

Grensesnitt mot Telenor - Bredbåndsforum

11. mars 2019

Dette dokumentet beskriver et forslag til løsning på hvordan ISP-ene skal ha grensesnitt mot Telenor i forbindelse med nytt VULA-produkt. Det beskriver grensesnitt for

- Fault Management and Trouble Ticketing
- Service Delivery
- Technical Self Service (ny)
- Test and Diagnostics
- Performance (ny)

Fault Management and Trouble Ticketing

Det finnes allerede system- og prosess-støtte for dette der ISP-ene har tilgang til Telenors feilhåndteringssystem (FHS). Telenor foreslår at det ikke gjøres noen endringer på dette.

Service Delivery

Service Delivery tilbys i dag igjennom VULA Web, Jara NB, og Jara B2B. Telenor foreslår at de evt nye produktene vil tilbys der på tilsvarende måte som dagens VULA. Alle betalbare elementer av en tjeneste må endres via dette grensesnittet. Resten av dette dokumentet er ikke relevant for Service Delivery.

Technical Self Service

Med teknisk selvbetjening så menes det at ISP-en ved hjelp av API-er skal kunne endre på features tilhørende en allerede levert tjeneste.

For å tilrettelegge for at ISP i størst mulig grad kan lage egne produkter til sluttkunde og at ISP i best mulig grad kan forvalte egne aksesser foreslår Telenor å utvikle et nytt API på NMS-nivå. Ved å tilby API-er på NMS-nivå så vil vi kortslutte resten av Telenors verdikjede brukt ved Service Delivery. Det vil gjøre det mulig å forvalte API-et og funksjoner på en mer agil måte enn det som er tilbudt via Jara i dag. På den annen side medfører det noen begrensninger:

API-et vil kun tilby konfigurerbare elementer av en tjeneste som

1. **ikke** har et betalingselement til seg
2. **kun** påvirker Telenors NMS-systemportefølje. Det inkluderer f.eks. ikke inventory-systemene.

Konkret hvilke features som vil være tilgjengelige må avklares mellom Engineeringsavdelingene i Telenor og de respektive ISP-ene. Alle ISP-er vil få tilgang til samme funksjonalitet.

Hvilke features som er tilgjengelige vil kunne endre seg over tid avhengig av hva som er tilgjengelig på nodene og evt sikkerhet- og skaleringsbegrensninger. Ved f.eks. software-oppggraderinger kan det være at ikke alle elementer vil være tilgjengelige i hele nettet samtidig.

Telenor vil informere ISP-ene om endringer på API-et. Telenor vil tilstrebe 3 måneders overlapp der ny og gammel funksjonalitet vil være tilgjengelig samtidig.

Alle verdier tilgjengelige ved teknisk selvbetjening vil være satt til default-verdier ved nyleveranse. Når vi innfører teknisk selvbetjening så medfører det en endring i tjenesten: Tjenesten vil bli satt opp med så mange parametere som mulig satt som default. Det er opp til ISP i etterkant å endre tjenesten sine egenskaper slik han selv ønsker.

Network Abstraction

Telenors NMS-systemer tilbyr Network Abstraction for alle nordgående systemer. Det vil også gjelde dette API-et. Det betyr at API-et vil eksponere funksjoner, ikke hvordan funksjonen er implementert i nettet. Som en konsekvens av dette så vil API-et være uberørt av evt implementasjonsendringer pga endret utstyr i nettet, hvordan VULA er designet osv så lenge funksjonen fremdeles er tilgjengelig.

Telenor vil søke å unngå at endringer i nettet vil påvirke API-et og utløse utvikling på ISP-enes side.

API-beskrivelse

Forslag til API-beskrivelse. Forslaget fokuserer på å holde kompleksitet og systemavhengighet på tvers av Telenor og ISP-er til et minimum.

Nøkkel

Nøkkel for API-kommandoer vil være sambandsnummer (CircuitId). Sambandsnummer til aksesser vil overleveres til ISP-ene på samme måte som i dag. ISP-ene vil kun få tilgang til aksesser som allerede er levert via Service Delivery.

Synkront

API-et vil være synkront. Det vil returnere etter en syntaktisk og semantisk sjekk av data og oppdatering av vår database. I etterkant vil Telenor asynkront oppdatere alle involverte nettelementer.

3. linje drift har anledning til å midlertidig stenge ned Network Automation ved f.eks. arbeid i nettet. Dersom dette er tilfelle, vil systemet legge ordrene i kø og utføre dem når arbeidet er ferdig.

3. linje drift har ansvaret for å rette opp i eventuelle konfigurasjonsfeil og andre feil i grensesnittet mot nettelementene.

Protokoll og Dataformat

Protokoll vil være RESTful HTTPS. Dataformat vil være JSON formatert som UTF-8.

Primitiver

API-et vil tilby primitiver både for å endre, resette til default og lese ut hva som er satt.

GET leser ut alle tilgjengelige verdier.

Teknisk selvbetjening POST

```
GET https://[IP]:[port]/[version]/vula-tech-self-service/$circuit-id
```

POST endrer på verdi til parametere. Parametere som ikke er nevnt vil ikke endres på.

Verdier kan settes til JSON null for å resette tilbake til standardverdi.

Teknisk selvbetjening POST

```
POST https://[IP]:[port]/[version]/vula-tech-self-service/$circuit-id
{
  'parameter[1]' : value,
  ...
  'parameter[n]' : value,
}
```

Test & Diagnosis

Med Test & Diagnose så menes det at ISP-ene selv skal kunne utføre tester, resette nettfunksjoner og lese ut data for i best mulig grad å drifte egne aksesser.

Det tilbys allerede noen tester på Jara. I tillegg ser Telenor for seg å tilby API-er på andre nivåer i systemstacken.

I første omgang vil Telenor tilby et API direkte mot NMS-systemene. API-et vil kun tilby funksjonalitet på NMS-nivå. Det vil ikke være mulig å se f.eks. status på leveranser som ikke er aktivert eller en tilhørende DMF til en aksess. Dette API-et blir beskrevet nedenfor.

Telenor arbeider med å modernisere vår T&D-plattform og vil i den sammenheng se på hvordan funksjonaliteten vil kunne utvides ytterligere.

API-beskrivelse

Forslag til API-beskrivelse.

Nøkkel

Nøkkel for API-et vil være sambandsnummer (CircuitId). Sambandsnummer til aksesser vil overleveres til ISP-ene på samme måte som i dag. ISP-ene vil kun få tilgang til aksesser som allerede er levert via Service Delivery.

Synkront

API-et vil være synkront. Det vil returnere når testen er ferdig utført i nettet.

Protokoll og Dataformat

Protokoll vil være RESTful HTTPS. Dataformat vil være JSON formatert som UTF-8.

Primitiver

GET leser informasjon.

Teknisk selvbetjening POST

```
GET https://[IP]:[port]/[version]/vula-tech-self-service/$circuit-id/[kommando]
```

POST utfører en kommando, evt. med parametere.

Teknisk selvbetjening POST

```
POST https://[IP]:[port]/[version]/vula-tech-self-service/$circuit-id/[kommando]
{
  'parameter[1]' : value,
  ...
  'parameter[n]' : value,
}
```

Kommandoer

status henter online-status for et samband

```
GET https://[IP]:[port]/[version]/vula-tech-self-service/$circuit-id/status
```

Response:

```
{  
  'status': 'online'  
}
```

EKSEMPEL, detaljer kommer senere.

Performance Management

PM-API er planlagt å være et maskinelt grensesnitt til flere av Telenors informasjons og støttesystemer som tilbyr statistikk og analysefunksjoner på nettverksressurser og tekniske tjenester. Det legges vekt på at API'et skal være enkelt å ta i bruk og ha minst mulig skreddersøm da det vil kunne ha ulike brukergrupper.

Eksempel på PM data er:

- opp/nedstrøm trafikk/båndbredde utnyttelse
- opp/nedstrøm pakker,
- pakketap, link errors
- QoS køing,
- Forsinkelse (latency) og jitter
- Optiske nivåer

Tidsseriedata

PM-API vil tilby tidsseriedata. Med tidsseriedata menes data som varierer over tid, eksempel trafikkmålinger.

Tidsserie data i form av "rådata" og aggregert data

All tidsserie være på normalisert form, enten i form av rådata eller aggregert data. Med rådata menes at dataresultatet tilsvarer verdier lest direkte av tellere fra tilhørende nettverksressurser. Aggregerte data er data som er behandlet og aggregert over en definert tidsperiode, eks time, dag, uke. Med aggregater følger min, max og gjennomsnittsverdier for forespurte målepunkt.

Ressurser og tilgang for tjenestetilbydere

Tilgang til måledata for tjenestetilbydere vil gis for ressurser direkte knyttet til tjeneste solgt av tilbyder, eksempelvis vlan, kundesamband, kundeinterface/port. En sentral del av det pågående design og arkitekturarbeidet med PM-API er å sikre segregering av måledata slik at en tjenestetilbyder X, får tilgang til ressurser og tjenester solgt av X - og kun det.

Design og implementering

Detaljer rundt design og implementering av PM-API er ikke avklart. Men det legges stor vekt på bruk av åpne og standard grensesnittet støttet av de fleste 3.parts systemer. APIet vil med høy sannsynlighet eksponeres som et RESTful API på Internett. Første versjon vil være PULL basert. Der er tenkt at tjenestetilbyder skal kunne be om data for en eller flere tjenester on-demand ved GET requests. Svaret vil ha JSON formatering. Input parametre vil være en kombi av tjeneste id, eks KundeId+interface, måledata navn (eks ifInOctets), oppløsning (eks "time"), periode (eks: siste 24 timer) og verditype (eks Max). I dette eksempelet vil JSON resultatet et datasett med 24 max verdier for målepunktet ifInOctets. Gitt bruk av RESTful API vil standard HTTP respons/statuskoder benyttes for å informere klienten om status på forespørselen.

Eksempel på forespørsel:

```
GET
/stats/customerX/vlan12345/name=ifInOctets&period=24h&resolution=1h&type=max
```

Infrastruktur

API-ene mot NMS vil være tilgjengelig på internett-eksponerte IP-adresser.

Redundans/Oppetid

APIet som tilbys skal ha høy oppetid. Det vil være duplisert på to maskiner på hver sitt geografisk adskilte datasenter. Kildedata vil ha tilsvarende redundans. Det skal være mulig å gjøre vedlikehold på en av maskinene uten at det skal påvirke klienter av APIet.

Autentisering og autorisasjon

Gitt bruk av RESTful API er OAuth2 mest sannsynlig metoden som vil benyttes for autentisering og autorisasjon.

Test og utviklingsmiljø

All utvikling og testing er tenkt gjort i separat utviklings og testmiljø. Dette miljøet vil også være tilgjengelig for 3.part for testformål.

Endringshåndtering

Bugfikser og mindre kodeendringer gjøres fortløpende som del av en agil utviklingsprosess, og kode slippes gjerne i korte release sykluser. Stort sett all utvikling håndteres av et fast team med dag-til-dag oppfølging ledsaget av teamleder ("scrum master"). For større endringer/releaser som

krever tilpasning hos 3.parts konsumenter av API'et vil det sendes ut informasjon i tilstrekkelig tid før release slik at 3.parts systemer kan tilpasses før endring trer i kraft.

GDPR/Personvern

Retention regler etableres og vil være i tråd med sikkerhetskrav og lov om personvernopplysninger / GDPR