Dynamiske arkitekturer

IMT3003 kyrre.Begnum@ntnu.no

Paradigmeskiftet innen tjenestearkitekturer

- Tidligere ble IT løsninger sett på som isolerte systemer (siloer)
- Store organisasjoner ville få mange slike siloer
- Synergier mellom løsningene var svært vanskelig, ofte umulig
- Alternativet ble en ny bevegelse, hvor tjenester i større grad snakker med hverandre

SOA - Service Oriented Architecture

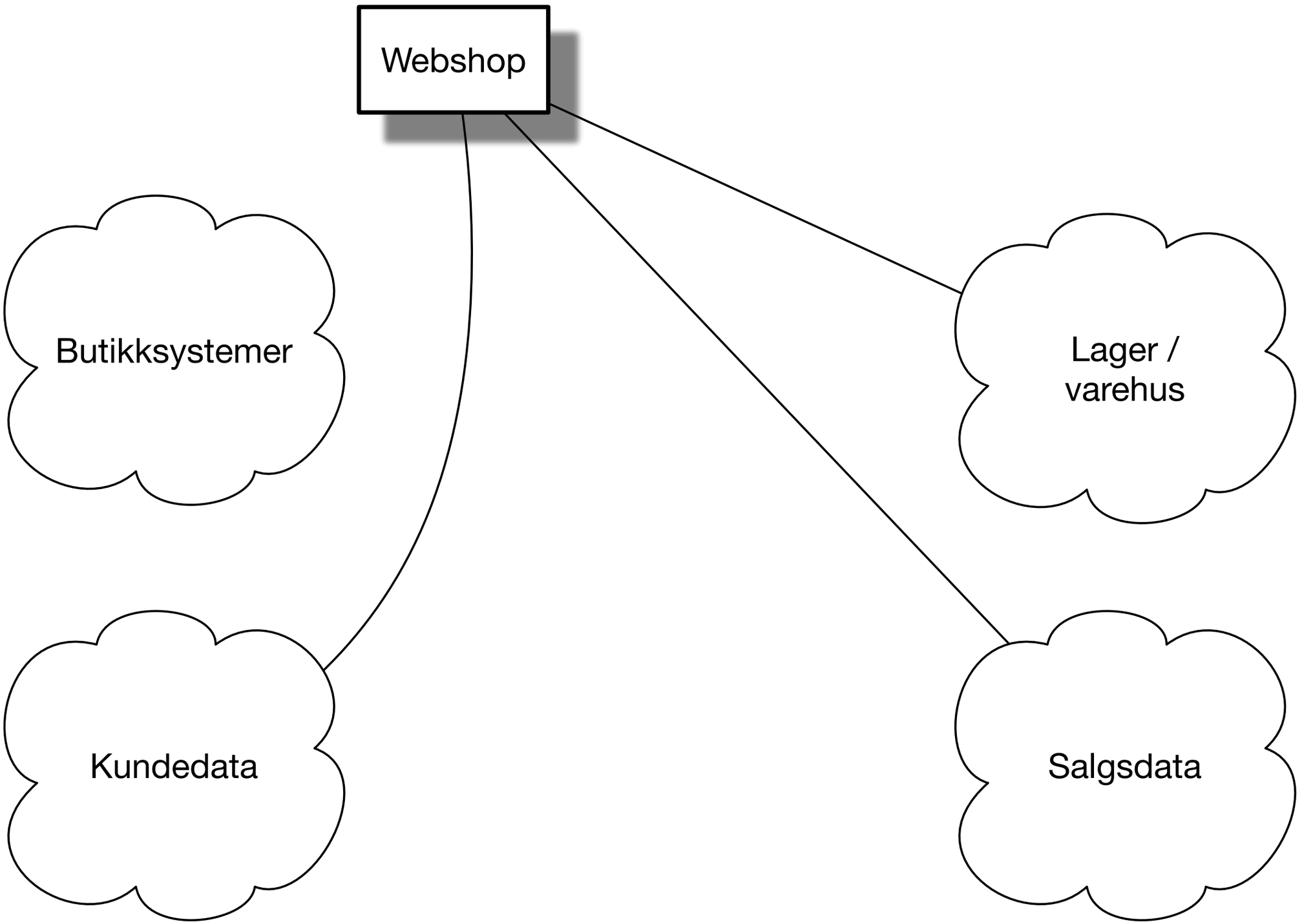
- Mindre tjenester som presenterer en API til hverandre
- Systemer drar nytte av tjenestene etter behov
- Mer komplisert, men samtidig evolusjonært
  - Tjenester kan videreutvikles parallelt
  - Nye tjenester kan tilføyes
  - Man kan "skjule" gamle systemer ved å pakke dem inn en proxy-tjeneste

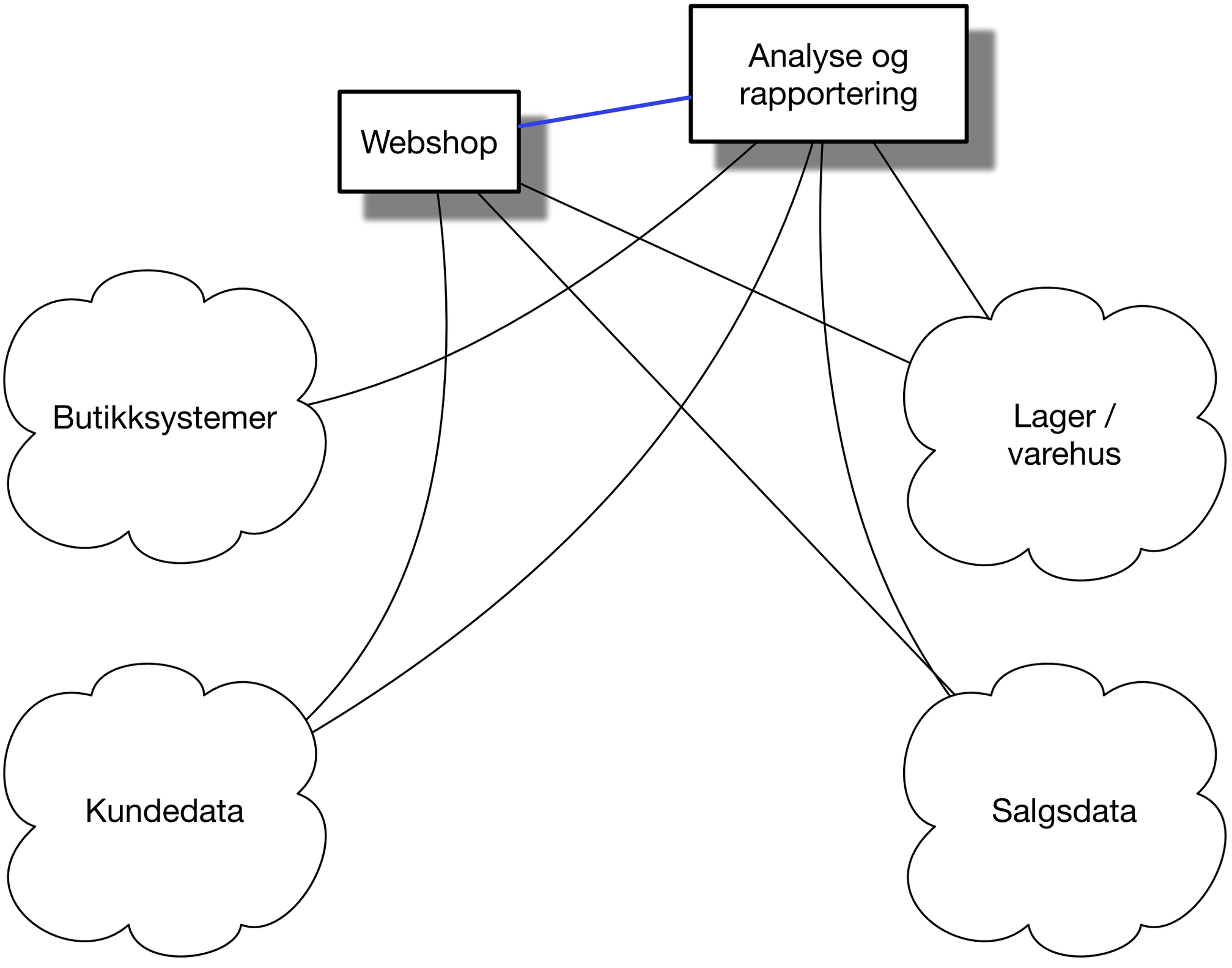
Butikksystemer

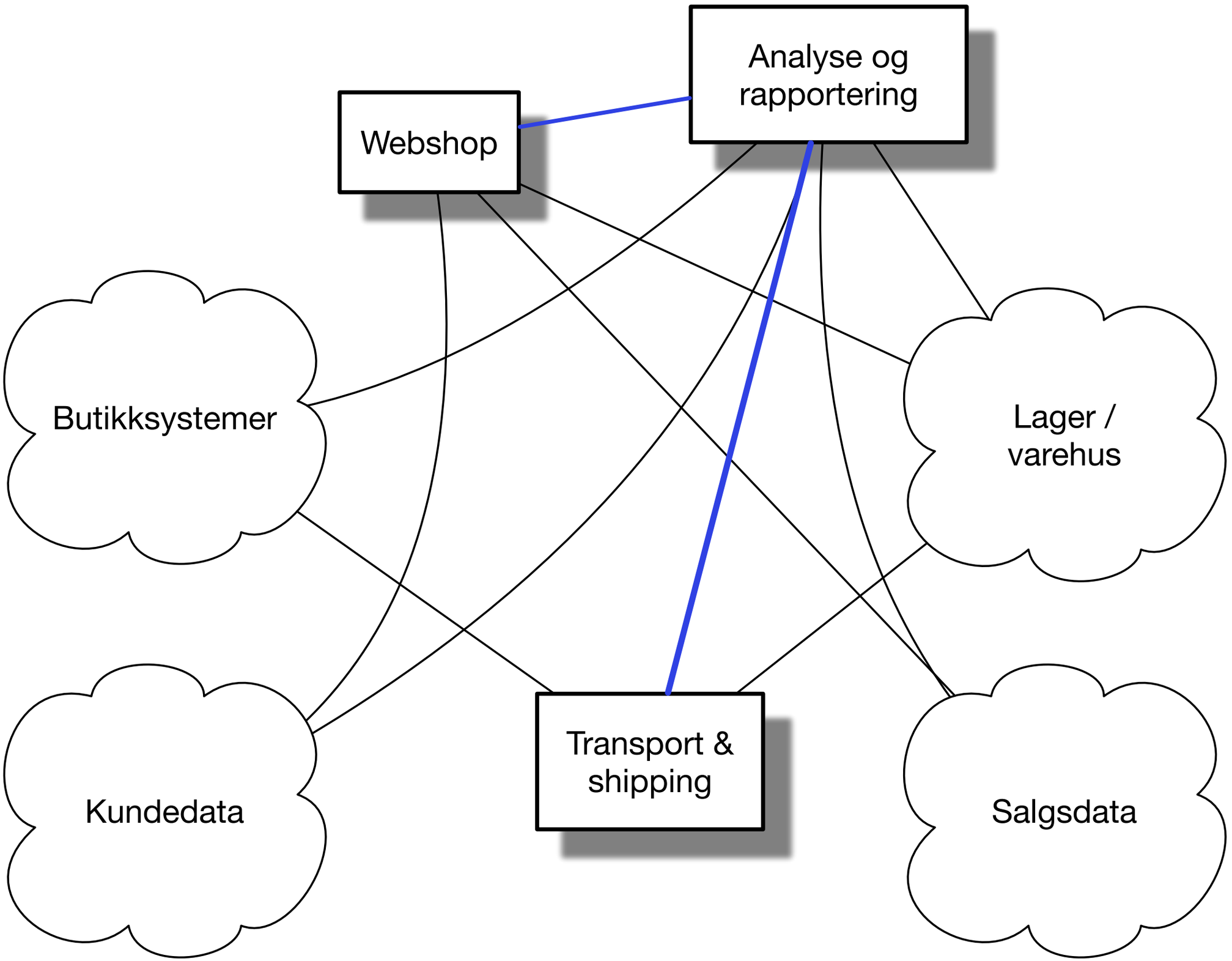
Lager /
varehus

Kundedata

Salgsdata


Webshop
Butikksystemer
Lager /
varehus
Kundedata
Salgsdata


Analyse og rapportering
Webshop
Butikksystemer
Lager / varehus
Kundedata
Salgsdata


Analyse og rapportering
Webshop
Butikksystemer
Lager / varehus
Kundedata
Transport & shipping
Salgsdata

Driftsmessige observasjoner

- Alle tjenestene skal være ganske "enkle"
  - Microservices
- Kommunikasjon er vanligvis på HTTP / REST med JSON, før var det XML
- Det kan være en viss robusthet
  - Noen tjenester kan fungere mens andre er nede
- Det store spørsmålet blir hvordan de finner hverandre

Service Discovery (Tjenesteoppslag)

- Hvordan kan tjenester som har blitt utviklet parallelt finne hverandre?
  - Bruk av faste IP adresser og portnummer nytter ikke i et skybasert miljø
  - Løsningen blir å ha et "oppslagsverk" hvor tjenester kan få IP adressen og portnummer på tjenesten de leter etter

Service Discovery and Configuration Made Easy

IMT3003 kyrre.Begnum@ntnu.no

Consul

- Et system utviklet av HashiCorp for å levere service discovery i store, globale arkitekturer
- Klyngebasert og distribuert for minimal nedetid
- Kan dynamisk legge til nye tjenester og servere
- Key / Value lagring av tekniske data
- Innebygget overvåkning
- Bruker DNS som oppslags-teknologi

Consul klyngen

- Hoveddelen av consul er en klynge med maskiner som sammen holder orden på:
  - Tjenester
  - Andre servere
  - Hvor de er i verden (datasentre)
- Ideen er at man har en slik klynge i hvert datasenter, men de vil selv speile og holde informasjon oppdatert
- Navnekonvensjonene er allerede satt av consul, men kan legges til eksisterende domener

DNS oppslag

- Consul vet om noder (servere) og tjenester
- I utgangspunktet bruker den ".consul" som domenenavn
- En tjenste vil være: tjenestenavn.service.consul
  eller: tjenestenavn.service.datasenter.consul
  f.eks: `mariadb.service.consul`
- Noder er lignende:
  navn.node.consul
  eller
  navn.node.datacenter.consul
  f.eks `db1.node.dc1.consul`
- Dersom consul får en forespørsel den ikke vet, sender den det videre til den ekte DNS serveren

Fordelen med DNS

- "Alle" systemer kan snakke DNS
- Brukere av service discovery trenger ikke kjøre consul eller egne agenter, bare pek mot en ny DNS server
  - F.eks en C# app på en Win 2008 Server kan bruke DNS for å finne databasen, uten at den vet at det er consul som driver lastbalansering
- DNS er en enkel protokoll som allerede er åpnet for i de fleste netverk
  - Man kan ta det i bruk uten å endre på mye

Tjenester i consul

- Hva som helst kan defineres som en tjeneste, f.eks web, databaser og API'er
  - Alle servere som er del av en tjeneste kjører en consul agent som sender info til consul klyngen
  - Klyngen vil automatisk oppdatere egen info og bygge DNS basert på det den vet

Nytten ved å inkludere overvåkning

- Consul vil gi en liste av tilgjengelige servere som DNS svar
- Dersom en server ikke svarer, vil den heller ikke dukke opp i svaret
- Klienter vil dermed kun ta kontakt med tilgjengelige servere
- Lastbalansering er ivaretatt, men trafikk går ikke gjennom en lastbalanserer

Consul Installasjon

IMT3003 kyrre.Begnum@ntnu.no

Mise en place

- Tre nye virtuelle maskiner:
  - consul1
  - consul2
  - consul3
  - client (for testing, ikke nødvendig om du kan teste fra andre steder)
- Eksisterende maskiner (en eller annen tjeneste man har)
  - db1,db2,db3
- Ubuntu 16.04 på alle
- IP til DNS server (prøv cat /etc/resolv.conf)

På consul 1, 2 og 3

- Først må vi installere consul
- Kjør følgende for å installere consul:

```
apt-get install -y unzip
wget https://releases.hashicorp.com/consul/1.0.7/consul_1.0.7_linux_amd64.zip
unzip consul_1.0.7_linux_amd64.zip
mv consul /usr/local/bin/
mkdir /opt/consul
```

Start i bootstrap modus

- Akkurat som en databaseklynge, må man starte en av nodene som en dedikert mester
- Start en screen sesjon på consul1

  ```
  screen -S consul
  ```

  og kjør følgende:

  ```
  consul agent -server -bootstrap -data-dir /opt/consul
  ```
- Denne kommandoen vil "henge", men nå kjører consul

Start resten av consul klyngen

- På consul2 og consul3 oppretter man også en screen sesjon, men kjører følgende kommando (på en linje):

```
consul agent -server -data-dir /opt/consul
-client 0.0.0.0 -dns-port 53 -recursor DNS-IP
```

Legg til alle noder og start consul på vanlig vis

- På consul1 (men uten å avbryte kommandoen i screen sesjonen):

```
consul join CONSUL2-IP CONSUL3-IP
```

- Deretter, gå inn i screen sesjonen og stop consul med Ctrl+c
- Nå kan du starte consul på ordentlig:

```
consul agent -ui -server -data-dir /opt/consul
-client 0.0.0.0 -dns-port 53 -recursor DNS-IP
```

Consul Dashboard

- Consul dashboard er en tjeneste som kan leveres av en consul agent.
- Når man legger til "-ui", så får man dashboard som lytter på port 8500
  - Dette ble lagt til på consul1
  - Man må muligens sjekke security groups og legge til floating-ip

Case: MariaDB klynge i consul

MariaDB med service discovery

- Når consul klyngen er oppe, kan man endelig begynne å høste fruktene fra arbeidet
- Neste steg er å legge til alle serverne i MariaDB klyngen som en tjeneste som consul holder orden på
- Når det fungerer, vil alle klienter kunne gjøre DNS spørringer etter:
  `mariadb.service.consul`
  Da får de IP addressene til alle serverne i klyngen som er oppe

På db1, db2 og db3

- Installer consul akkurat som før
- I tillegg lag en ekstra mappe:
  `mkdir -p /etc/consul/services`
- Her vil vi lage konfigurasjonsfiler for alle tjenestene som vi skal fortelle consul om

På db1, db2 og db3

- Lag en fil /etc/consul/services/mycluster.conf med følgende innhold:

```
{
  "service": {
        "name": "MariaDB",
        "port": 3306,
        "tags": ["mariadb", "mycluster"],
        "check": {
            "script": "netstat -anltp | grep 3306",
            "interval": "10s"
        }
    }
}
```

Start consul på db1, db2 og db3

- Nå kan vi starte consul på alle tre database serverne
- Start dem på samme måte som før, nemlig i en screen sesjon:

```
consul agent -data-dir /opt/consul -join
CONSUL1-IP --config-dir /etc/consul/
services
```

Sjekk og test

- Når consul agenten kjører på databaseklyngen, burde den også være synlig i dashboard
- Man kan også sjekke DNS funksjonaliteten slik:

```
dig @CONSUL1-IP mariadb.service.consul ANY

dig @CONSUL1-IP db1.node.consul
```

Integrere Docker Swarm med Consul

IMT3003 kyrre.Begnum@ntnu.no

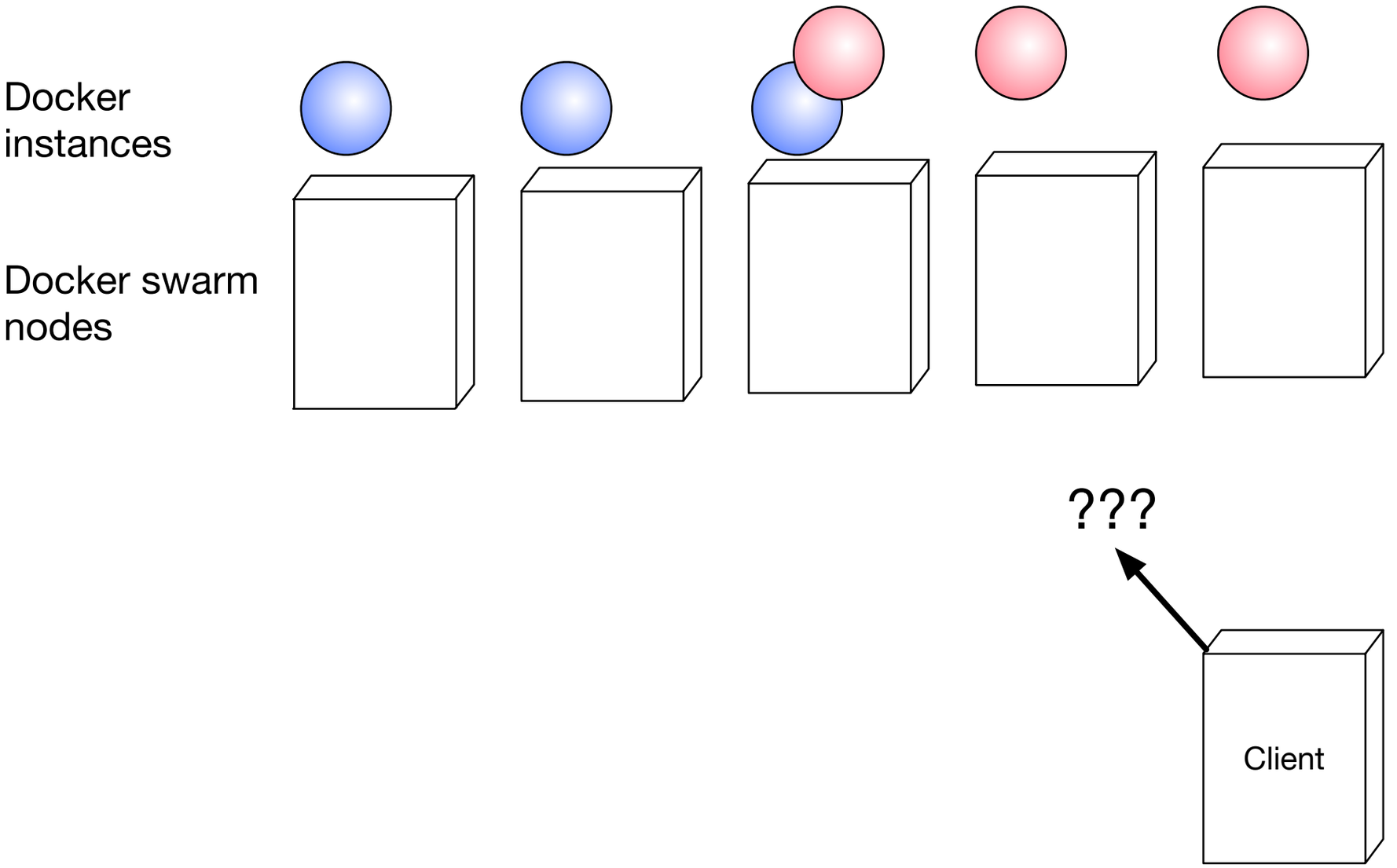
Consul og Docker

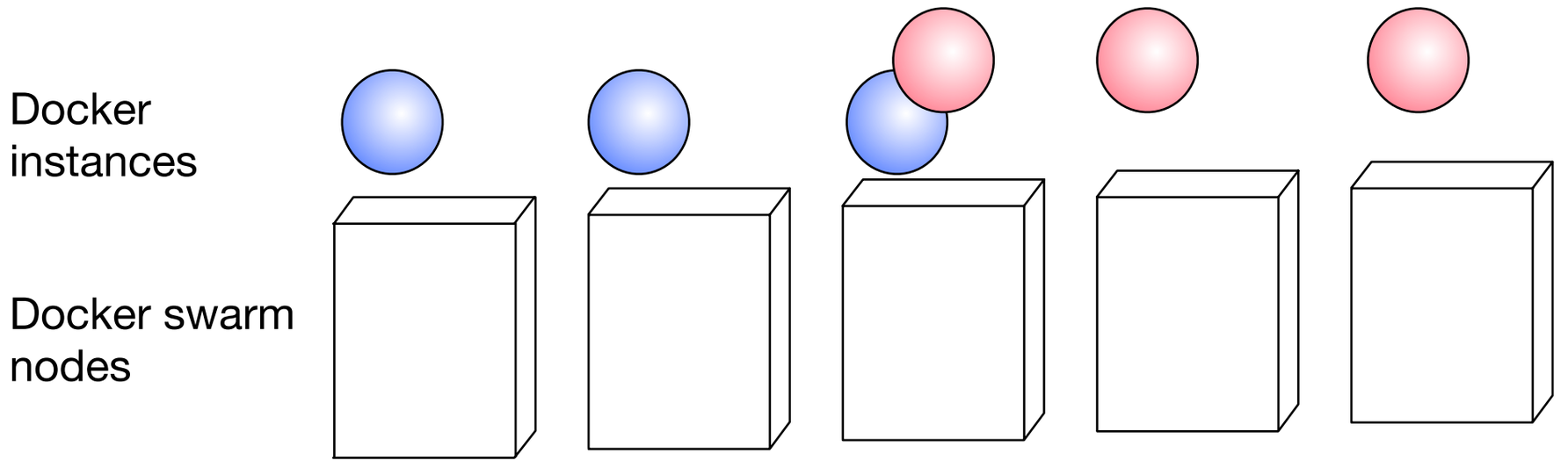
- En utfordring med Consul, er at det kjører en consul agent på hver server som har en tjeneste
- Med containere som Docker, er det egentlig ikke hensikten at hver instans også skal inneholde en slik agent
- Løsningen er å kjøre en agent på hver Docker server sammen med et program som kan automatisk registrere de instansene som kjører

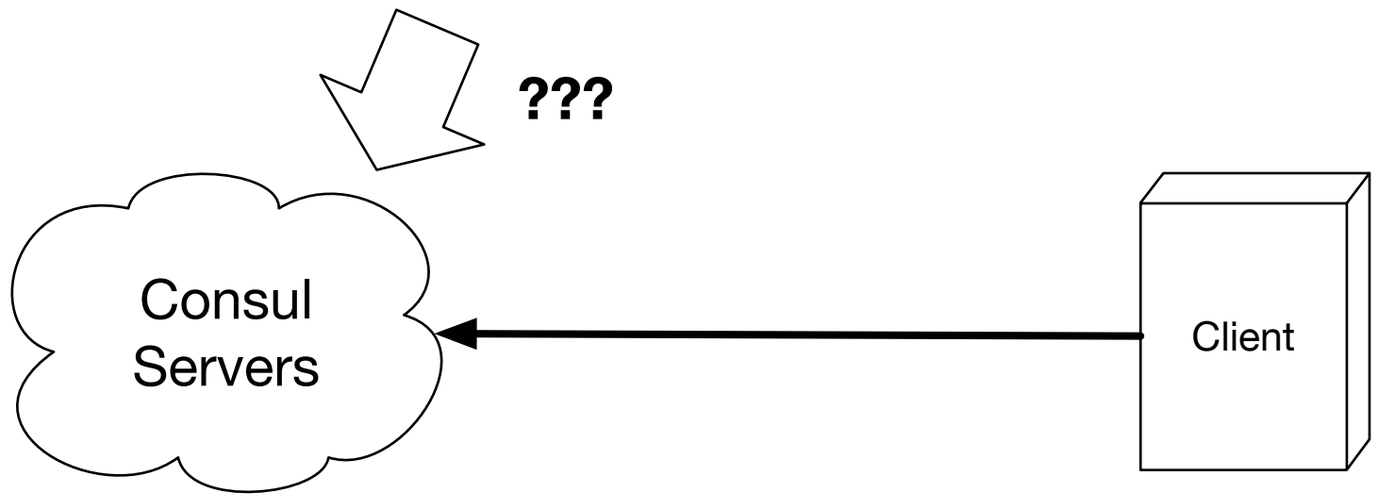
Registrator

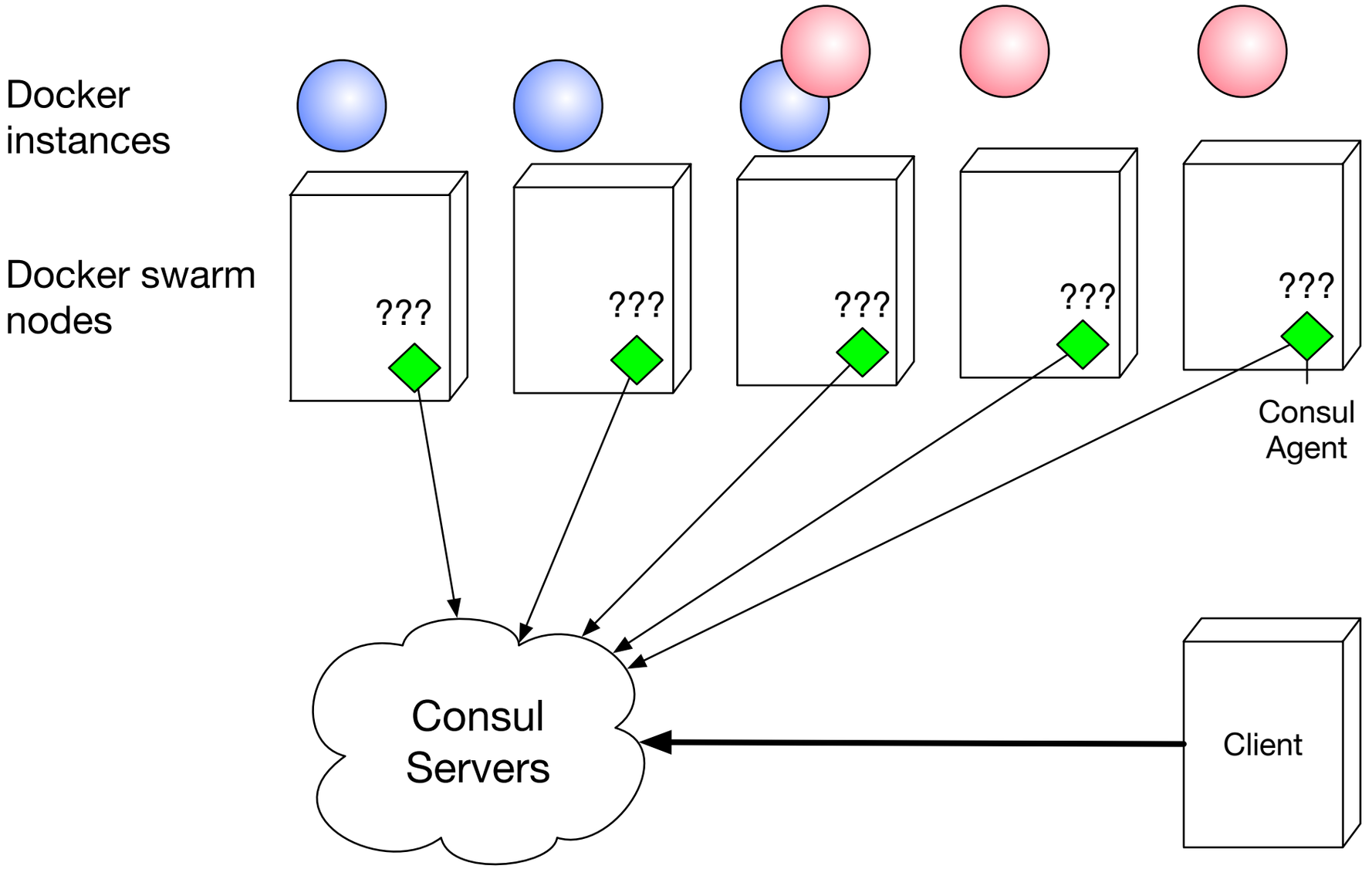
- Registrator er et prosjekt hvor man lytter til Docker tjenesten lokalt på en server og sender inn endringer til en lokal consul agent om hver Docker instans som kjører
  - Man får dermed en automatisk registrering av alle Docker instanser rett inn i Consul
  - Registrator kjører selv som en Docker instans

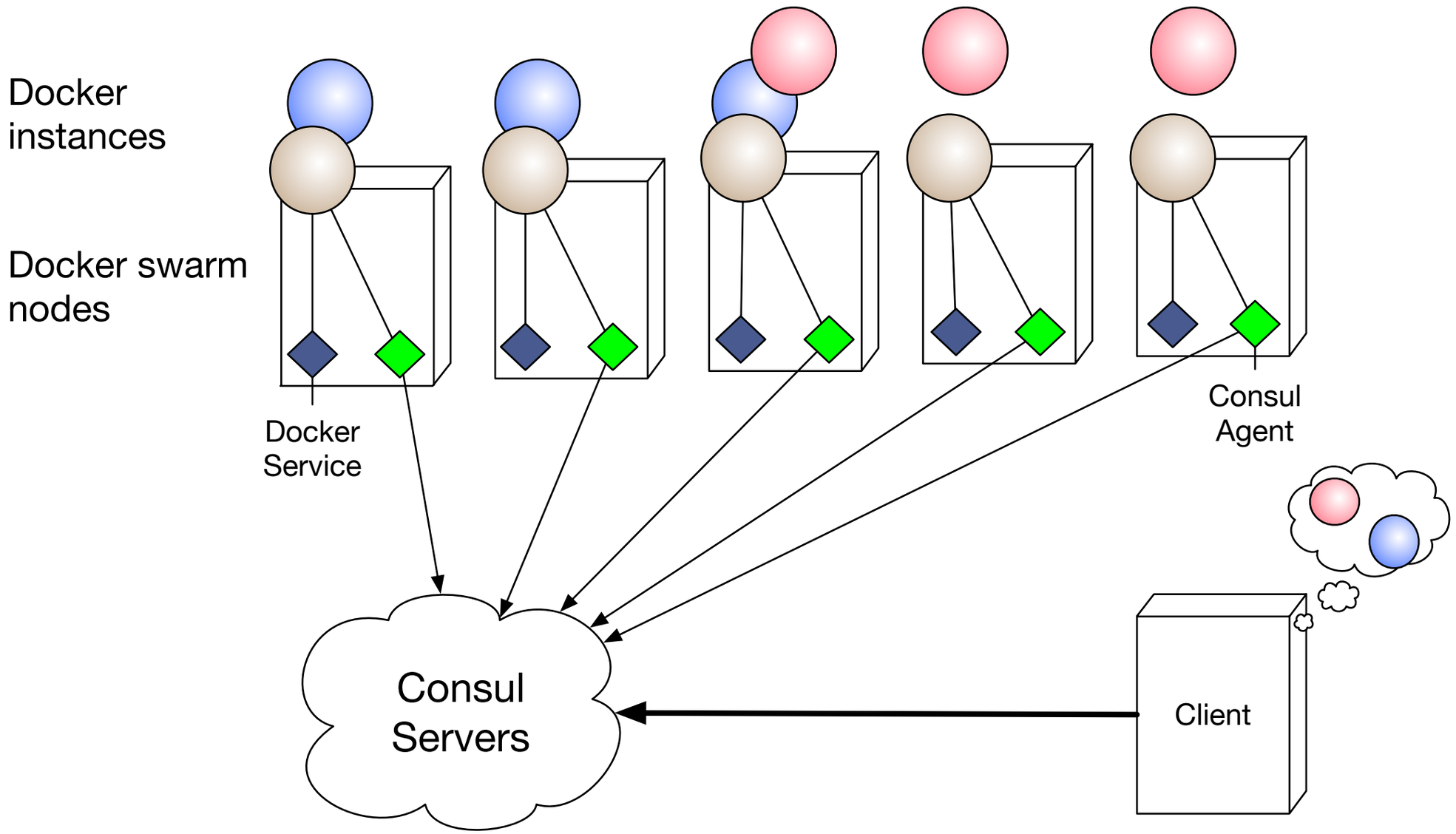
  https://github.com/gliderlabs/registrator


Docker instances
Docker swarm nodes
???
Client


Docker
instances
Docker swarm
nodes


???
Consul
Servers
Client


Docker instances
Docker swarm nodes
???
???
???
???
???
Consul Agent
Consul Servers
Client


Docker instances
Docker swarm nodes
Docker Service
Consul Agent
Consul Servers
Client

Fremgangsmåte

- Installer consul på samtlige Docker swarm noder
- Start consul slik*:

```
consul agent -data-dir /opt/consul -join
consul1-ip -bind local-ip
```

- Start registrator instansen:

```
docker run -d --name=registrator --net=host --volume=/
var/run/docker.sock:/tmp/docker.sock gliderlabs/
registrator:latest consul://localhost:8500
```

- *Dette vil ikke overleve reboot

Spesialnavn på tjenester i Consul

- Registrator vil i utgangspunktet gi navn på tjenesten basert på navnet til imaget + portnummer
- Navnet blir da overført til DNS i consul
- Ønsker man å spesifisere navnet, kan man sende det som hint når man starter tjenesten / instansen i Docker:

```
docker run -d -p 8000:80 -e SERVICE_NAME=helloworld
tutum/hello-world:latest
```

- Nå kan man bruke følgende:

```
dig @consul1-ip helloworld.service.consul ANY
```

DHCP integrering av consul

IMT3003 kyrre.Begnum@ntnu.no

DHCP og DNS

- En vanlig del av DHCP oppsettet, er at DNS server også blir kommunisert gjennom DHCP svaret
- Dette er en utfordring for oss, siden SkyHigh kommer til å gi oss DNS til OpenStack
- Det vi egentlig vil ha DNS til, er vårt egen consul klynge
  - Svaret er å endre DHCP klienten, men det må gjøres på hver enkelt server

Endre DHCP klient oppsett

- På en server, f.eks client, åpne filen
  `/etc/dhcp/dhclient.conf`
- Find linjen som ser slik ut:
  ```
  #prepend domain-name-servers 127.0.0.1;
  ```
- Endre linjen til:
  ```
  prepend domain-name-servers consul1-ip,consul2-ip,consul3-ip;
  ```
- Restart DHCP klienten:
  ```
  killall dhclient
  dhclient
  ```

Test

- Man burde nå se consul adressene i filen /etc/resolv.conf
- Man burde også kunne gjøre følgende:

```
ping db1.node.consul
ping db1.node.dc1.consul
ping mariadb.service.consul
ping mariadb.service.dc1.consul
```

Interessert i mer?

- Det vi ikke har gått igjennom:
  - consul config filer og ordentlig start / stop
  - kryptering
  - flere datasentre
    - bruk av Key / Value
    - Consul templates
- For et eksempel på dynamisk haproxy.cnf basert på innhold i consul se her:
  https://git.cs.hioa.no/kyrre.begnum/consul_deploy
  Se også: www.consul.io