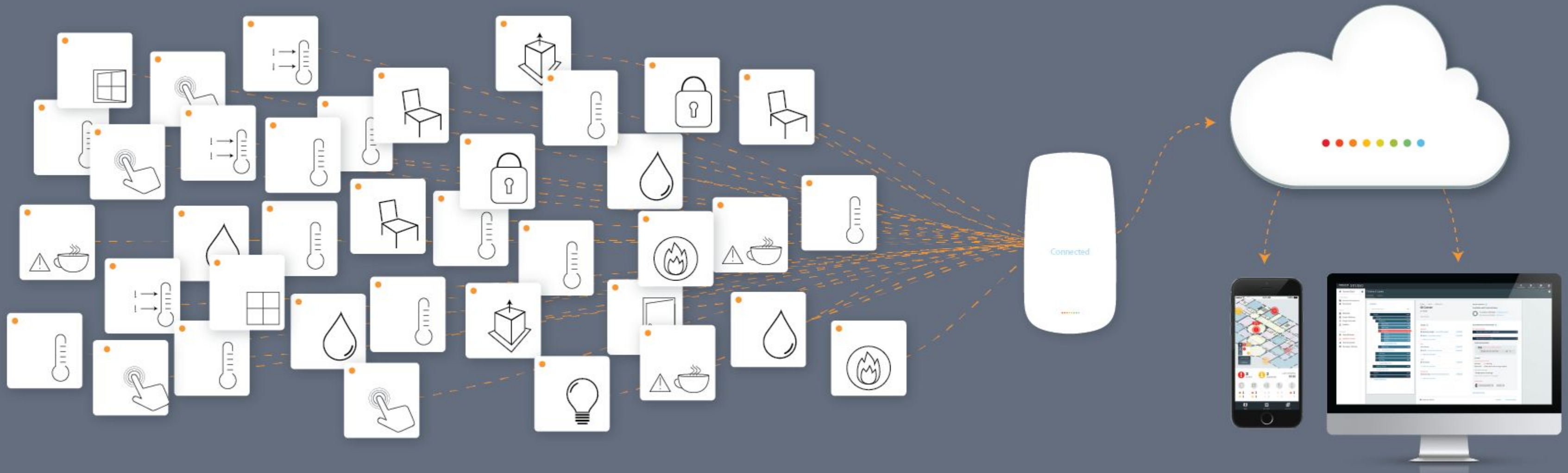
DISRUPTIVE
TECHNOLOGIES

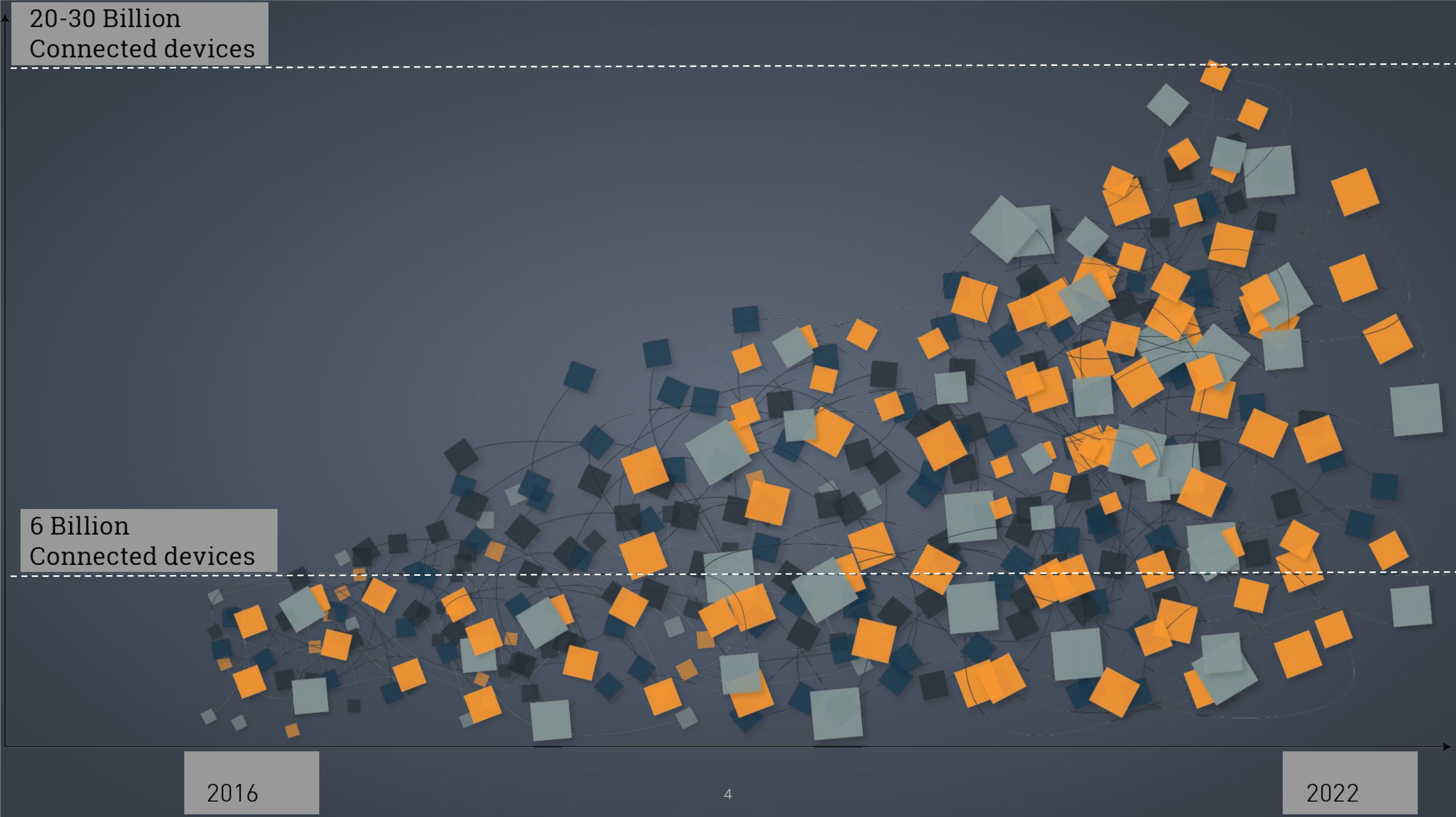
“Skyen-først arkitektur” (CNA)

Hva er det og hvorfor bry seg?

av Olve S. Hansen

Sensing As a Service




20-30 Billion
Connected devices
6 Billion
Connected devices
2016
2022

Def: Software architecture

“Software architecture encompasses the set of significant decisions about the organization of a software system including the selection of the structural elements and their interfaces by which the system is composed;
behavior as specified in collaboration among those elements;
composition of these structural and behavioral elements into larger subsystems;
and an architectural style that guides this organization.

Software architecture also involves functionality, usability, resilience, performance, reuse, comprehensibility, economic and technology constraints, tradeoffs and aesthetic concerns.”

Booch et al

Def: Software architecture

“Software architecture encompasses the set of **significant decisions about the organization** of a software system including **the selection** of the structural elements and their interfaces by which the system is composed;

behavior as specified in collaboration among those elements;

composition of these structural and behavioral elements into larger subsystems;

and an architectural style that guides this organization.

Software architecture also involves functionality, usability, resilience, performance, reuse, comprehensibility, economic and technology constraints, tradeoffs and aesthetic concerns.”

Booch et al

Def: Software architecture

“The highest-level breakdown of a system into its parts; the decisions that are hard to change; there are multiple architectures in a system; what is architecturally significant can change over a system's lifetime; and, in the end, architecture boils down to whatever the important stuff is.”

Martin Fowler

Def: Software architecture

“The highest-level breakdown of a system into its parts; the decisions that are hard to change; there are multiple architectures in a system; what is architecturally significant can change over a system's lifetime; and, in the end, architecture boils down to whatever the important stuff is.”

Martin Fowler

Public

Hva er egentlig “skyen”?

Compute Engine

App Engine

Kubernetes Engine

Cloud Functions

Container-Optimized OS

Public

Har vi en delt forståelse for hva en “nettsky” er?

Image: Sticker Mule

I skyen så...

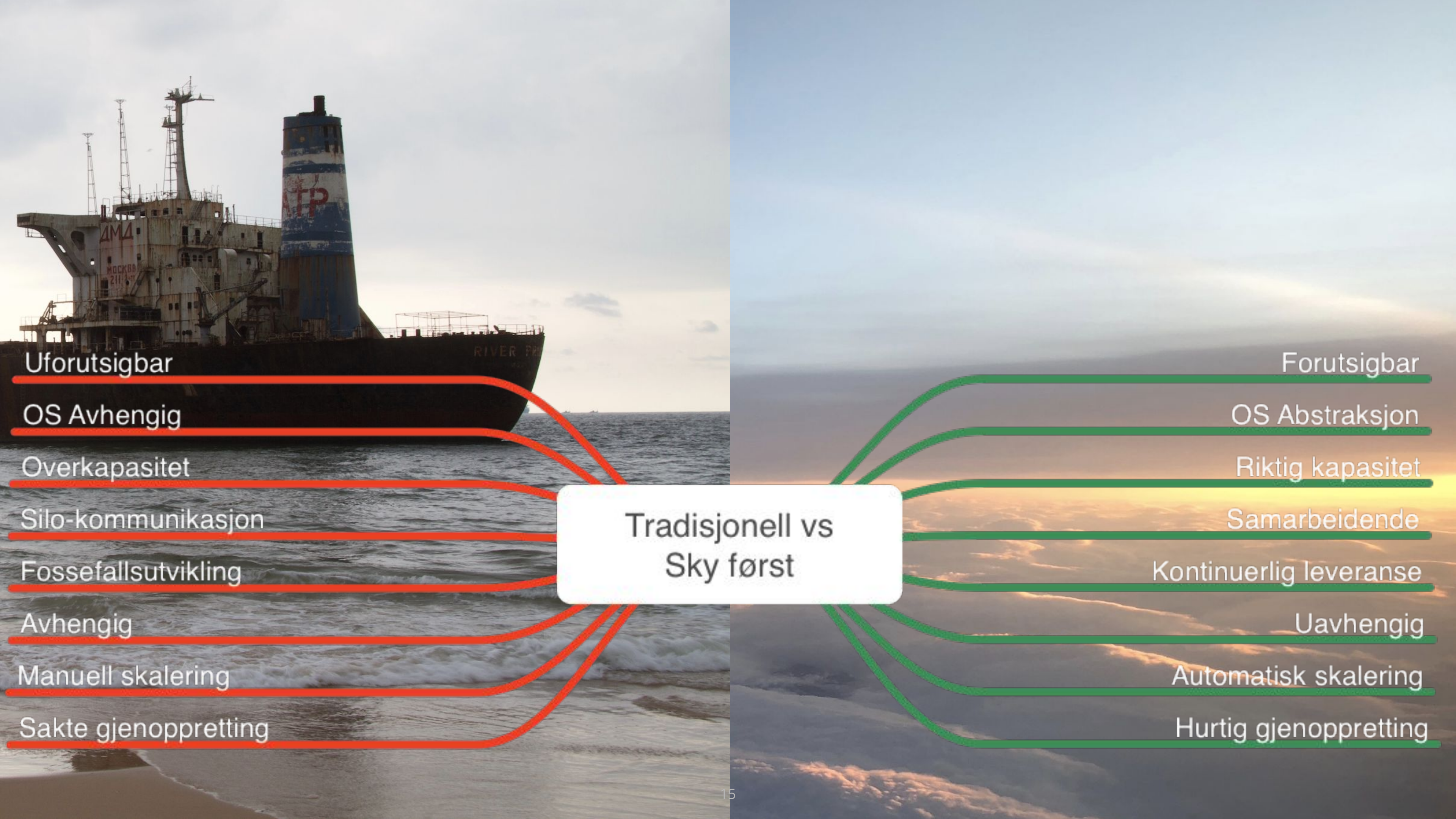
- er vi ikke i komplett kontroll over miljøet vi kjører i
- har ting en tendens til å endre seg når vi ikke vil at de skal det
- ting oppfører seg annerledes enn “det vi var vant til”

Image: Sticker Mule

En tjeneste for å **sette opp og fjerne, elastisk og på forespørsel:**

- datakraft
- lagring
- nettverk

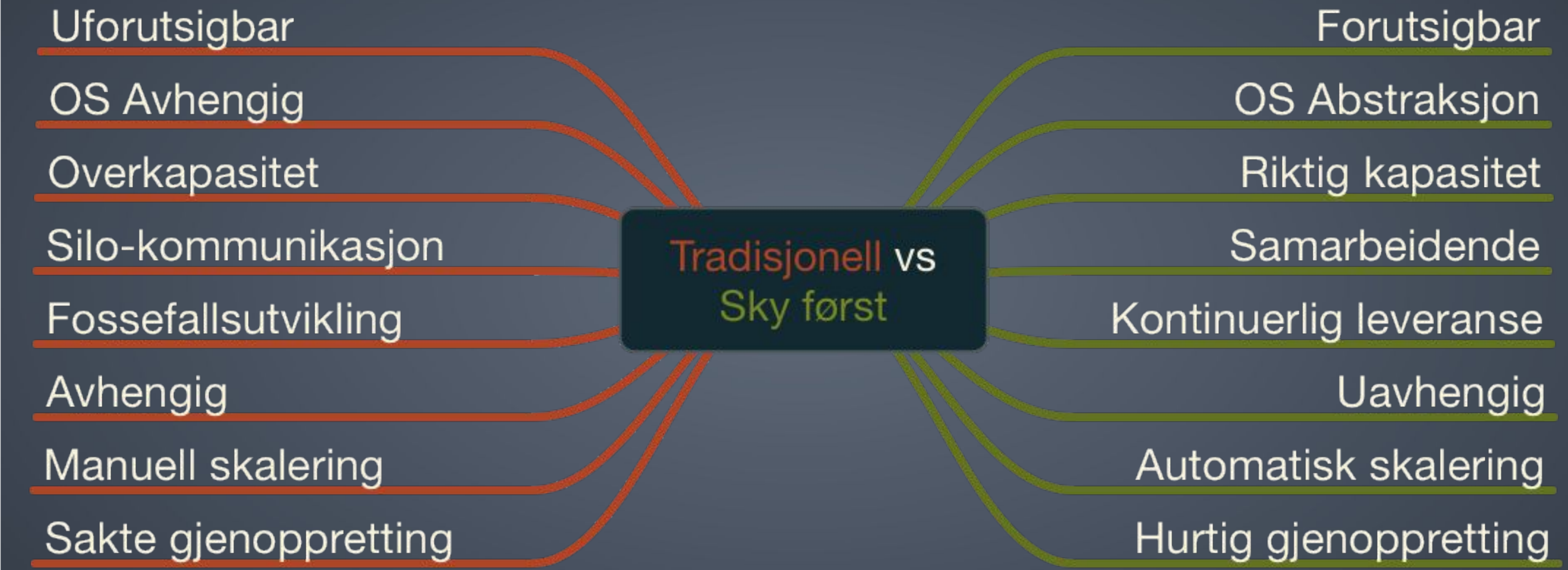
ДМД
МОСКВА
211
ATP
RIVER PR
DANGER



Tradisjonell vs Sky først

- Uforutsigbar
- OS Avhengig
- Overkapasitet
- Silo-kommunikasjon
- Fossefallsutvikling
- Avhengig
- Manuell skalering
- Sakte gjenoppretting

- Forutsigbar
- OS Abstraksjon
- Riktig kapasitet
- Samarbeidende
- Kontinuerlig leveranse
- Uavhengig
- Automatisk skalering
- Hurtig gjenoppretting



Tradisjonell vs Sky først

Tradisjonell	Sky først
Uforutsigbar	Forutsigbar
OS Avhengig	OS Abstraksjon
Overkapasitet	Riktig kapasitet
Silo-kommunikasjon	Samarbeidende
Fossefallsutvikling	Kontinuerlig leveranse
Avhengig	Uavhengig
Manuell skalering	Automatisk skalering
Sakte gjenoppretting	Hurtig gjenoppretting

Forutsigbar

Følger et rammeverk eller en kontrakt (f.eks. 12factor app)

→ Maksimer robusthet gjennom forutsigbar oppførsel

Svært høy grad av automasjon

Uforutsigbar

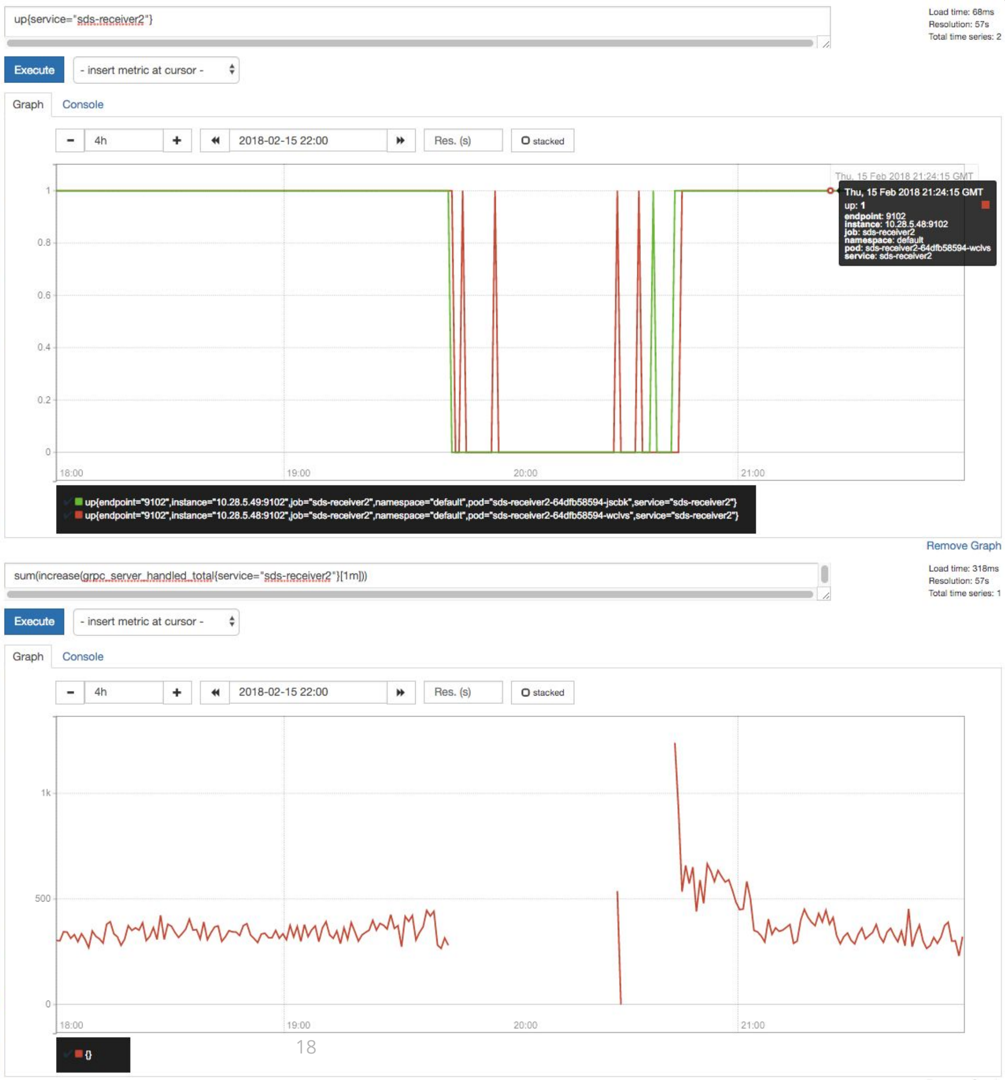
Øker byggetiden

Skalerer gradvis (manuelt)

Manuelle prosesser

Store releases

Observerbarhet



```
up{service="sds-receiver2"}
```

Load time: 68ms
Resolution: 57s
Total time series: 2

Execute | - insert metric at cursor -

Graph | Console

− 4h + | ⏪ 2018-02-15 22:00 ⏩ | Res. (s) | ○ stacked

Thu, 15 Feb 2018 21:24:15 GMT

Thu, 15 Feb 2018 21:24:15 GMT
up: 1 🟥
endpoint: 9102
instance: 10.28.5.48:9102
job: sds-receiver2
namespace: default
pod: sds-receiver2-64dfb58594-wclvs
service: sds-receiver2

```
🟩 up{endpoint="9102",instance="10.28.5.49:9102",job="sds-receiver2",namespace="default",pod="sds-receiver2-64dfb58594-jscbk",service="sds-receiver2"}
🟥 up{endpoint="9102",instance="10.28.5.48:9102",job="sds-receiver2",namespace="default",pod="sds-receiver2-64dfb58594-wclvs",service="sds-receiver2"}
```

Remove Graph

```
sum(increase(grpc_server_handled_total{service="sds-receiver2"}[1m]))
```

Load time: 318ms
Resolution: 57s
Total time series: 1

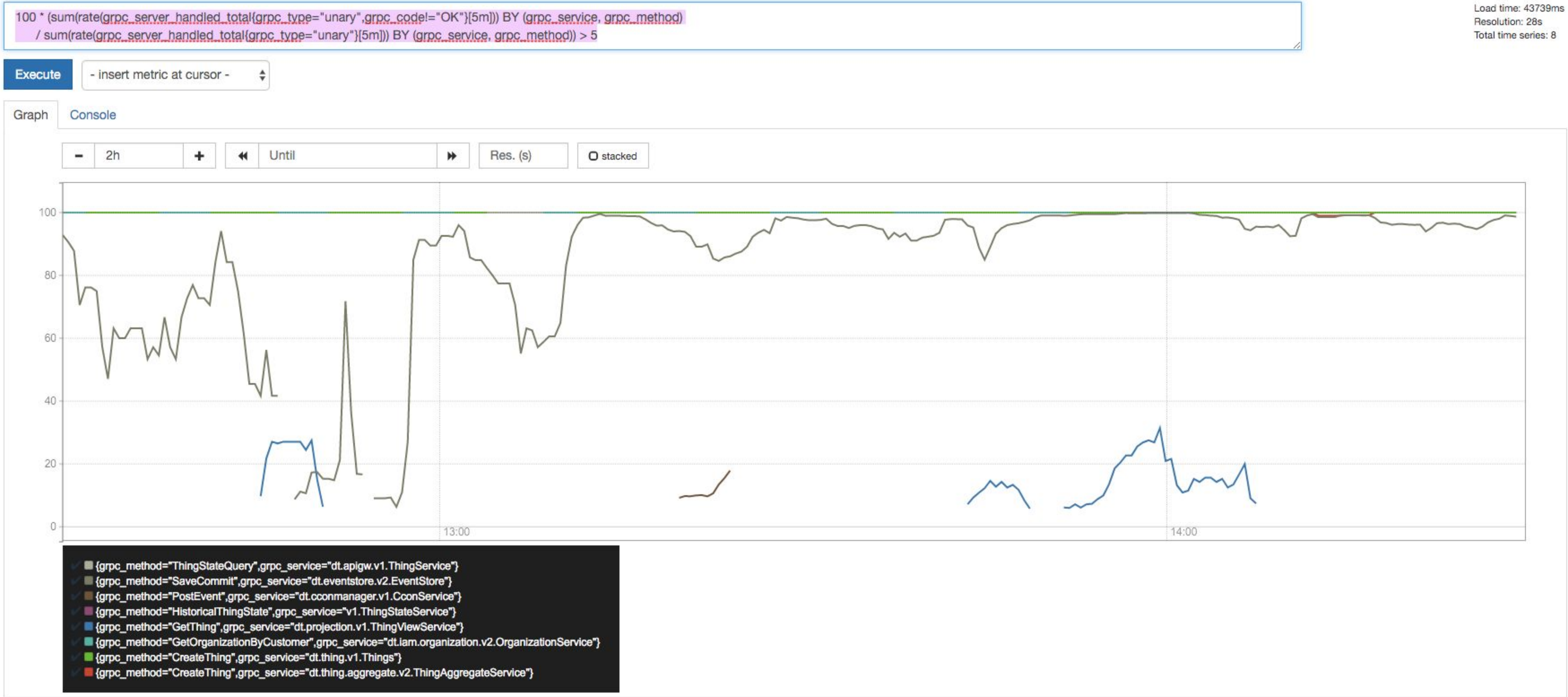
Execute | - insert metric at cursor -

Graph | Console

− 4h + | ⏪ 2018-02-15 22:00 ⏩ | Res. (s) | ○ stacked

🟥 {}

Observerbarhet

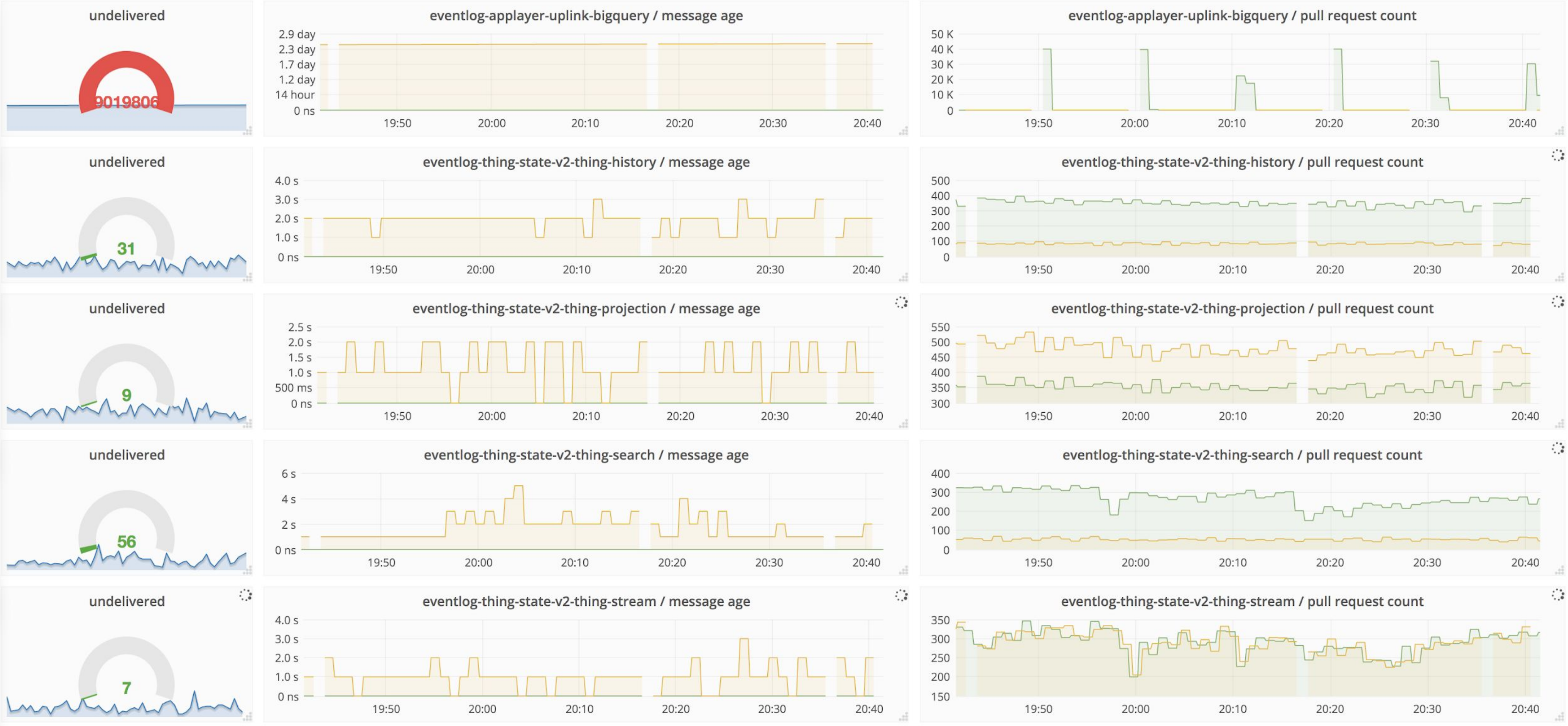


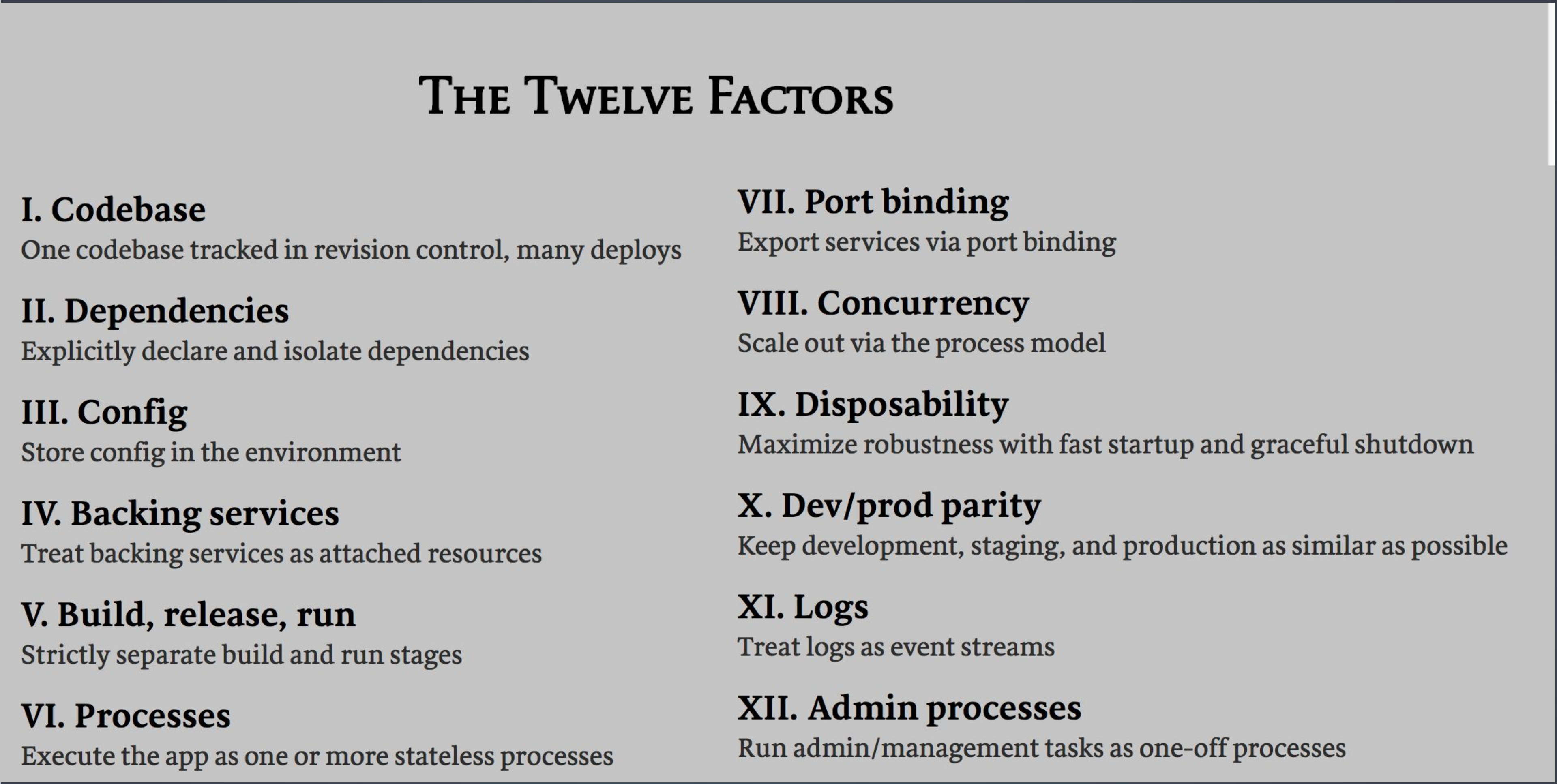
```
100 * (sum(rate(grpc_server_handled_total{grpc_type="unary",grpc_code!="OK"}[5m])) BY (grpc_service, grpc_method)
   / sum(rate(grpc_server_handled_total{grpc_type="unary"}[5m])) BY (grpc_service, grpc_method)) > 5
```

Load time: 43739ms
Resolution: 28s
Total time series: 8

- ✓ ■ {grpc_method="ThingStateQuery",grpc_service="dt.apigw.v1.ThingService"}
- ✓ ■ {grpc_method="SaveCommit",grpc_service="dt.eventstore.v2.EventStore"}
- ✓ ■ {grpc_method="PostEvent",grpc_service="dt.cconmanager.v1.CconService"}
- ✓ ■ {grpc_method="HistoricalThingState",grpc_service="v1.ThingStateService"}
- ✓ ■ {grpc_method="GetThing",grpc_service="dt.projection.v1.ThingViewService"}
- ✓ ■ {grpc_method="GetOrganizationByCustomer",grpc_service="dt.iam.organization.v2.OrganizationService"}
- ✓ ■ {grpc_method="CreateThing",grpc_service="dt.thing.v1.Things"}
- ✓ ■ {grpc_method="CreateThing",grpc_service="dt.thing.aggregate.v2.ThingAggregateService"}

Observerbarhet


undelivered
9019806
eventlog-applayer-uplink-bigquery / message age
2.9 day
2.3 day
1.7 day
1.2 day
14 hour
0 ns
19:50
20:00
20:10
20:20
20:30
20:40
eventlog-applayer-uplink-bigquery / pull request count
50 K
40 K
30 K
20 K
10 K
0
undelivered
31
eventlog-thing-state-v2-thing-history / message age
4.0 s
3.0 s
2.0 s
1.0 s
0 ns
eventlog-thing-state-v2-thing-history / pull request count
500
400
300
200
100
0
undelivered
9
eventlog-thing-state-v2-thing-projection / message age
2.5 s
2.0 s
1.5 s
1.0 s
500 ms
0 ns
eventlog-thing-state-v2-thing-projection / pull request count
550
500
450
400
350
300
undelivered
56
eventlog-thing-state-v2-thing-search / message age
6 s
4 s
2 s
0 ns
eventlog-thing-state-v2-thing-search / pull request count
400
300
200
100
0
undelivered
7
eventlog-thing-state-v2-thing-stream / message age
4.0 s
3.0 s
2.0 s
1.0 s
0 ns
eventlog-thing-state-v2-thing-stream / pull request count
350
300
250
200
150



The Twelve Factors

I. Codebase

One codebase tracked in revision control, many deploys

II. Dependencies

Explicitly declare and isolate dependencies

III. Config

Store config in the environment

IV. Backing services

Treat backing services as attached resources

V. Build, release, run

Strictly separate build and run stages

VI. Processes

Execute the app as one or more stateless processes

VII. Port binding

Export services via port binding

VIII. Concurrency

Scale out via the process model

IX. Disposability

Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity

Keep development, staging, and production as similar as possible

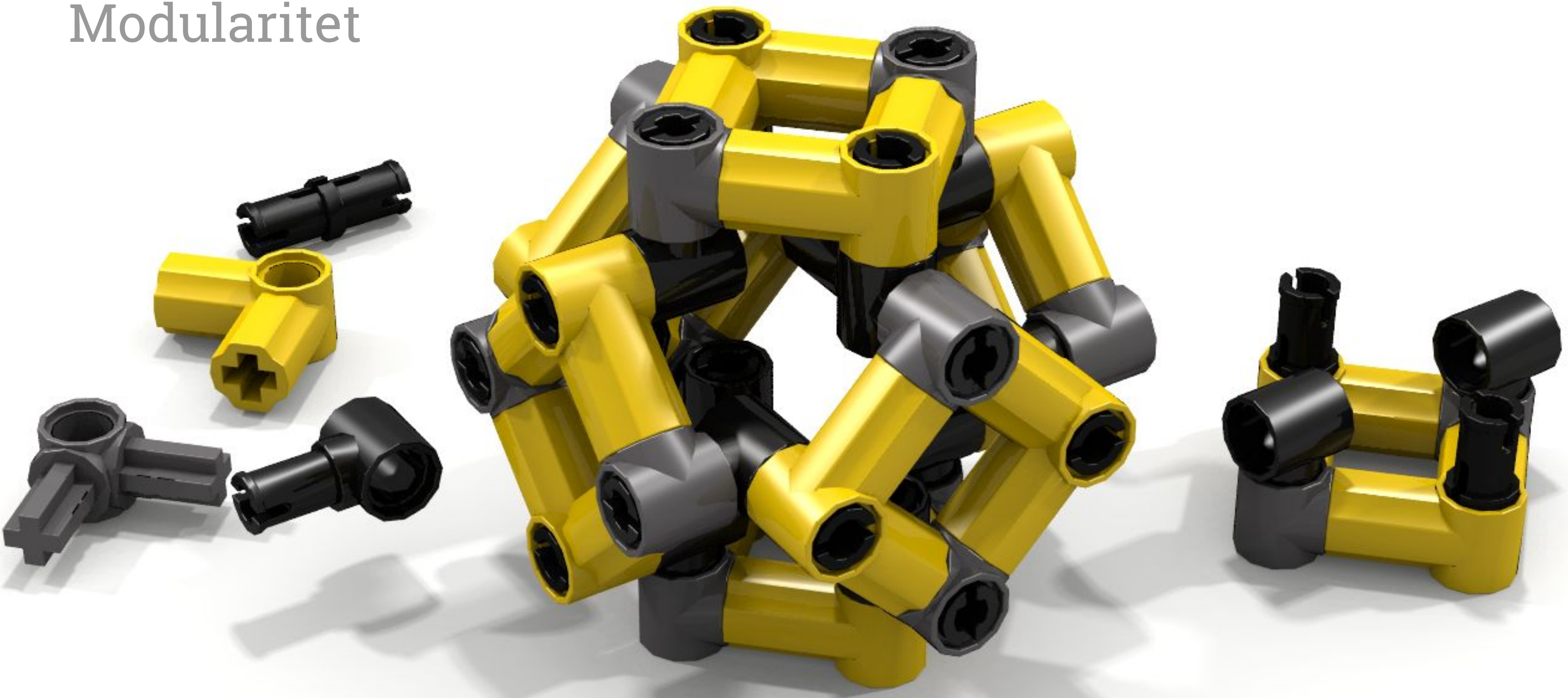
XI. Logs

Treat logs as event streams

XII. Admin processes

Run admin/management tasks as one-off processes

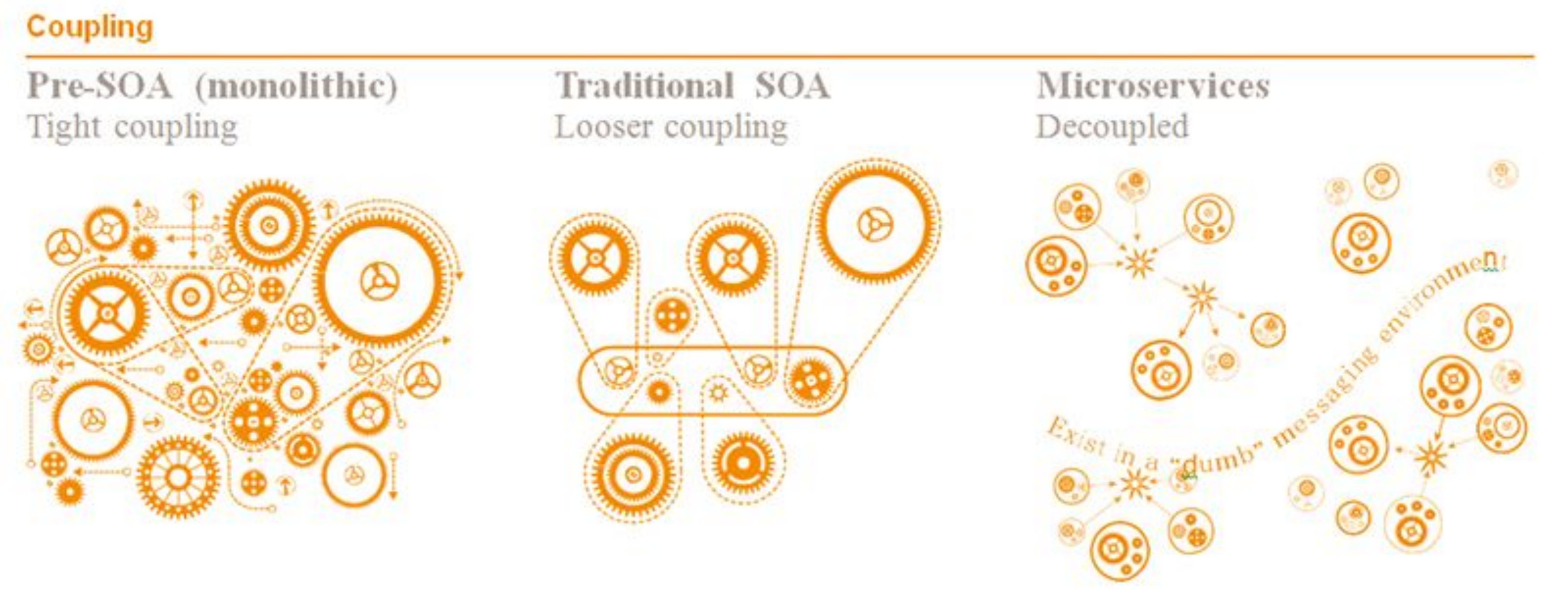
Modularitet



https://flic.kr/p/Y8hVDJ CC BY-SA 2.0

Avhengig

Uavhengig



http://usblogs.pwc.com/emerging-technology/agile-coding-in-enterprise-it-code-small-and-local/

Public

Deployerbar

Photo: WFP/Peter Smerdon

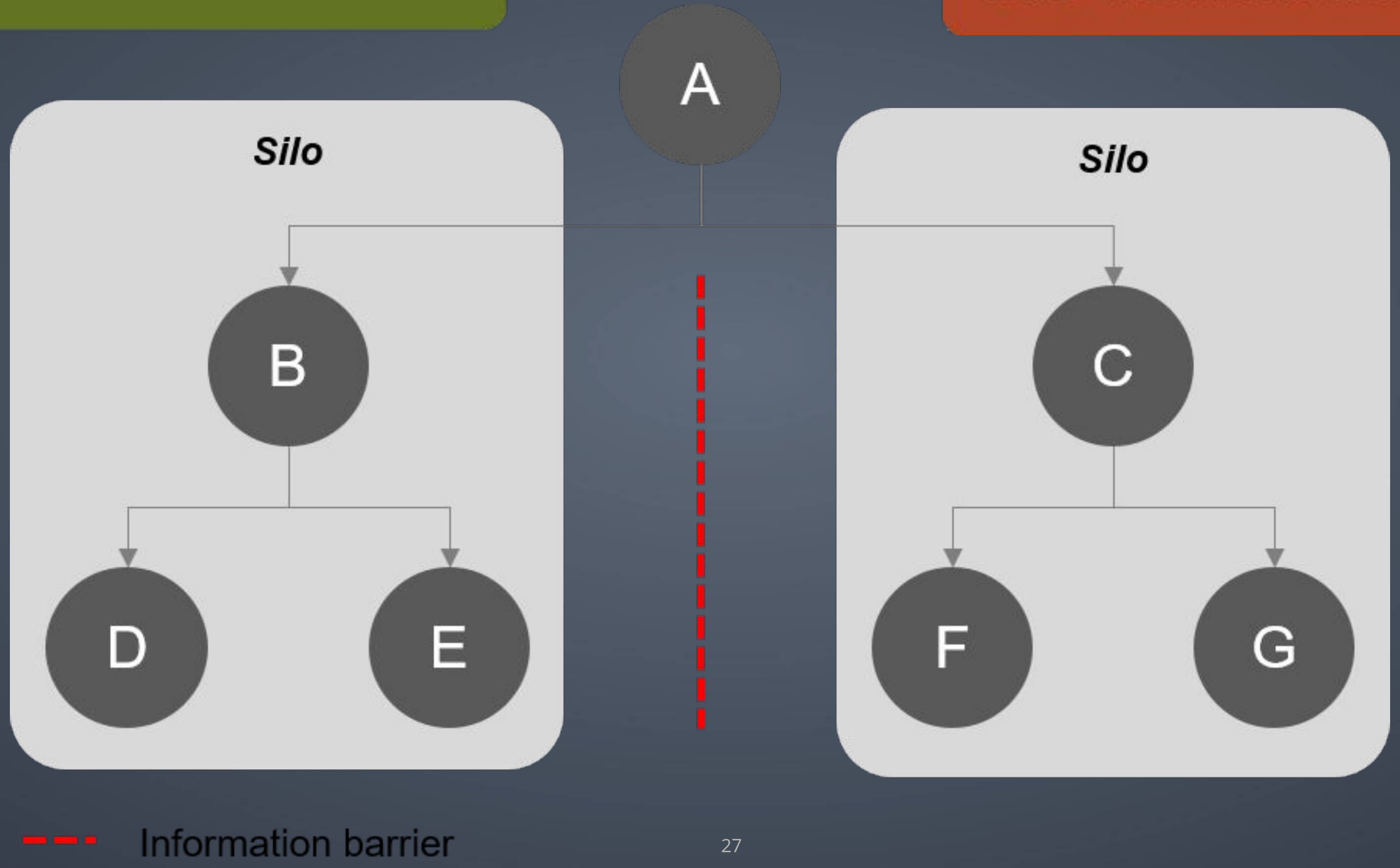
Kastbart / Byttbart

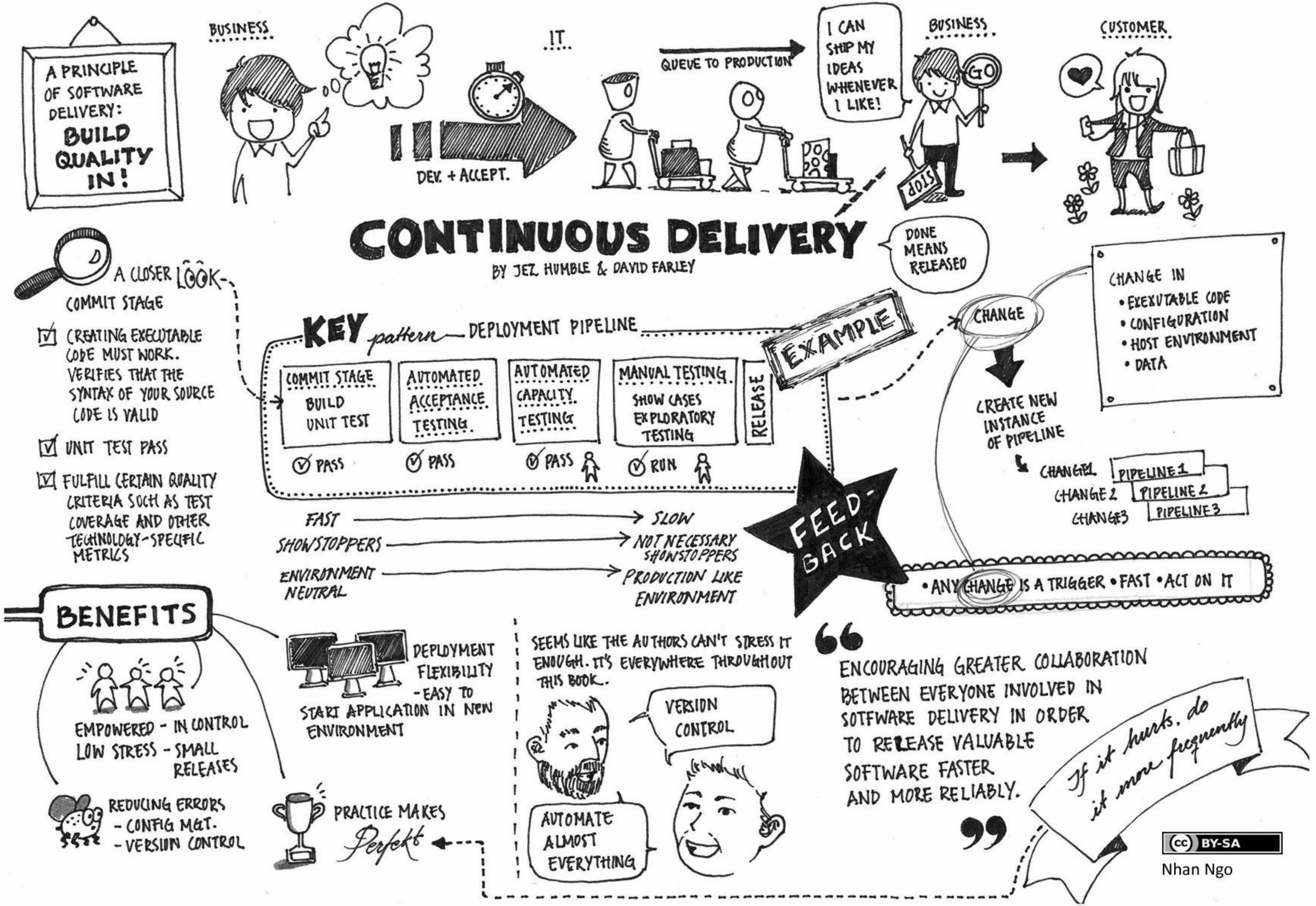
By Carlos Delgado, CC BY-SA 3.0,
https://commons.wikimedia.org/w/index.php?curid=19747330

Public

Samarbeidende

Silo-kommunikasjon

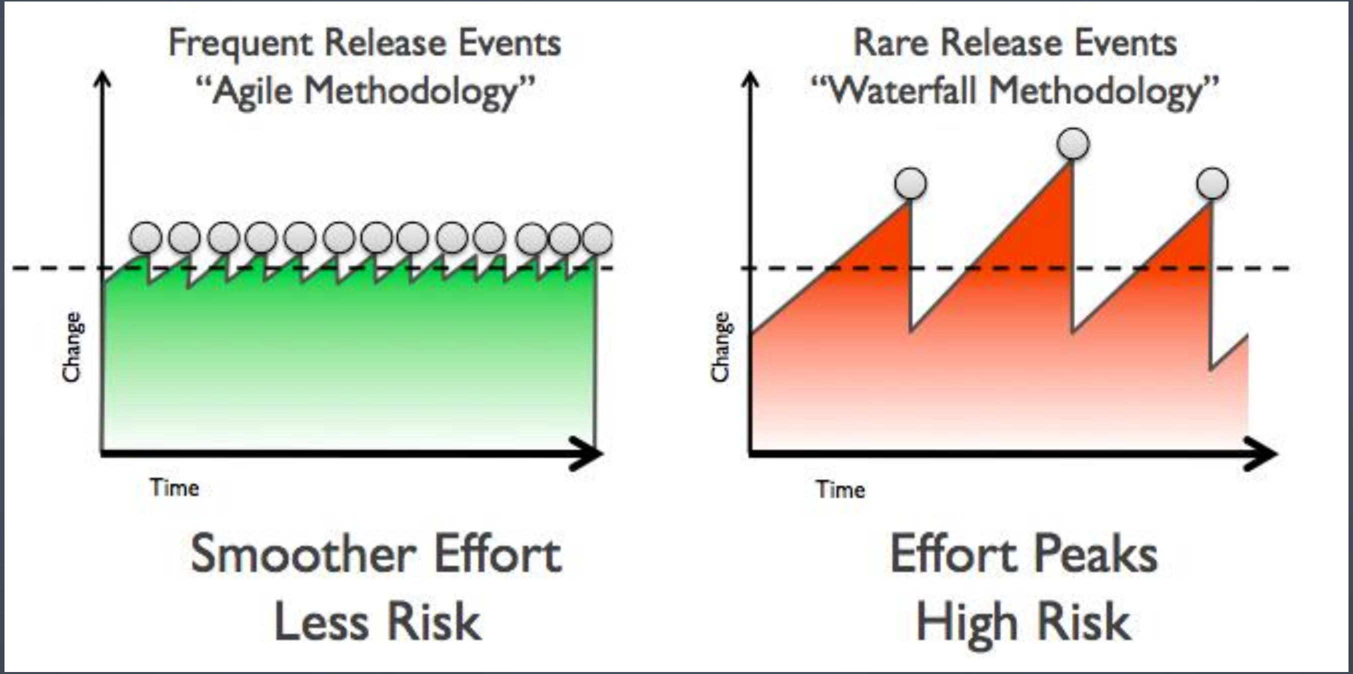



A PRINCIPLE OF SOFTWARE DELIVERY: BUILD QUALITY IN!
BUSINESS
IT
DEV. + ACCEPT.
QUEUE TO PRODUCTION
I CAN SHIP MY IDEAS WHENEVER I LIKE!
BUSINESS
GO
STOP
CUSTOMER
CONTINUOUS DELIVERY
BY JEZ HUMBLE & DAVID FARLEY
DONE MEANS RELEASED
A CLOSER LOOK
COMMIT STAGE
☑ CREATING EXECUTABLE CODE MUST WORK. VERIFIES THAT THE SYNTAX OF YOUR SOURCE CODE IS VALID
☑ UNIT TEST PASS
☑ FULFILL CERTAIN QUALITY CRITERIA SUCH AS TEST COVERAGE AND OTHER TECHNOLOGY-SPECIFIC METRICS
KEY pattern — DEPLOYMENT PIPELINE
EXAMPLE
COMMIT STAGE
BUILD
UNIT TEST
✓ PASS
AUTOMATED ACCEPTANCE TESTING
✓ PASS
AUTOMATED CAPACITY TESTING
✓ PASS
MANUAL TESTING
SHOW CASES
EXPLORATORY TESTING
✓ RUN
RELEASE
FAST → SLOW
SHOWSTOPPERS → NOT NECESSARY SHOWSTOPPERS
ENVIRONMENT NEUTRAL → PRODUCTION LIKE ENVIRONMENT
FEED-BACK
CHANGE
CHANGE IN
• EXEXUTABLE CODE
• CONFIGURATION
• HOST ENVIRONMENT
• DATA
CREATE NEW INSTANCE OF PIPELINE
CHANGE1 PIPELINE 1
CHANGE2 PIPELINE 2
CHANGE3 PIPELINE 3
• ANY CHANGE IS A TRIGGER • FAST • ACT ON IT
BENEFITS
EMPOWERED - IN CONTROL
LOW STRESS - SMALL RELEASES
REDUCING ERRORS
- CONFIG MGT.
- VERSION CONTROL
DEPLOYMENT FLEXIBILITY
- EASY TO START APPLICATION IN NEW ENVIRONMENT
PRACTICE MAKES Perfekt
SEEMS LIKE THE AUTHORS CAN'T STRESS IT ENOUGH. IT'S EVERYWHERE THROUGHOUT THIS BOOK.
VERSION CONTROL
AUTOMATE ALMOST EVERYTHING
“ ENCOURAGING GREATER COLLABORATION BETWEEN EVERYONE INVOLVED IN SOTFWARE DELIVERY IN ORDER TO RELEASE VALUABLE SOFTWARE FASTER AND MORE RELIABLY. ”
If it hurts, do it more frequently

Nhan Ngo

Kontinuerlig leveranse

Fossefallsutvikling



https://en.wikipedia.org/wiki/File:Agile-vs-iterative-flow.jpg

Public

Forretningsdrivere for en Sky-Først Arkitektur

Forretningssmidighet

Forretningssmidighet

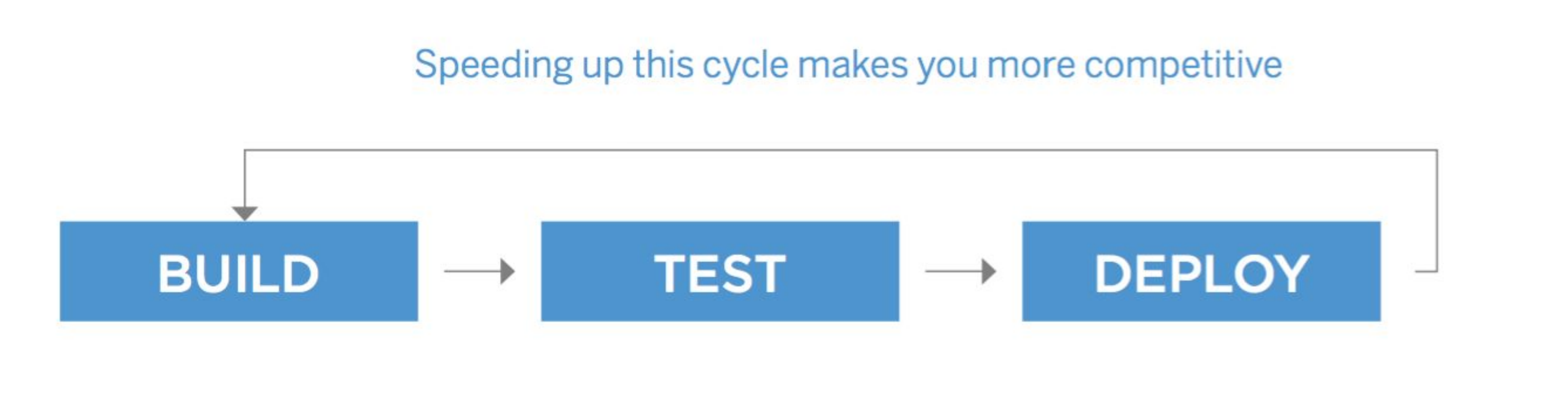
Optimere “fra Konsept til Kontanter syklusen”

- Respondere til endringer i forr. område
- Være driver for endringer i forr. området

Forretningssmidighet

Optimere “fra Konsept til Kontanter syklusen”

- Respondere til endringer i forr. område
- Være driver for endringer i forr. området



Speeding up this cycle makes you more competitive

Forretningssmidighet

En typisk metode for å unngå feil:

- legg til mer testing
- mer release engineering
- flere kode gjennomganger
- flere sjekklister
- flere godkjennelser

MY MOTTO IS "MOVE FAST AND BREAK THINGS."
JOBS I'VE BEEN FIRED FROM
FEDEX DRIVER
CRANE OPERATOR
SURGEON
AIR TRAFFIC CONTROLLER
PHARMACIST
MUSEUM CURATOR
WAITER
DOG WALKER
OIL TANKER CAPTAIN
VIOLINIST
MARS ROVER DRIVER
MASSAGE THERAPIST

https://xkcd.com/1428/

Forretningssmidighet

Omfavn forsømmelser og mistak

=> Lær av de

=> lag feiltolerant software

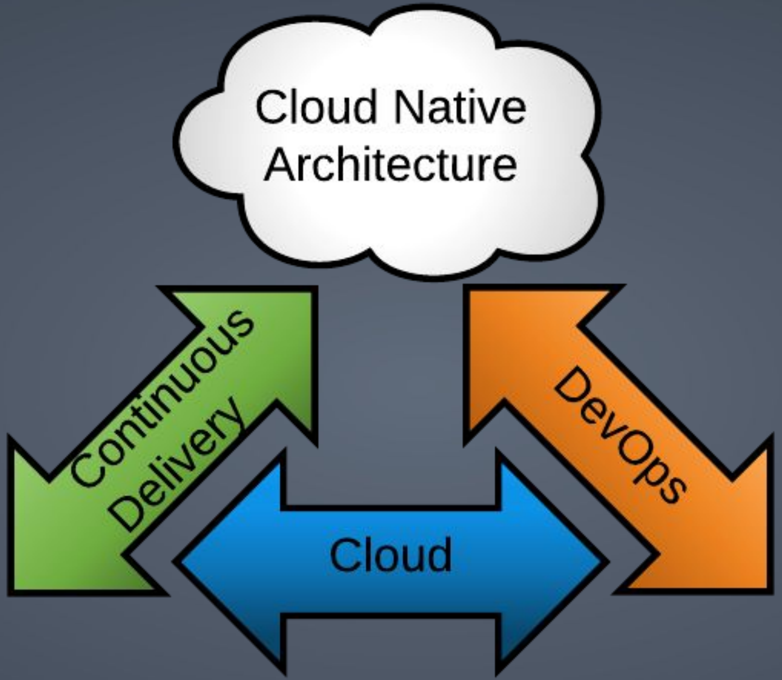
Målet er å løpe raskt uten å ødelegge ting!

MY MOTTO IS "MOVE FAST AND BREAK THINGS."
JOBS I'VE BEEN FIRED FROM
FEDEX DRIVER
CRANE OPERATOR
SURGEON
AIR TRAFFIC CONTROLLER
PHARMACIST
MUSEUM CURATOR
WAITER
DOG WALKER
OIL TANKER CAPTAIN
VIOLINIST
MARS ROVER DRIVER
MASSAGE THERAPIST

https://xkcd.com/1428/

DevOps+CD+Skyplattform => CNA

DevOps	● Modulært ● Observerbart
Kontinuerlige Leveranser (CD)	● Deployerbart ● Testbart
Skyplattform	● Kastbart ● Utbyttbart



Cloud Native Architecture
Continuous Delivery
DevOps
Cloud

POWERED BY

DISRUPTIVE

TECHNOLOGIES

Real time awareness
on a global scale