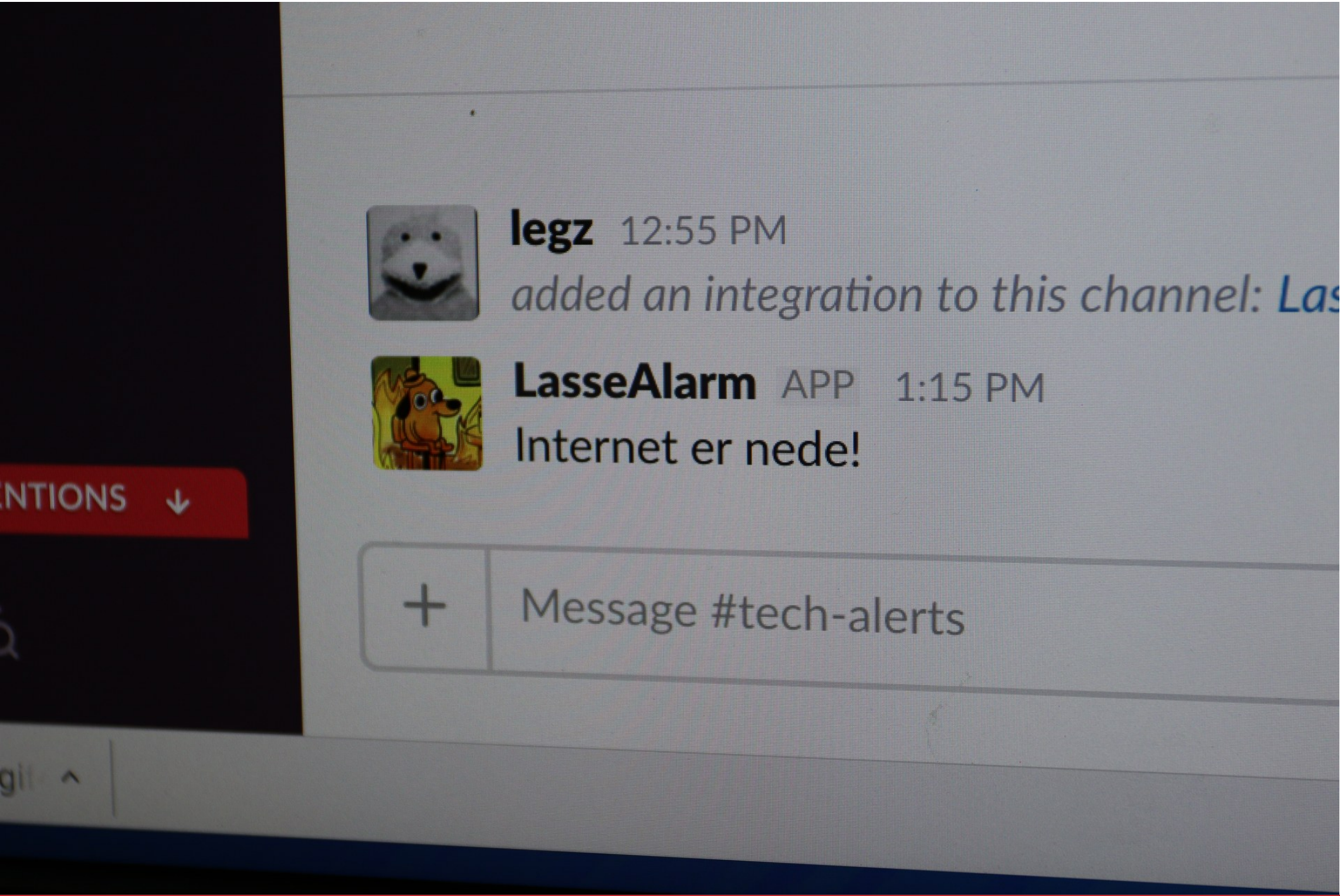
Redpill Linpro

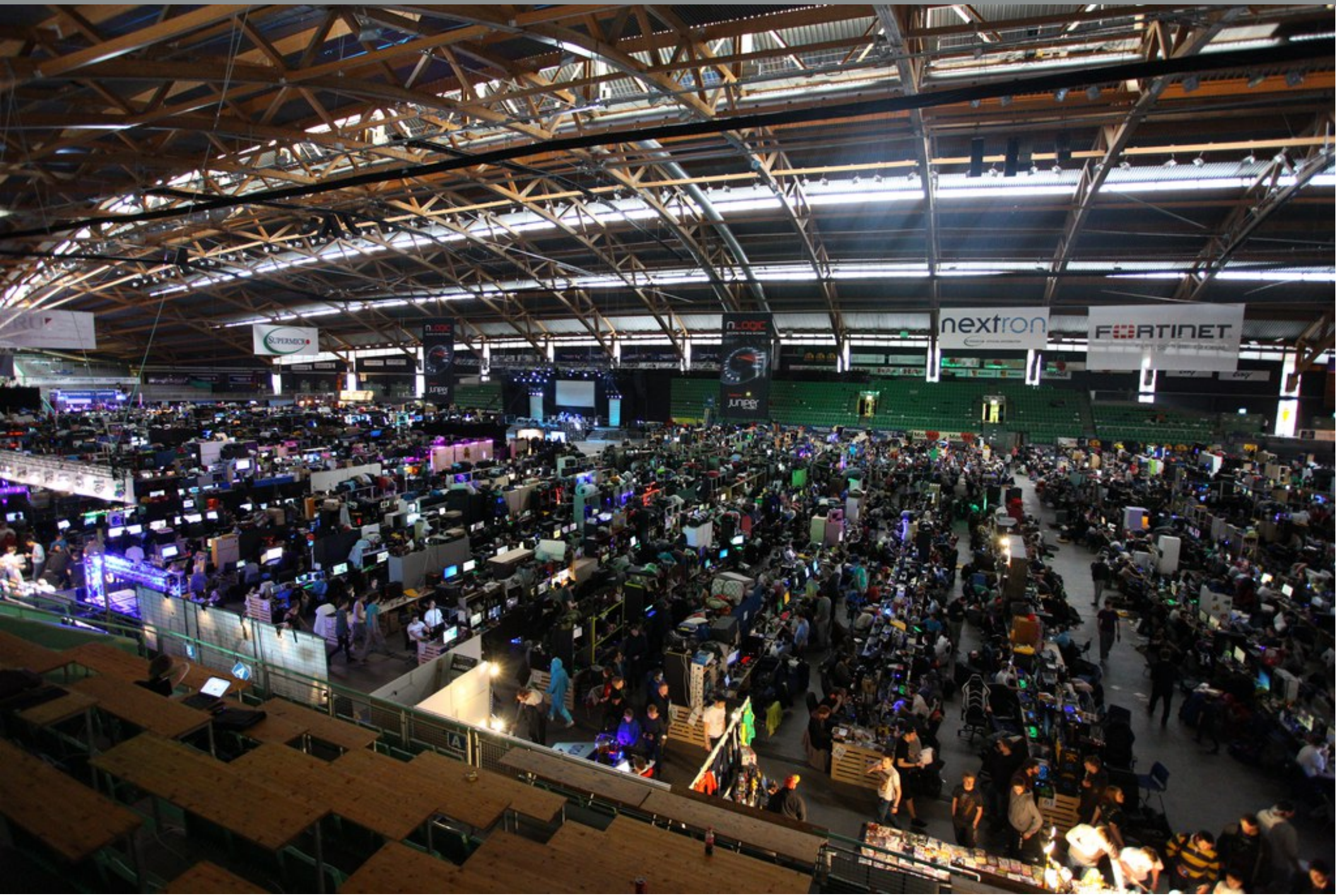
TG's overvåkning - Fra perl-ræl til mikrotjenester

Kristian Lyngstøl (The Gathering/Redpill Linpro)
Grimstad, 2018


legz 12:55 PM
added an integration to this channel: Las
LasseAlarm APP 1:15 PM
Internet er nede!
Message #tech-alerts

Thehvafforno?

- The Gathering, populært kalt TG (Theghe), er en digital festival som avholdes i Vikingskipet hver påske.
- Trekker rundt 5000 deltakere med “ukespass”, og noen tusen til er på besøk.
- Typisk alder på deltakere er 15 til 20 år
- Spilling, kreative innslag, show, boder.
- “Dataverdenens svar på Norway-cup”
- Bring your own Computer/Device/Bugs
- Non-profit



PRODUCTS • CONSULTING • APPLICATION MANAGEMENT • IT OPERATIONS • SUPPORT • TRAINING

Redpill Linpro

PRODUCTS • CONSULTING • APPLICATION MANAGEMENT • IT OPERATIONS • SUPPORT • TRAINING

Crew på TG?

- Søk crew! Opptak for TG18 er nå!
- Intenst arbeid, svært lærerikt både faglig og mer generelt.
- Vi trenger tekniske folk!
  - Tech:Net - nettverksinfrastruktur
  - Tech:Support - bygger nesten alt. “Support”-biten er missvisende, selv om de gjør det og.
- (Core:Systems - programmerere for web osv, men opptaket er ferdig)

Redpill Linpro

Jaok, men hvem er du?

Jeg er....

- Kristian Lyngstøl
- Frivillig på TG. Seniorkonsulent hos Redpill Linpro
- Tidligere Varnish-utvikler og Compiz-utvikler
- Programmerer, drifter, nettverksdude, arkitekt, osv.
- Hobbypedant. Gammel grinebitter siden tenåra.
- C/Perl/Python/PHP/Ruby/Go/Bash/TCL/Java/Awk/Postgres/MySQL/Chef/Puppet/Salt/Ansible/VIM/git/JavaScript/CSS/HTML/Autotools/Deian/RHEL/CoreOS/Kubernetes/Docker/errrr.../calc.exe/bc/more/less

Redpill Linpro

Kjapp teknisk gjennomgang

Redpill Linpro

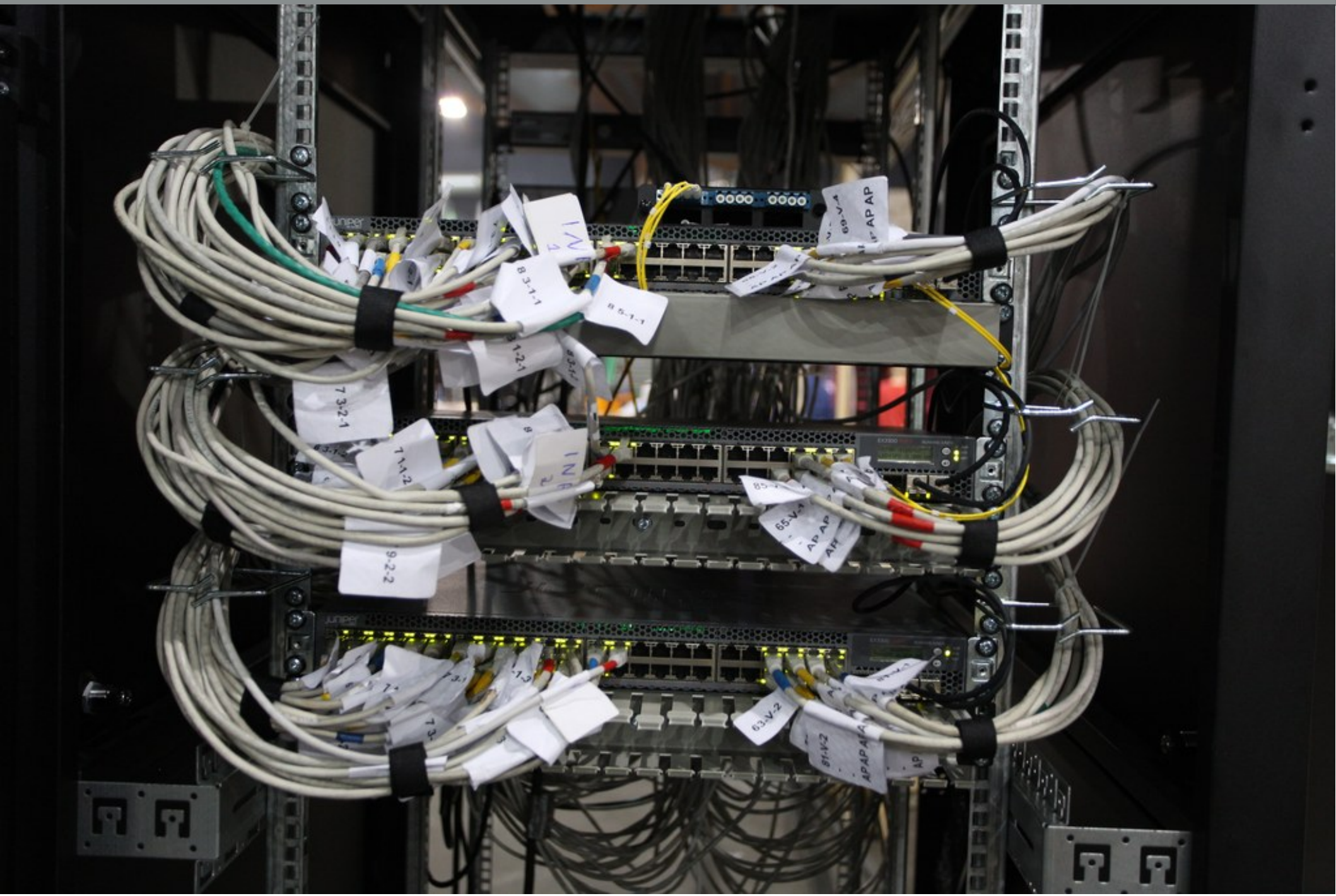
200 * Juniper EX2200 48P

Redpill Linpro

9-ish distribusjonssvitsjer

Består av 3 EX3300-svitsjer i “Virtual Chassis”

Vi eier rundt 50 EX3300-svitsjer.



PRODUCTS • CONSULTING • APPLICATION MANAGEMENT • IT OPERATIONS • SUPPORT • TRAINING

Redpill Linpro

Redpill Linpro

Wireless:

Store mengder Meru AP’er.

Etc

- Peaker på rundt 7000 aktive DHCP-leaser med unike mac-adresser (les: 7000 aktive brukere).
- Under TG16 så vi totalt 10500 unike MAC-adresser
- Redundant ring-nettverk for ymse. Både design-messig og bokstavelig - det går i ring rundt skipet - infrastrukturen er bygd av TG-folk.
- Ofte mye eksperimentell/ny/fancy hardware på lån
- Lav terskel for å teste ny teknologi, en fokus på god leveranse for deltakerene.

Redpill Linpro

Servere: Komisk overspecca Supermicro-bokser fra Nextron (TG15-TG17, tg18?)

“Vi hadde ikke en passende 15TB-lagringsboks til dere, men går det greit med denne 128TB-boksen i stedet?”

PRODUCTS • CONSULTING • APPLICATION MANAGEMENT • IT OPERATIONS • SUPPORT • TRAINING

PRODUCTS • CONSULTING • APPLICATION MANAGEMENT • IT OPERATIONS • SUPPORT • TRAINING

Redpill Linpro

Ubrukt under TG16:
To QFX10002-72Q Juniper
“Switcher” (24 100GBit porter
eller 72 40GBit ports)

Pris? 2x “herreguuuud”

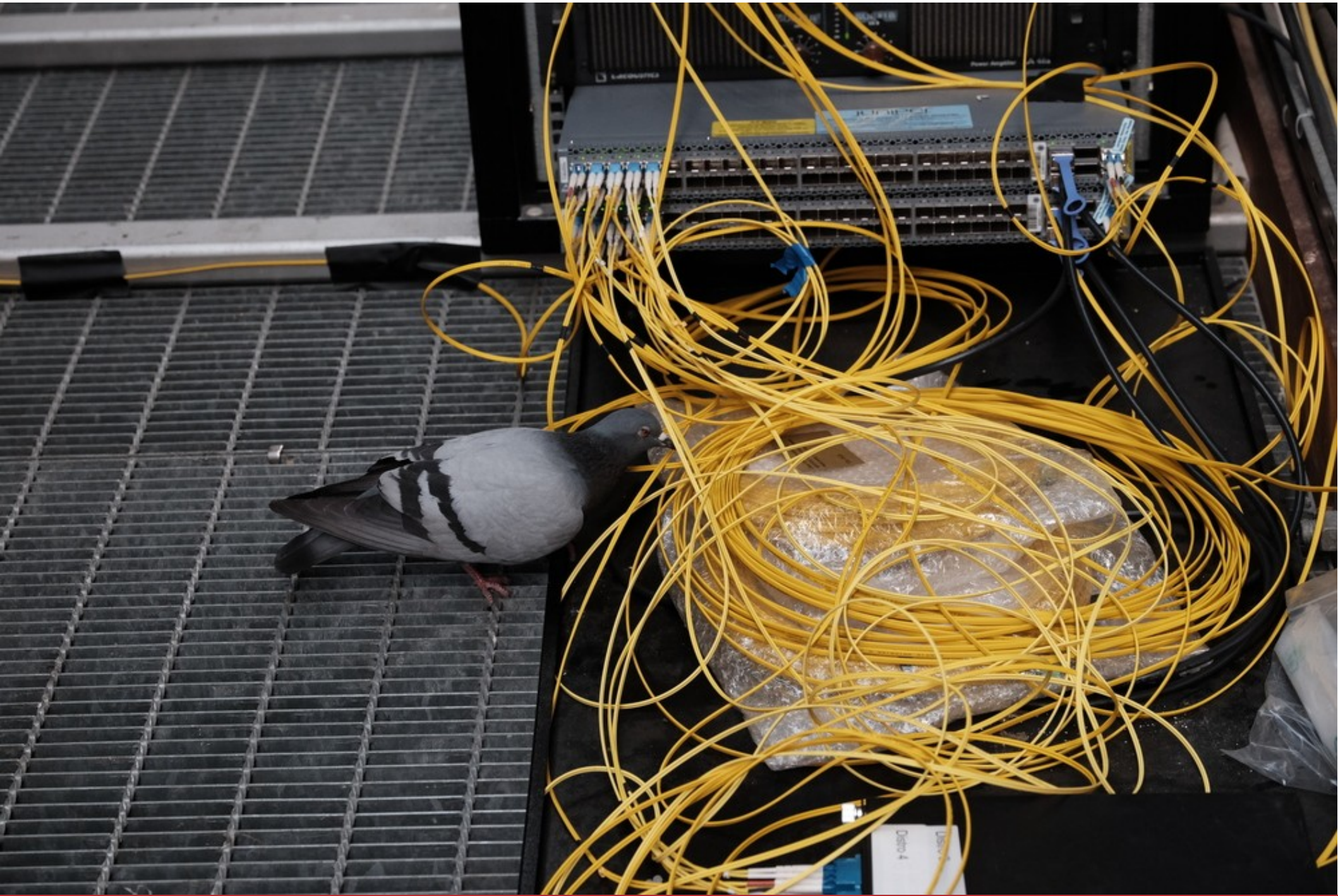
PRODUCTS • CONSULTING • APPLICATION MANAGEMENT • IT OPERATIONS • SUPPORT • TRAINING

Redpill Linpro

Core:

Juniper QFX5-et-eller-annet.

Den fine (redundante) dingsen bak haugen med fiber.



PRODUCTS • CONSULTING • APPLICATION MANAGEMENT • IT OPERATIONS • SUPPORT • TRAINING

Redpill Linpro

Brannmur, TG17:
2* SRX5800
4Tb/s dpi
2*7kW varme idle.
2*Herreguuud kr.

Redpill Linpro

Jeg er mest involvert på server-siden, overvåking, planlegging, tooling...

Redpill Linpro

(Og for the lulz)

Monitorering fram til TG15

- Bygd en samling smarte, men hacky perl-script
- Det var tett kobling mellom hva som ble samlet inn, hvordan det ble samlet inn og hvordan det ble vist.
- Å, hvil du ha serienummer også? Ny tabell, hacke litt mer på SNMP-polleren, lage noen obskøne Postgres-funksjoner (!). Kopier “ping2.pl” til “serienummer.pl”, og gjør det samme med “ping2.html”... osv
- Veldig mange bra småting som var helt umulig å ha det gøy med.

TG15

- Stort utbytte av personell.
- Undertegnede kommer inn.
- Undertegnede iverksetter en slags digital bråtebrann/ulmebrann midt under arrangementet.
- Vi beveger oss sakte men sikkert over til mer moderne løsninger.

Nøkkelord for utviklingen

- Sakte
- men
- sikkert.

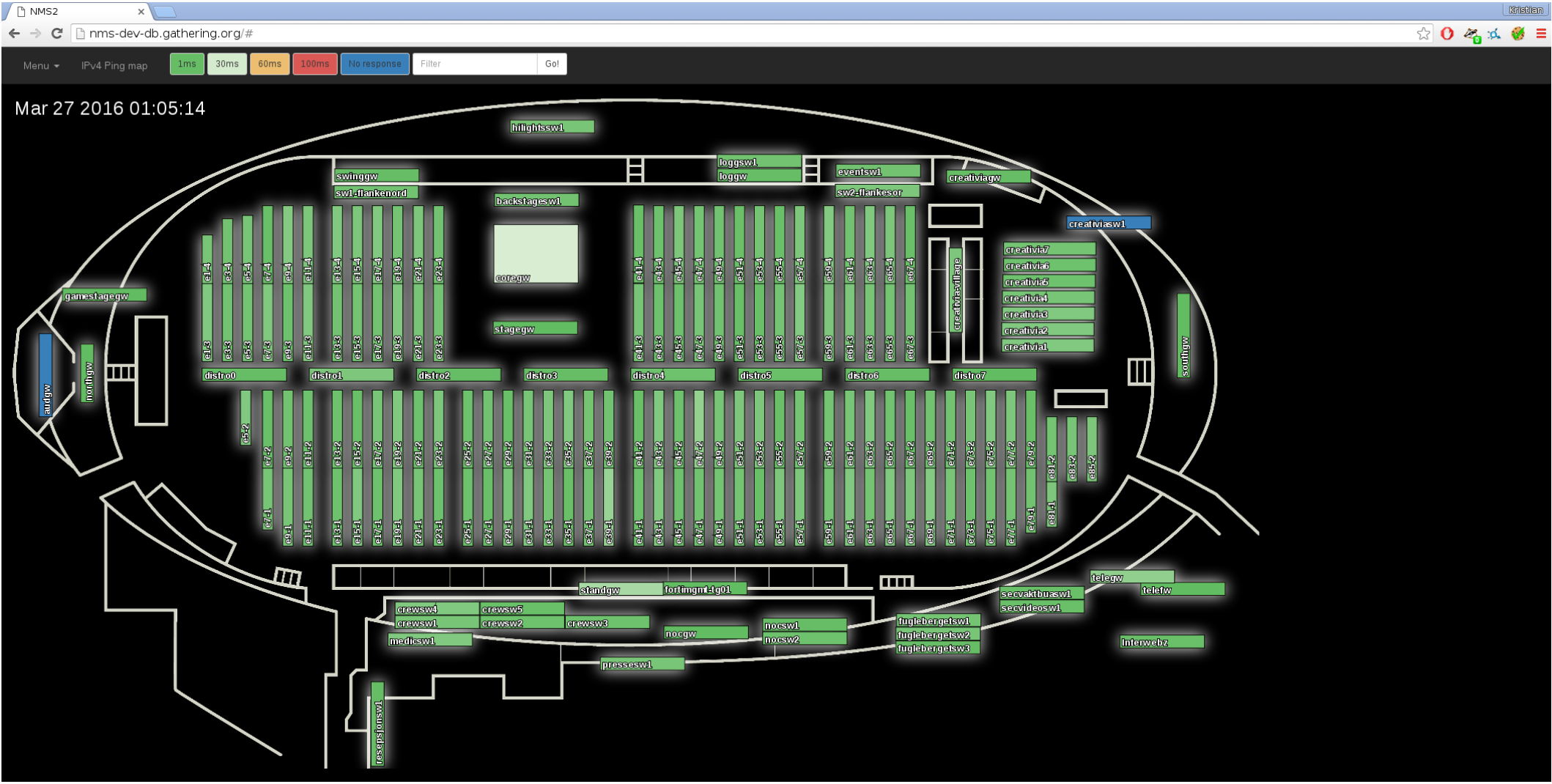
- Det var aldri noe “versjon 2”. Eller “Nå kjører vi det gamle til det nye er klart”.
- Begynte med gradvis modernisering i frontend. Litt ekstra stash her, litt bedre innsamling der. Og plutselig var det noe helt annet.

Monitoring: Gondul

- Open Source
- Små, enkle komponenter
- Home brew
- Også brukt på andre arrangement

- Ok, også brukt til deploymen. Aka “FAP”.

Redpill Linpro



To faser under drift:
1. Pre-produksjon (opprigg)

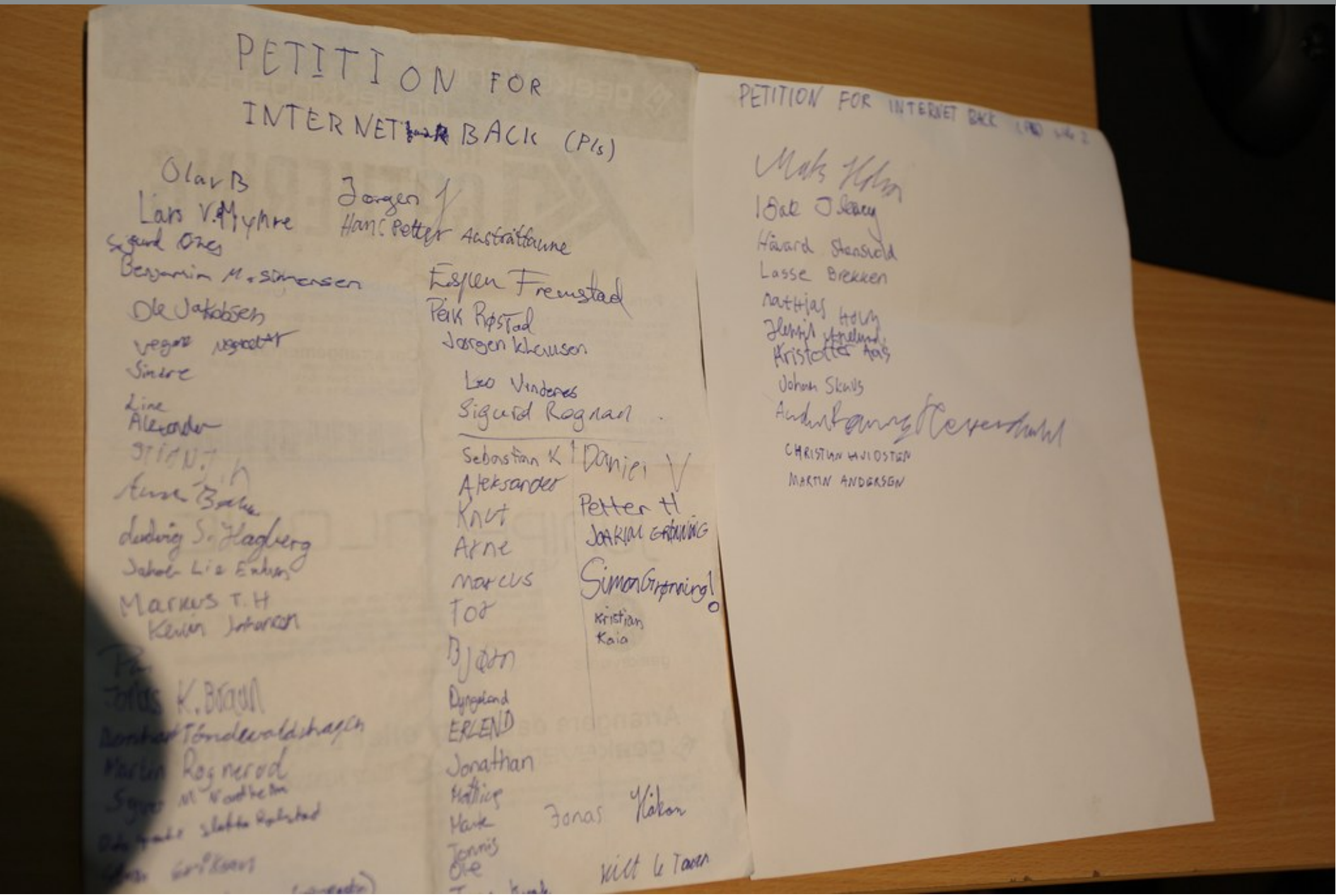
- Vi har under en uke på å sette opp alt.
- Grafer og tradisjonelle alarmer er meningsløse under opprigg.
- Men vi trenger oversikt over fremdrift.
- Dette er også første anledning til å se resultatet av mye “tørrtrening” i utvikling.
- Vi “leverer” nett til en del brukere lenge før deltakere kommer (logistikk, sikkerhet, event, osv).

Gonduls bidrag...

- Fremdriftsoversikt
- Snappe opp regresjoner
- Konfigruere alt av kantsvitsjer
- Snappe opp våre menskelige, logiske kortslutninger.

Fase 2: Produksjon

- Under en uke med fullt trykk.
- Klokka 09:00 har vi kanskje 300 enheter på nettet. 19:00 har vi 7000.
- 20 minutter nedetid er ille.
- TG15: Smooth. TG16: Vi manglet en distro-svitsj (700-deltakere) når dørene åpnet. TG17: Vi var “ferdig” to dager før innslipp.
- Tradisjonell varsling er unødvendig - altid folk på vakt, og folk roper (bokstavelig talt) om vi gjør en dårlig jobb.
- Typisk 2-4 feature-requests (på Gondul) per 8-timers skift.



Gonduls bidrag...

- Snapper opp at deltaker plugger seg inn i en ubrukt uplink port.
- Varsle når en hel seksjon (F.eks.) manger DHCP (hvorfor?)
- Merke båndbreddeissues (Tegn på at vi trenger å føre fram en ny uplink-kabel)
- Merke det når svitsjer ramler ned.
  - “Plugger du strømmen på en svitsj kan vi se det før du har sluppet strømkabelen”
- Configruere erstattningssvitsjer eller ekstrasvitsjer
- Debuging

Redpill Linpro

Umiddelbar innsikt!

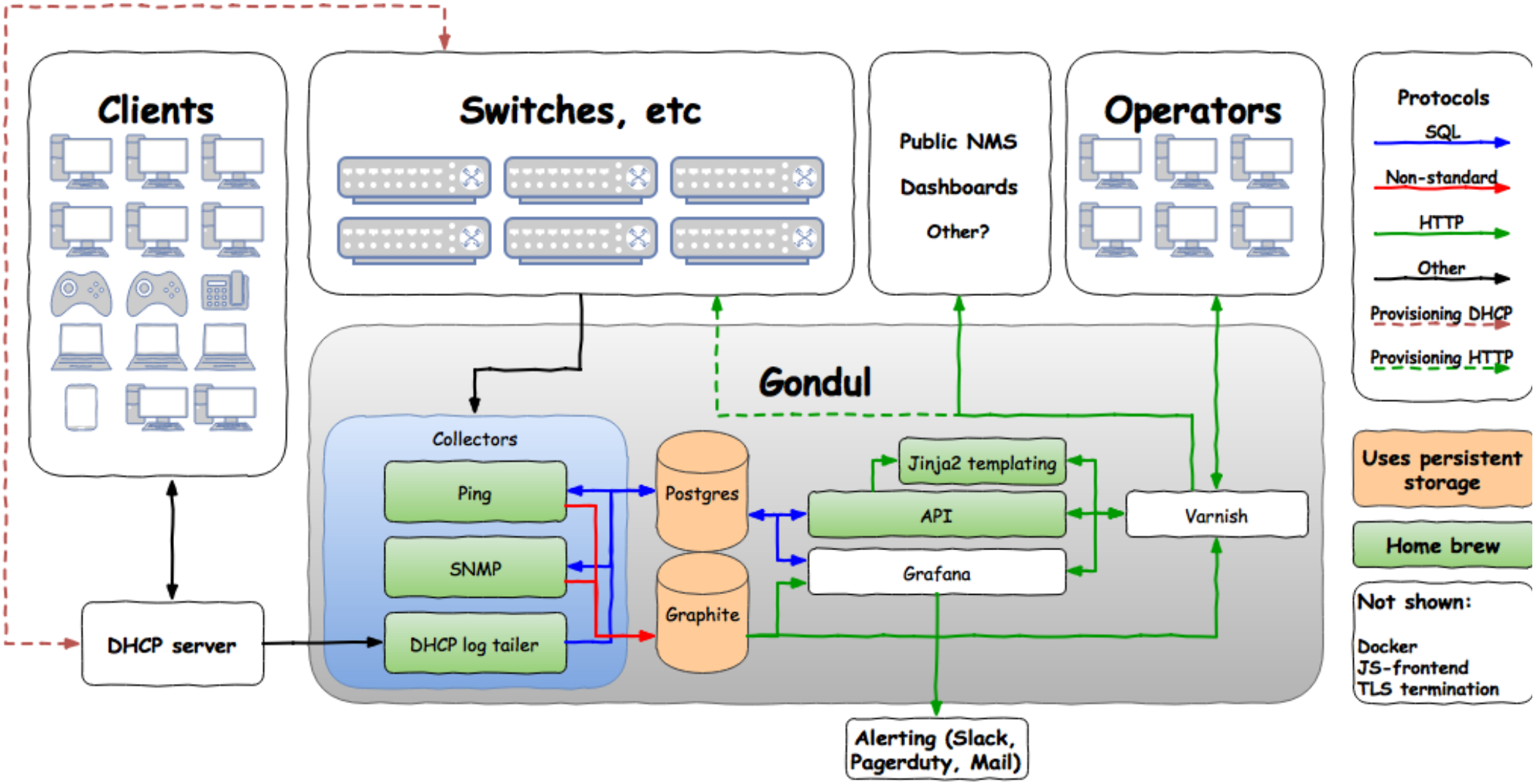
Umiddelbar innsikt

- Et enkelt kart over hele infrastrukturen.
- Lavterskel: Ingen skal trenge å få forklart særlig mye om hvordan det brukes.
- Alt brytes ned i “På en skala fra 0 til 1000, hvor bra funker dette?”
- Fargekoding basert på verste verdi.
- “Click and play”

Deploying av Gondul

- Planlegging
- planning.cpp – Arvegods som regner ut bra plasseringer for distro-svitsjer for å spare kabel-lengder, regner ut hvilke svitsjer som bør plugges i hvilke distro-svitsjer, og fordeler IP-ranger deretter.
- Mat Gondul med output fra planning
- Sett opp DHCP-servere med “option 48” eller hva det er for å gjøre “zero touch provisioning” - dette peker dem på Gondul.
- Access switches get configuration over DHCP + Gondul
- Vi generer det meste av config i/fra Gondul - mer og mer.

Arkitektur


Clients
Switches, etc
Public NMS
Dashboards
Other?
Operators
Protocols
SQL
Non-standard
HTTP
Other
Provisioning DHCP
Provisioning HTTP
Gondul
Collectors
Ping
SNMP
DHCP log tailer
Postgres
Graphite
Jinja2 templating
API
Grafana
Varnish
DHCP server
Alerting (Slack, Pagerduty, Mail)
Uses persistent storage
Home brew
Not shown:
Docker
JS-frontend
TLS termination

Bygging og kjøring: Et sabla styr

- Har måttet gå MANGE runder for å finne ut hva som funker bra.
- Docker er ikke så fantastisk som alle tror. Virkelig ikke.
- Ansible for deployment funker greit.
- Docker for enkelte tjenester, stort sett kjører ting mer native i dag.
- Mer “mikrotjeneste”-tilnærming er helt sentralt for å gjøre det mulig å faktisk jobbe med dette.

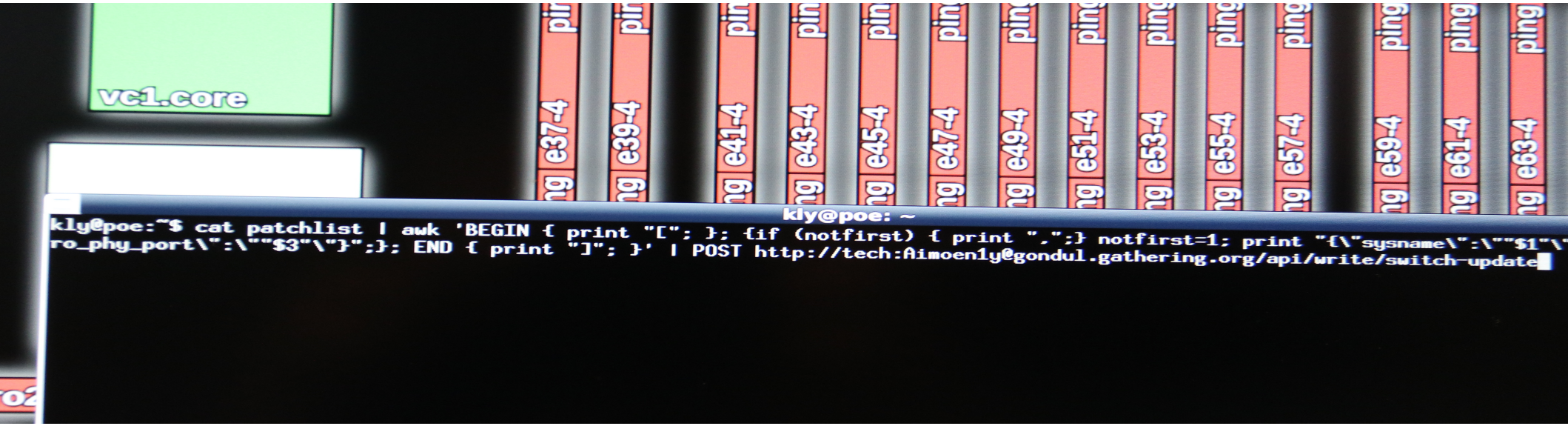
Collectors: Simple, fleksible og raske

- Ping-collector: Pinger nettet hele tiden (type: Pinger IPv4 og IPv6-adresser til hele infrastrukturen en til to ganger i sekundet). Desidert mest kritisk.
- SNMP-collector: Samler SNMP-data fra rundt 200-250 enheter. Kan kjøres i paralell. Må kunne rekonfrigueres lett.
- DHCP log tailer: Følger med på DHCP-logger og shipper info til Gondul. Må være tolerant for rot.
- Alle disse kjører som native systemd-tjenester i dag, etter flere runder med docker-rot.

Backends: Simple og generiske

- Enkelt og greit bare API-kall.
- Tonnevis av data.
- Minimalistisk rammeverk for å gjøre data cache-bar.
- Idempotent der det er relevant. Gjør det enkelt å oppdatere enkeltfelter på svitsjer.

Redpill Linpro



Cachelaget: Generisk, men smart

- Backend sender cache-control headere
- Ingen spesielle tilpassinger i cache-laget utover litt tilrettelegging for å cache innlogga brukere
- 99.999% cache hit rate, til tross for svært kort cache-tid.
- ETag, max-age, stale-while-revalidate… Alt dette er brukt av både Varnish-cachen OG browseren(!).
- Hver ny klient gir cirka 2 requests per sekund.
- Serverload er i praksis upåvirket av antall brukere.
- Vi har “public”-grensesnitt også, og får fort noen hundre samtidige brukere.

Frontend: Logikken.

- All logikken er flyttet ut til frontend/browser.
- Browseren har til enhver tid all informasjon om hele infrastrutkuren. Fordi du har råd til å vie noen få MB RAM til Gondul.
- Gradvis mer intelligent modularisering. Datainnsamling, karttegning, logikk. Alt separert.
- Flere HTML5 Canvaser osv.
- Utvikling er lett og fort. Testing på laptop, redeploye live uten nevneverdig risiko. Release-prosess? “Trykk F5”.

Kantprovisjonering: FAP/Gondul

- Før TG15 ble “FAP” skrevet. Dette var en DHCP-server skrevet i Python med integrasjon mot en MySQL-database og noe php-greier.
- Gradvis har dette blitt slått sammen med Gondul.
- TG16: Database. TG17: Alt. (Null DHCPd i python!)
- Svitsjer booter opp “blanke”, får DHCP med referanser tilbake til Gondul for å hente konfig.
- ID er basert på hvilken fysiske port DHCP-spørselen kommer fra.
- Jobber med Juniper for bug-fiksing i deres DHCP-relay.

Provisjonering

- TG17 sin FAP var “klassisk” Gondul med et lite interface i Go skrevet av Lasse Haugen - alt jobber mot Gondul sine APIer.
- Ny kantsvitsj? Legg den til i Gondul. Skriv inn hvilken distro-svitsj og -port den kobles til. Løp ut, koble svitsjen opp, skru på strømmen, ta deg en velfortjent pause.
- Gjør det enkelt for “hvem som helst” å sette opp nye svitsjer uten at det blir cowboy-preg.
- Templatingen kan også brukes til mye annet snacks - det vil komme mer med årene håper vi/jeg.

Generisk templating?

- Enkleste use-case: Svitsjkonfig.
- Kan også brukes til: Konfe DHCP-serveren. Lage rapporter fra inventory. I grunnen alt du vil og kan med jinja2.
- POST /... templating - Folk kan mekke template på egen laptop og kompilere den med gondul.
- GET /...?ekstra=verdi&sw=e13-1 templating - Fellestemplates (f.eks. svitsjkonfig, distro-config, brannmuroppsett, osv).

Vi er ikke ferdig

- Graphite er døende, og sviktet totalt under TG17
  - (Og jeg hadde ikke tid til å fikse det der og da)
  - Byttes ut, sansynligvis med Influx, for TG17
- FAP - Før TG17 ble det lagd TO templatingsystemer for å støtte FAP. Den i GO ble brukt. Skal forenes.
  - Viser hvor viktig det er å ha fleksible systemer: Dumt at det ble to. Bra at det var så uproblematisk å erstatte den som ikke nådde opp.
- Mer og mer av “forarbeidet” skal inn i Gondul
- Eksperimenter med ansible mot svitsjer.

Redpill Linpro

Ting som er skrota/lagd på nytt de siste 2-3 årene:

SNMP, PING og DHCP-taileren
Databasemodellen (JSON FTW)
Backend -> API
API-modell.
Frontenden.
Templating.
Grafing.
Byggesytemer/deployment.
Navnet!

Redpill Linpro

Vi har nesten aldri truffet helt på første forsøk.

Å annerkjenne dette er det viktigste vi som utviklere gjør.

Redpill Linpro

Vi har fokusert på å lage små, enkle komponenter som gjør en ting og gjør det bra.

Redpill Linpro

KISS