

System-administration

De bitre gamle mænds klub

Lars Fischer

CTO, NORDUnet

AaU, februar 2006



Før pause

- Lidt om mig – og lidt om NORDUnet
- Hvad er systemadministration
- Arbejdsområder for systemadministration
- Design og mål
- Værktøjer
- Organisation

Efter pause

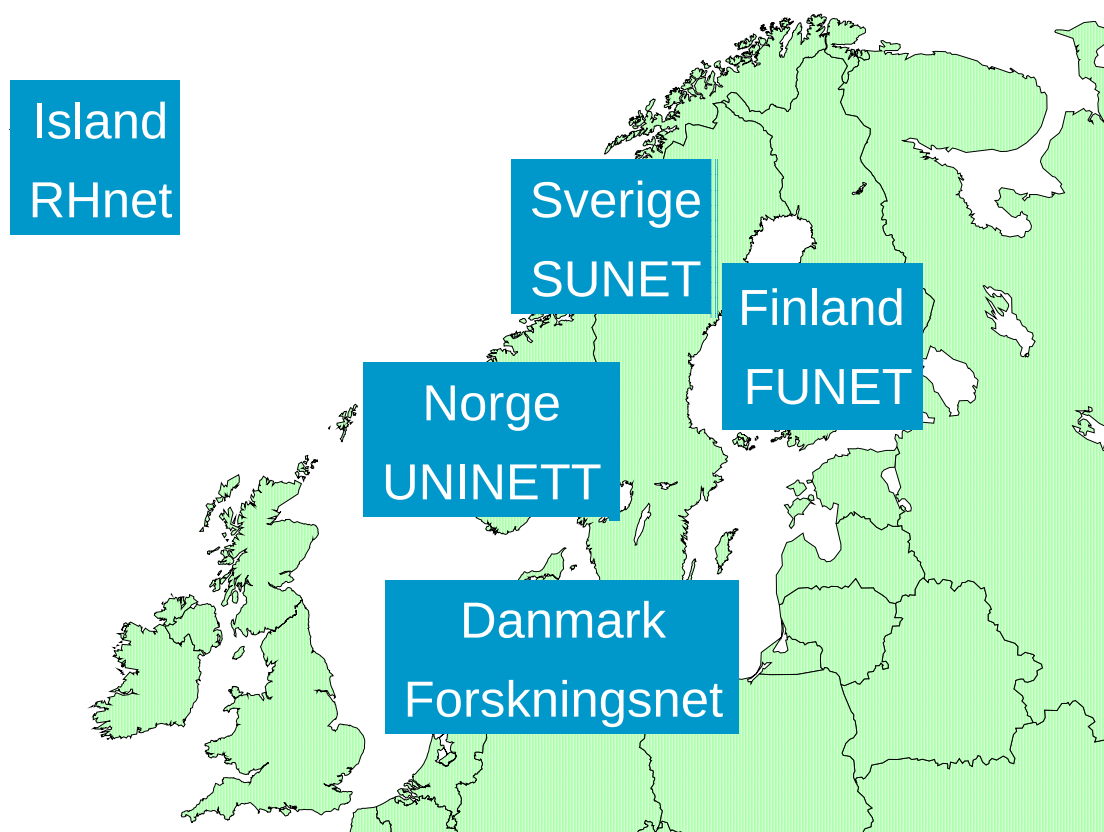
- Hvordan bygger man pålidelige og stabile IT-systemer?
- Hvad er en systemadministrator for en slags person?
- Hvad skal en systemadministrator kunne
- Hvilken adfærd har en systemadministrator?
- Findes der en teknisk løsning?
- Lidt om storage
- Metoder og systemer?
- Ledelse

Lars

- Civ.Ing. AUC (!) 1988, MBA DTU 2001
- Junta 1985-1988
- 20+ år med drift af UNIX og IP-net
 - ...og alt det andet der nu hører med
 - Har levet med nattevagter og pagere
 - Har set ting gå galt på rigtig mange måder.
- Teknisk chef i Tele2, COLT
 - Drift af IT installationer, netværk, mobilnet
 - Design a komplekse systemer
 - Lederlse af driftsfunktioner
- Pt. CTO i NORDUnet



NORDUnet – et netværkssamarbejde



NORDUnet

- 25 års samarbejde om netværk til forsknings og uddannelse
- 5 lande, 1 netværk, 1 stemme
 - At gøre sammen hvad vi ikke kan gøre alene
 - Representere nordiske netværk internationalt
- 25 års med state-of-the-art netværk
 - Første i Europa til at bruge IP
 - Mange og stærke internationale relationer
 - 1 million brugere



Governance

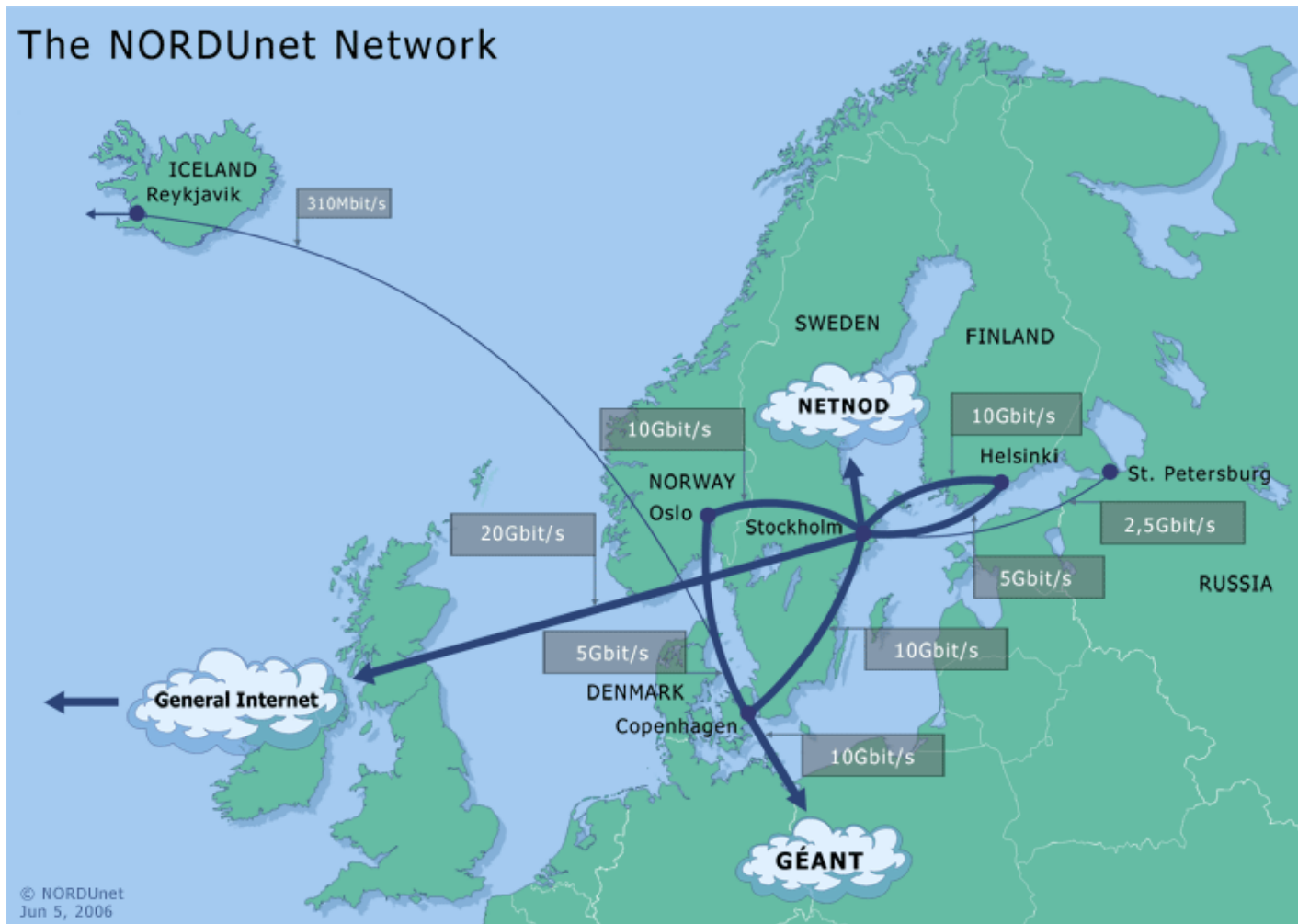
- Dansk A/S
- Aktionærerne er nordiske institutioner
 - DK – Forskingsministeriet
 - FI – Undervisningsministeriet
 - IS – Islands Universitet
 - NO – UNINETT (forskningsnet)
 - SE – Vetenskabsrådet (forskningsråd)
- Et bestyrelsesmedlem fra hvert land
- Financieret af de nordiske lande efter GNP model

Organisation

- Hovedkontor i København
 - Ledelse
 - Administration
 - Tæt på Københavns lufthavn
- NOC, NMS i Stockholm
- Låner personale af de nationale forskningsnet og outsourcer i de enkelte lande
- Personale: 35, inkl. NDGF

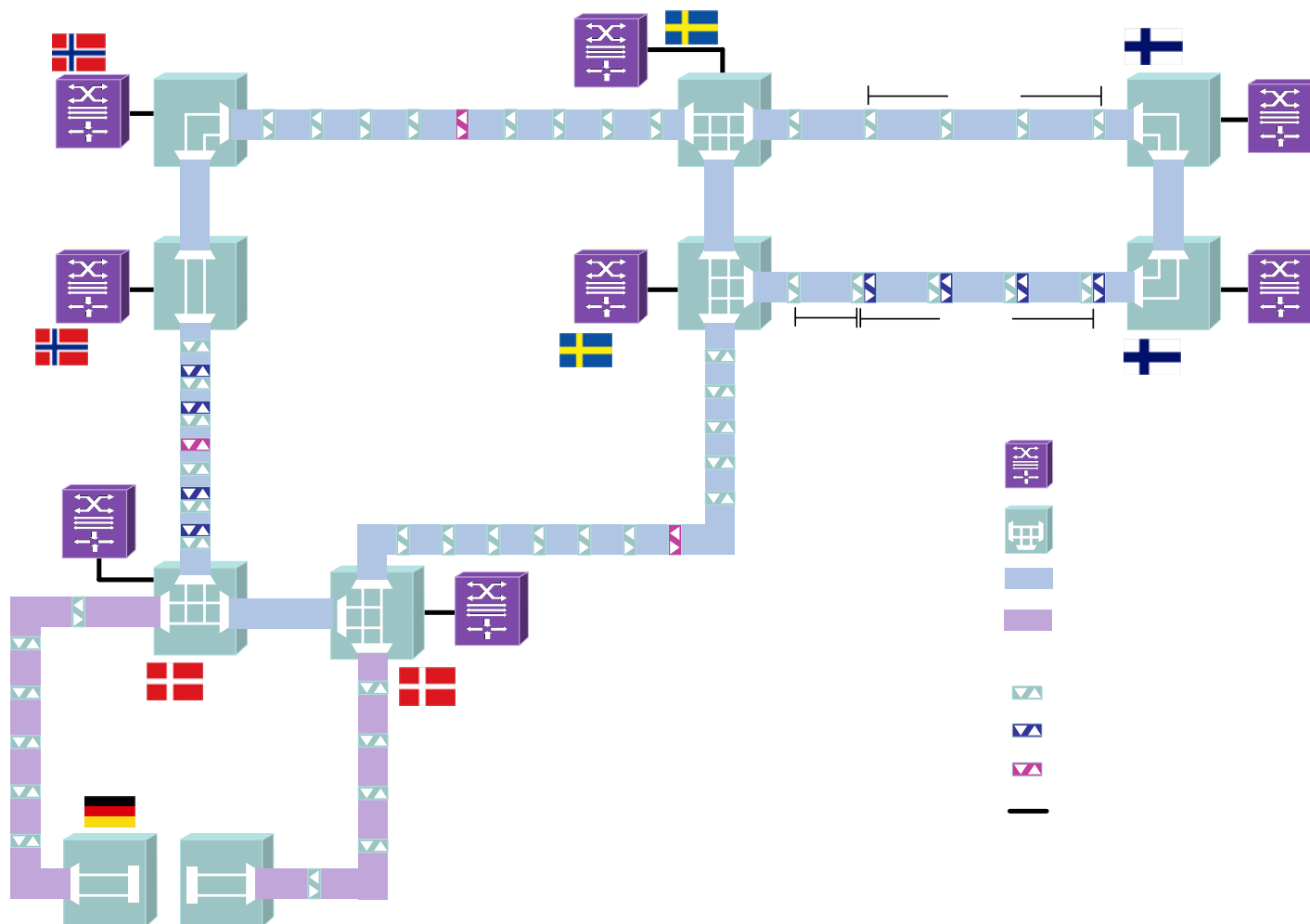
NORDUnet IP Netværk

The NORDUnet Network



```
10101 11110
011011 10101
1001101 10010
0101010010001
1111010101001
11010 0101010
00000 101010
01100 01101
```

NORDUnet fiber netværk



USIT

```
10101 11110
01101 10101
100110 10010
0101010010001
111101010001
11010 0101010
00000 101010
01100 01101
```

Hvad er systemadministration?

- At sikre pålidelighed og tilgængelighed af (komplekse) IT-installationer
- At sikre data i bredest mulig forstand
- At sikre en fornuftig balance mellem ydelse og omkostning

Arbejdsområder (I)

- Facility management
 - Site management
 - Strøm
 - Cabling
 - Køling
 - Fysisk sikkerhed
- Projekter og planlægning
 - Installation, system introduktion, idriftsættelse
 - Opgraderinger
 - Revision
 - Planlægning, faseopdeling, styring, osv.

Arbejdsområder (II)

- Logistik
 - Leverandører
 - Reserverdele
 - Supportaftale
- Design
 - Systemdesign
 - Dimensionering, performanceanalyse
 - Lagring
 - Recovery-strategi

Arbejdsområder (III)

- Support
 - Brugere
 - Applikationer
 - Projekter
- Teknik
 - Systemdrift – overvågning, performance, vedligehold, fejlretning
 - Netværksdrift - ditto
 - IT sikkerhed
- Brandslukning

Facility Management

- Det er ikke trivielt
- Hvor mange sites? Afstand? Bemanding?
- Racks: Dybde, bredde, luft omkring, pladsudnyttelse
- Strøm
 - Dimensionering, peak, kabler, fordeling, sikringer, antal kilder?
 - UPS, dieselgenerator(er)
- Cabling: netværk (mange?), storage, ducts, fleksibilitet, labels(!)
- Køling:
 - Redundans?
 - 1W strømforbrug = 2W køling (= omkostning)
- Brandslukning?
- Gulvets bæreevne?
- Adgangskontrol, tyverisikring, har vi adgang i nat kl. 3?
- Det er slet ikke trivielt

Eksempler på FM problemer

- Brand
- Strømafbrydelse
 - 1 time, 4 timer, 1 dag, 1 uge
- Tyveri, hærværk, sabotage
- Byggeri, vejarbejde, landmænd, ...
- Gulvvask en etage oppe
- Jordskælv, oversvømmelse, storm, ...
- Løsgående fly

Projekter

- Projekter og daglig drift
- Typiske projekter
 - Installation af ny cluster server
 - Opgradering af en række servere med kritiske applikationer, med minimum nedetid
 - Opbygning af en ny site
 - Opgradering a strømsystem
 - Udskiftning af backup platform
 - Installation af storage system
- Projektstyring, projektdeltagelse
 - Planlægning
 - Konsekvensanalyse
 - Risikoanalyse
 - Samarbejde med andre funktioner, kunder, etc.



Design

- Systemdesign, systemplanlægning
 - Det er her systemernes skæbne afgøres!
 - Det er her der er brug for erfaring og teknisk kompetence
 - Det er her der er brug for ingeniører!
- Levetid af system?
- Samspil mellem design og organisation
- Design skal laves af drift-folk, med drift for øje

Designspørgsmål

- Komplexitet, integration, interaktion?
- Hvor kritisk er det? For hvem?
 - Hvad koster svigt?
 - Redundans? Site-redundans?
 - Vedligehold uden nedetid? Opgradering?
 - Hvilket slags svigt? (Utilgængeligt, tab af data, uautoriseret adgang)
- Hvor mange steder skal det bruges fra?
- Omkostninger
- Brug af standard-systemer?
- Osv. Osv.

Designmål

- At gøre drift mulig
 - Log-information, historik
 - Adgang til status
 - Synlighed
 - Mulighed for tuning
- Så enkelt som muligt
 - ...men ikke enklere
- Pålidelighed, forudsigelighed, overskuelighed
- Integration med OSS, arbejdsgange

Systemdrift

- Overvågning, systemstatus
- Performance-måling, tuning
- Løbende vedligehold
 - Løbende forbedringer
 - Change management
- Fejlretning (hw, sw, ...)
 - Incident response
- Operations Support Systems
 - Overblik
 - Integration
 - Roller

Værktøjer

- Drift er ofte meget værktøjstungt
 - Overvågning & Alarmer
 - Logning
 - Change management / config management
 - Tickets – interne og eksterne
- Typer af værktøjer
 - Integrerede vs. enkeltstående
 - Selvudviklede vs. udefra

Organisering af drift

- Klassisk
 - 1st level
 - 2nd level
 - 3rd level
 - Støttefunktioner: hardware, logistik, osv.
- Funktionsopdelt
 - Platforme (Unix, Windows, ...)
 - Funktioner (netværk, backup, ...)
 - Specialer (SAP, Oracle, ...)

Kan man blande drift og udvikling

- Næh...
 - ...men man gør det tit alligevel
- Hvorfor overlap mellem drift og udvikling?
 - Hand-over
 - Systemer i "permanent" udvikling
 - Deling af personale eller ressourcer
- To problemer
 - Udviklerne får aldrig tid til at udvikle
 - Udviklerne tænker i hvordan det virker, ikke i hvordan det går galt

PAUSE



```
10101 11110  
011011 10101  
1001101 10010  
0101010010001  
1111010101001  
1101010101010  
00000 101010  
01100 01101
```

Pålidelige og stabile systemer

- Hvordan bygger man pålidelige og stabile IT-systemer?
 - Det gør man ikke!
 - Man kan bygge med pålidelighed og stabilitet for øje
 - Man kan bygge med fejlretning og vedligehold for øje
- Pålidelighed og stabilitet er altid et grads-spørgsmål



To slags IT-folk

- Dem der tror på, at man kan få IT-systemer til at virke
 - Udviklere
 - Sælgere
- Dem der ved, at IT-systemer aldrig (helt) kommer til at virke (som forventet)
 - Systemadministratorer

De bitre gamle mænds klub

- Systemadministratorer er besværlige
 - De stiller spørgsmål
 - De er skeptiske
 - De siger "men hvad nu hvis...?"
- Fordi
 - De ved af erfaring, at det går galt
 - Det skal kunne repareres igen
 - De tænker i svagheder
 - De tænker i fejl-scenarier
 - De syn's det er en udfordring at finde svagheder
 - De syn's det er udfordrende at finde og rette fejl

Hvordan en sysadm tænker

- Stillet over for et nyt system tænker en systemadministrator
 - OK, hvordan går det her galt?
 - Hvordan ser vi, det er gået galt?
 - Hvem får det ud over?
 - Hvad gør vi, når det går galt
 - Er der de ting jeg skal bruge for at fixe problemet?
- En sysadm tænker ikke så meget over, hvordan det virker når det virker
 - Det er jo alligevel ikke meget sjov ved systemer når de virker



Drift – et spørgsmål om holdning

- Tag udgangspunkt i at ting går galt
- Tænke I fejl-scenarier
 - Lav risiko-analyse
 - Vid på forhånd, hvordan scenariet skal håndteres – det er for sent at tænke når fejlen er opstået
- Tænke i fejlretning
 - Det skal være nemt at finde & rette fejlen
 - Det skal være nemt for alle!
- Lad være med at blive for begejstret
 - Tænk alle de små ting igennem
 - Tænk over tid – især for ting vi ikke selv har kontrol over



Eksempel: Hvor går kablerne?

- Redundans er ikke bedre end afstanden mellem kablerne
- Hvad med de andres kabler? Dem vores leverandører bruger?
- Det går osse galt i stor målestok
 - Et Sjællands eksempel: landmænd ved Holbæk
 - Et svensk eksempel: Kista uden strøm
- Husk: nogen fejl tager lang tid at løse
 - Et eksempel i Herlev: igen netværk i 6-8 uger

Valg og konsekvenser

- Ingen valg uden konsekvenser
 - Mange valg forfølger os i årevis
- De ting vi gør får stor betydning for andre
 - Så tænk på dem
 - En nem beslutning nu kan skabe rigtig meget arbejde andre steder
 - Beslutninger med indbyggede, fremtidige fejl
 - Hvem retter fejl kl. 3 om natten
- Et københavnsk eksempel:
 - Når man først har lagt metroen over på jorden, så er den svær at grave ned
- Et simpelt eksempel: log rotation



Ambitions-niveau?

- Man kan altid gøre det bedre
 - Der er altid noget mere man kan gøre
 - Der kan altid købes endnu smartere udstyr
 - Man kan altid gøre ting på måder der ikke generer brugerne så meget
- Hvad er behovet?
 - Hvem betaler?
 - Hvad er omkostningen ved at tage færre hensyn?
 - Hvor meget nedetid har vi råd til?
 - Sund fornuft: beskyt mod de rigtig dyre fejl, ikke mod de fejl der kan undgås med sexet hardware
- Husk: IT er altid en omkostning!



Eksempel: en opgradering

- Planlægning
 - Valg af tidspunkt: Det går sikkert galt, så vi gør det kl. 03:00 så vi har tre timer til at få det op igen
 - Konkret plan for hvad der skal gøres: vi skal ikke tænke under pres, og vi skal ikke improvisere mens systemer er utilgængeligt
 - Roll-back plan
 - Roll-back points! (tid, hændelser)
 - Sanity-check planen inden vi går igang
- Udførelse: vi holder os til planen

Hvordan man skifter en generator...

- En lille anekdote om hvordan det kan gå, når man skal udskifte sin diesel-generator

Adfærd

- Sund fornuft
 - Skepsis
 - Holder hovedet koldt
 - Overblik
 - Holder alle muligheder åbne
- Tænker, tænker lidt mere
 - Og handler så
- Spiller på den lange bane
 - Systemet skal osse kunne drives når jeg ikke er her
 - IT systemer lever forbavsende længe

Hvad skal en sysadm kunne (I)

- Samarbejde
 - Med brugere
 - Med leverandører
 - Med support-funktioner
 - Med andre tekniske funktioner
 - Med andre systemadministratorer
 - Lade være med at gøre det hele selv

Hvad skal en sysadm kunne (II)

- Planlægge
 - Store opgaver
 - Gentagne opgaver
 - Tænke til slutningen!
 - Konsekvens-analyse
- Projektarbejde, projektorganisering
- Systemdesign

Hvad skal en sysadm kunne (III)

- Beherske sine platforme
 - Operativsystemer
 - Middleware (AAI)
 - Applikationer (Oracle, SAP, sendmail, ...)
 - Værktøjer (OSS, Backup)
 - Storage
- Netværk – solid IP indsigt er et must!
- Scripte / programmere
 - Kunne lave sine egne værktøjer
 - Kunne hacke sig ud af en krise

Tekniske løsninger...

- Tekniske løsninger for øget pålidelighed
 - Nødstrøm, dieselgenerator
 - Intern redundans
 - Clusters
 - Multi-site redundans, clustering
 - Distribuerede systemer
 - Fail-over software
 - Grid-system-design (Google!)

...men hjælper de?

- "A distributed system is a system where something I want to do is impossible because a computer I did not even know existed is down"
- Leslie Lamport

Komplekse løsninger

- Tekniske løsninger til stabilitet og pålidelighed øger kompleksiteten
 - Der er mere der kan gå galt
 - Fejlene er mere komplicerede
 - Risiko-analyse er svært
- Høje krav gør kompleks teknik nødvendig
 - Men man skal stadig tænke som en systemadministrator
 - Kompleks teknik gør det blot mere udfordrende – og mere nødvendigt – at være systemadministrator

Pas på falsk tryghed

- Der er to måder at designe et system: at gøre det så enkelt, at det er indlysende at det virker, eller at gøre det så kompliceret at det ikke er indlysende at det ikke virker.
 - frit efter C.A.R.Hoare

Google teknik

- Tag for givet at hardware fejler
 - Byg din software til at overleve
 - Byg din software til at fungere, selvom dele af hardware er ude af drift
 - Husk: man kan i netværk, så hvorfor ikke med servere?
- Platformsstandardisering
 - Mange ens systemer
 - Udskiftelige komponenter
 - Standard hardware



Storage

- Lagring som design-driver
 - Lagring er ofte mere kritisk en beregning
 - Lagring er tungt og kostbart
 - Lagring kræver sikkerhed (beskyttelse, tilgængelighed, ingen tab, ...)
- CPUer er nemme at erstatte
 - Det er data ikke
- Krav til lagring styrer (ofte) design
 - På samme måde som når vi koder – datastruktur komme først



Virtualisering af storage

- Valg af niveau
 - Virtualisering af diske (traditionel SAN)
 - Virtualisering af filsystemer (grid storage)
 - Virtualisering af persistente objekter (google)
- Sikkerhed, redundans, tilgængelighed
 - Kan tænkes og designed på alle niveauer
 - Er ofte dikteret af den valgte software – meget standard software idag forventer diske eller til nød virtuelle filsystemer
- Vær pragmatisk – sund fornuft



Storage teknik - eksempler

- SAN: virtuelle diske
 - Opløser integration af server og diske
 - Spejling på disk niveau
 - Site-redundans på disk niveau
 - Men: afstand? SAN-protokoller? Komplexitet? Omkostning? Flexibilitet?
- Backup og tilgængelighed
 - Snapshots
 - Serverless backup
 - Rullende backup

Tænk radikalt?

- Storage er kostbart
- Storage behandles stadig typisk som (meget avancerede) diske
- Virtualisering af diske er en dyr vej at gå
 - Det virker, men det er tungt og kostbart
 - Det binder os til hardware
- Virtualisering på højere niveau
 - Muliggør anvendelse af standard-hardware
 - Gør hardware udskiftbar & underordnet (næsten)
- Men: din organisation skal være gearet til at drive den løsning du vælger

Metoder og Systemer

- Der findes en række metoder til drift, f.eks. ITIL
- Disse metoder handler primært om
 - Garantier, i situationer med leverandør / kundeforhold (outsourcing)
 - Repeterbarhed, forventningsstyring
 - Systematisering af sund fornuft

Ledelse (I)

- Ledelses-spørgsmål
 - Organisering
 - Metoder
 - Værktøjer – egne eller købte?
 - Drift vs udvikling?
- Kultur
 - Skepsis – og entusiasme
 - Sund fornuft – og risikovillighed
 - Ansvarlighed – og faglig stolthed
- Kompetence-kultur
 - Tillid, indlevelse, agilitet
 - Systemadministration skal være sjovt

Ledelse (II)

- Drift skal tages alvorligt af ledelsen
 - Det er virksomhedens overlevelse det handler om
 - Hvis ledelsen ikke tager det alvorligt, ryger den sunde fornuft – både med alt for avancerede løsninger og med ansvarsløshed
 - Eksempel: en Diesel-generator i Luxembourg
- IT er en omkostning
 - Og drift er dyrt
 - Det er et ledelsesansvar at vælge det rette niveau
 - Ledelsen skal ofre det der skal til – og ikke mere
- HUSK: IT doesn't matter

Spørgsmål?

<lars@nordu.net>