made for free at coggle.it

Fx tegning af en cluster - udtryk for et topologi og arkitektur loadbalancer tegningen er også en topologi Fully connected - fuldt distribueret Hvem taler sammen med hvem? Det kunne ligeså Ring, Mesh, Star, Line, Tree, Bus godt være et organisationsdiagram, der kalder man 🛑 det bare ikke en topologi Består af nogle kabler, routere, computere der taler sammen. Kan du huske hvad den teknologi hedder at vi har private og offentlige netværk --> Det hedder NAT Network Address Router 🗆 et WAN stik i væggen. Så får routeren typisk én IP adresse som er udadtil. Routeren deler denne IP adresse ud til mange forskellige – lokale IP adresser. Offentlig IP adresse udadtil, privat netværk indadtil. DHCP 🗆 dynamic host Netværksarkitektur Nu er Kasper online på sin ipad -> send update til facebook. Hans interne IP er xx. Det første er at den sender den til routeren og siger den skal til facebook. Routeren sender pakken videre med sin egen ydre IP adresse og sender det videre til facebook. Så kommer der en pakke retur til routeren. Routeren har en tabel der siger, at den her pakke der er kommet tilbage skal sendes til denne specifikke lokale ip adresse. Router kan have statisk ip og dynamisk ip adresse. Meget ofte er det irriterende at have en dynamisk. Det kommer an på internetudbyderens opsætning er. De kan have et behov for at dele dem op til et antal kunder. Hvis ISPs kunder overskrider antallet af deres ip adresser, så gør de dem dynamiske. Der er ikke som sådan noget for dig som kunde. Noget lort hvis du Platform architectures are modularizations of complex systems in which certain components (the platform itself) remain stable, while others (the complements) are encouraged to vary in crosssection or over time. Compilere – den oversætter det kode du skriver til maskinekode. Det maskinekode passer til processoren i maskinen. Derfor kan man ikke bare Arkitektur er altså en sammensætning kompilere et program og så sætte det ind i en anden **Processorarkitektur** af nogle moduler, som udgør en computer, fordi der ikke er den samme processor i. opdeling af funktioner, data osv. Indenfor IT er compilere noget af det sværeste at Fundament i et hus – svært at ændre, taggrende – nemt. Nogen ting skal ikke ændre sig så meget... Arkitektur er et spørgsmål om at løse et Det kan dække over hvordan servere i en service skal tale sammen med hinanden. problem, der ikke kan løses med kun et modul. Fx en server En service – bare en eller flere computere som stiller et eller andet tilrådighed for os - det kunne være vejrdata, bank der har en server der viser hvad vi har på vores bankkonto. En **service** med en masse servere der skal tale sammen – eksempel på arkitektur An important property of platform systems is that they are **evolvable**, in the sense that they can adapt to unanticipated changes in the external Når vi taler om arkitektur i distribuerede systen environment. Fx er internettet godt - nye HTML, indler det om den måde, som det hele **hænge**r PHP versioner

Topologi **Platform Layer Maps** OSI modellen fx Serviceorienteret Arkitektur (SOA) vi har nogle services der ►hænger sammen på en eller anden måde DIS - Arkitektur 1 Modulariseringer

An especially common design pattern has come to be associated with the concept of a platform, which we define as a set of stable components that supports variety and evolvability in a system by constraining the linkages among the other components.

En platform er altså en abstrakt brug af et ord, der dækker over en sammenhæng. Når I nu vil udvikle det nye sort, så laver I en platform til det.

Jo mere genbrug, jo bedre - for jo bedre fordelingen af ansvar. Genbrug er interessant i alt hvad man laver. Fordeler noget ansvar. fx har man kun ét endpoint, som står for brugerlogin osv. Et andet endpoint for noget andet. Isolering og genbrug. DRY - Dont repeat yourself

Et meget normalt designmønster er det concept der hedder en platform, og det bliver defineret som en samling af stabile komponenter som kan udvikle sig med andre komponenter Iøbende. Så når vi går ud og laver et startup med en platform der kan et eller andet. Der er nogle servere, et API osv.. Det hele er en platform. Når der kommer en venture capital så køber den platformen..

Most platform definitions focus on the reuse or sharing of common elements across complex products or systems of production. Svarer lidt til den måde, som I deler jeres kode op på. Består af nogle komponenter der kan genbruges. genbrug

Application Programming Interface.

▶ REST API'er --> Twitter, Telefoner osv.

Inden for arkitektur, der er et API altså noget, der udstiller de muligheder, som man har for interaktion. Fx et kald til en server, der returnerer noget data.

Et API er også et udtryk for det, som man kalder design rules. Regler for hvordan man kan tilgå

Enterprisearkitektur

Repræsenterer hvorledes IT taler sammen med forretningen. De to skal gerne komplementærer

Enterprisearkitektur er samspillet mellem virksomhedens forretningsstrategi, dens forretningsprocesser og organisering, og dens teknologiske fundament med det formål at optimere samspillet og deres dele i forhold til at nå de mål, som forretningsstrategien stiller.

Interfaces

Når man taler om microservices – det man mener er at man opdeler noget kode. Dvs. hvis man nu har et API som er meget stort som ligger på én server, så kune det være smart nok at dele det op i nogle dele ,og fordele koden og fordele det ud på flere endpoints

Når man taler om microservices, så taler man om en **opdeling af kode**. **Jo** mindre og flere komponenter, jo bedre ifølge denne diskurs

Containerization

Docker er et populært værktøj til at containerize services.

anisatorisk synspunkt er det jo smart hvis nan deler det op, så flere kan arbejde på mere på samme tid.

> Forestil helt generelt – hvis man har et API ligesom twitters – når man går ind på api.twitter.com – man slår noget op, det får en server til at vågne, npoget kode vågner, og som sender noget returner. Hvis nu at man går ind og det resulterer at man skal igennem 1000 linjers kode, så kan det godt være lidt besværligt, hvis man skal vedligeholde alt det her kode. Lidt problematisk. Fra et organisatorisk synspunkt så er det smart at dele noget

Microservices er et skridt væk fra den klassiske, monolitiske forståelse af et (distribueret) system. Meget ofte er microservices afhængige af hinanden. Hvis en er afhængig af 2 andre, som ikke er færdige endnu, så er det ikke så godt..

kode op.

Microservices

Finally, all platform systems exhibit tensions between platform owners and

udvides løbende.

complementors. These tensions are variously expressed. For internal platforms, tension arises through the threat of entry by third-party component makers who can hook into the platform at its visible interfaces and create compatible substitutes for the firm's own components.

En god arkitektur er en arkitektur, der kan

Hvis man har en platform, og man godt vil udbrede den (fx appel) – her har i nogle API'er som i kan bygge nogle ting og udbygge vores platform med. Så er der nogen der gør det, og så overtager de noget af markedet. Når man bygger en arkitektur og

man godt vil udbygge platformen, så er man eksponeret til at andre kan få en del af

Generelt

sammen på

ændre sig på.

vi taler om arkitektur i distribuerede systemer

så handler det om den måde, som det hele kan