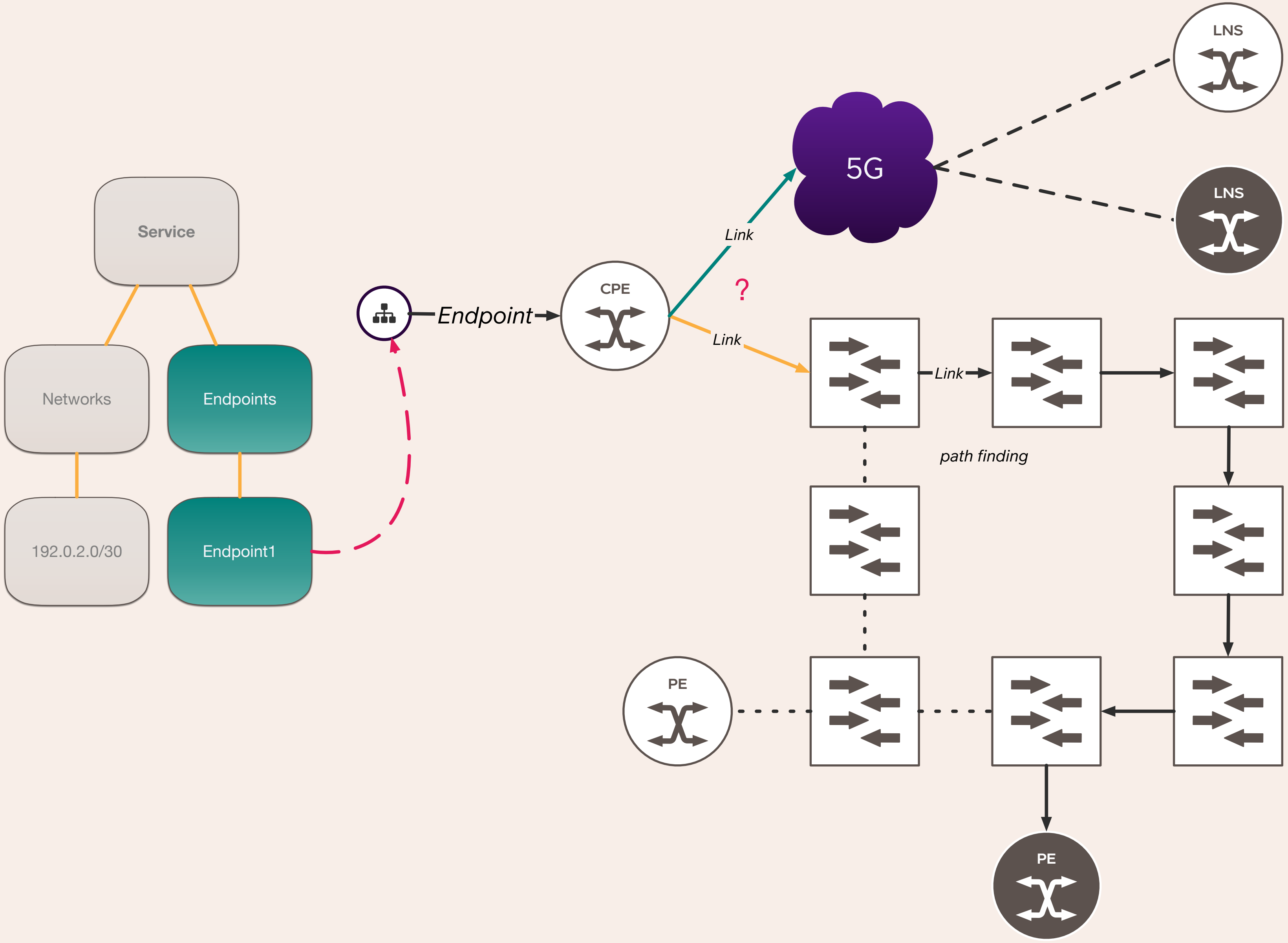
Services can have multiple accesses



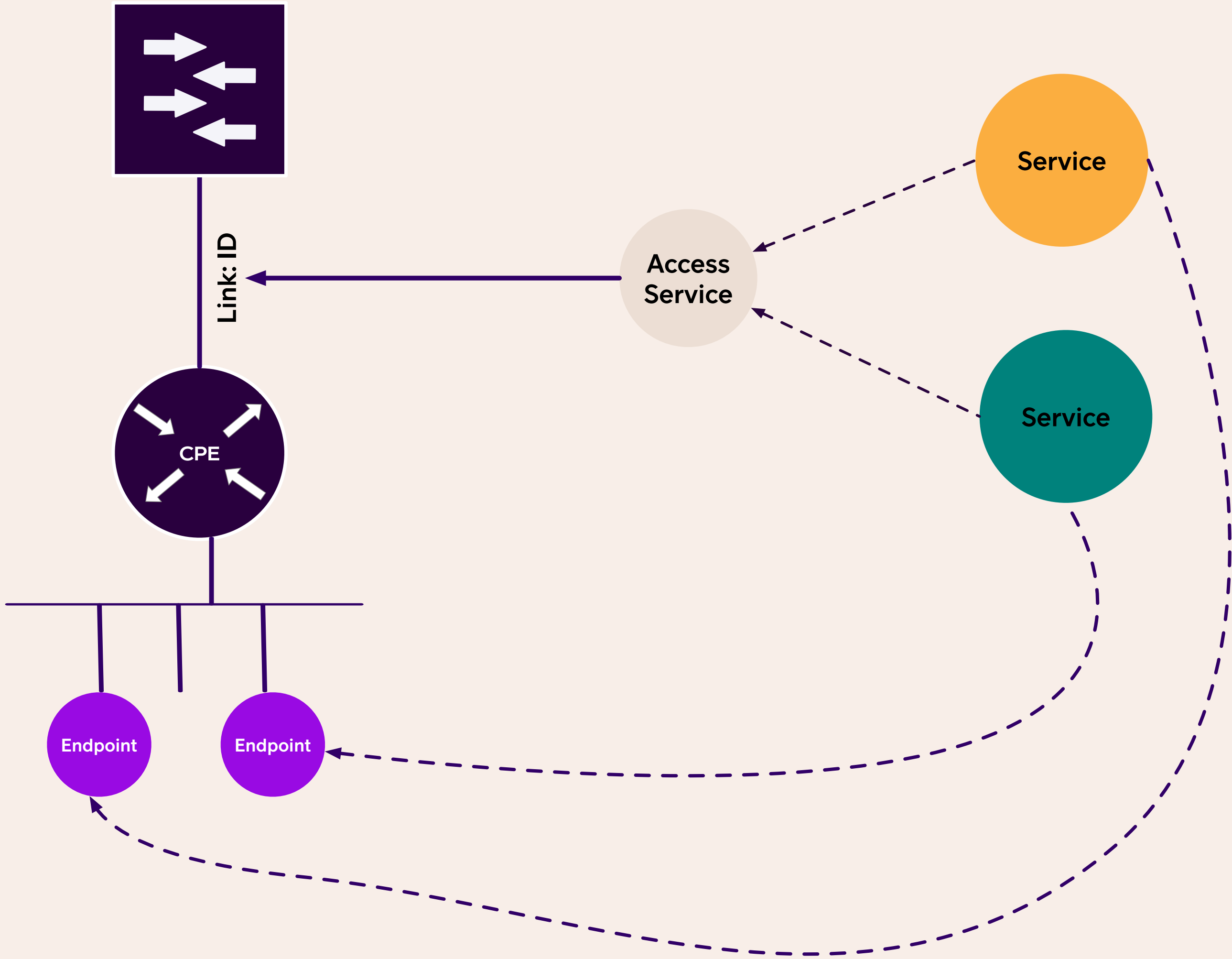
The old model

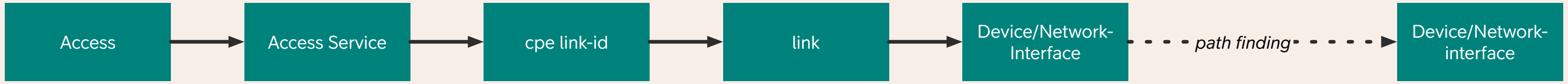




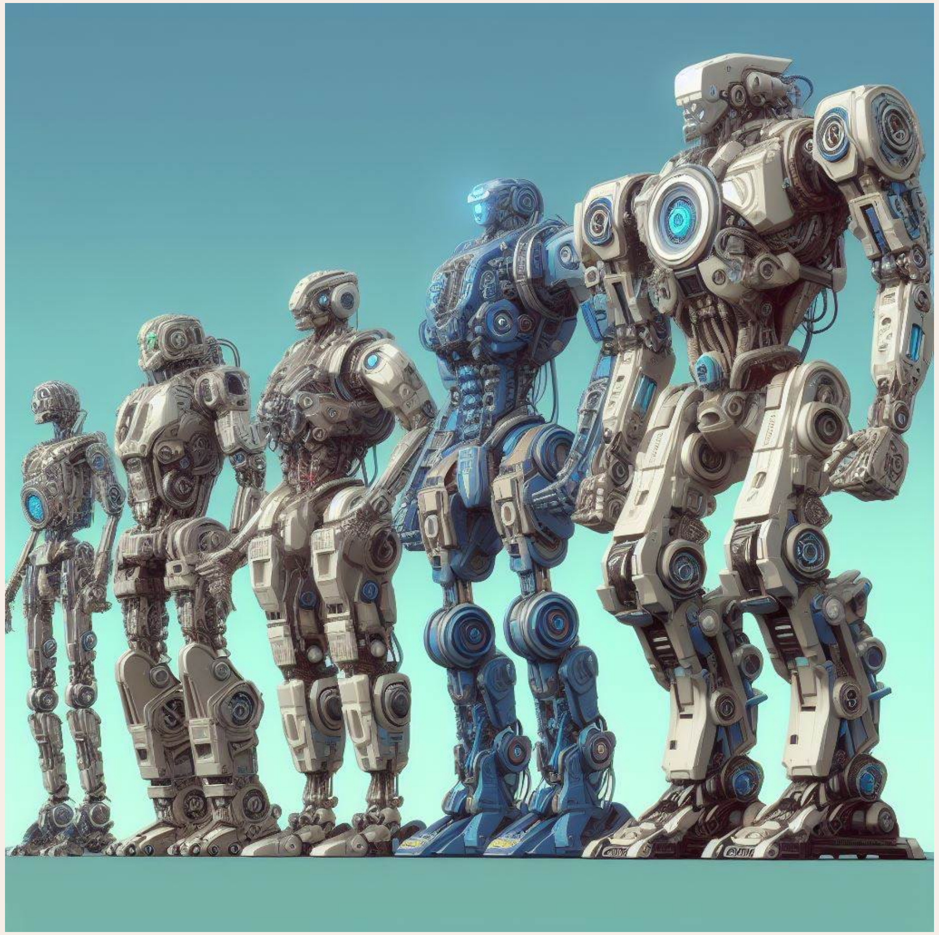

Service
Networks
Endpoints
192.0.2.0/30
Endpoint1
Endpoint
CPE
Link
?
Link
5G
LNS
LNS
Link
path finding
PE
PE

Access Connectivity Mode





The evolution



- Our services become better and better
- But the older services rarely benefit

Technical debt

- Every time we launched an MVP there was more to do
- Some things were small, but they all contributed to the overall technical debt
- Even outside of the known issues, growth and success exposed other weaknesses

- We have had to fight a lot to get time to improve our services
- There are still much more to go and little time.

What can you do?

- Visibility is key. Document everything!
- Make road-maps for your services that span longer than just the MVP
- Try to sell internal improvements together with small new features
  - Example: “If you let me spend time improving the service overall, I can give you IPv6 support”
- It’s a lot harder to justify spending time on tech debt than on new developments
- Reserve a little bit of time regularly for improvements outside of just bugfixing

Thank you!

Questions or comments?