Aine teemade kokkuvõte & Eksami konsultatsioon

Pelle Jakovits

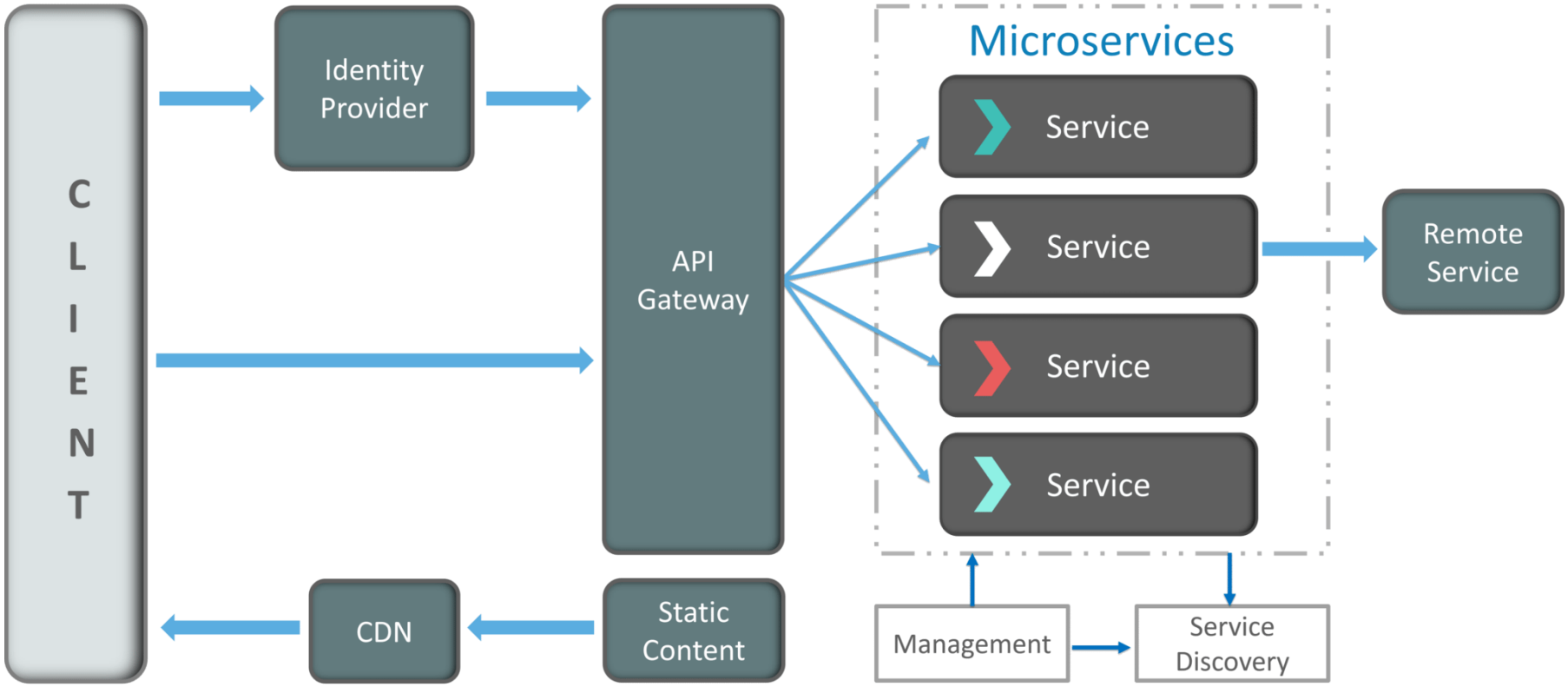
Mai 2024, Tartu

Sisukord

- Aine teemade kokkuvõte
  - Arhitektuuri näited
  - Mustrid
  - Teemade kokkuvõte
- Eksam
  - Korraldus & Reeglid
  - Näite küsimused

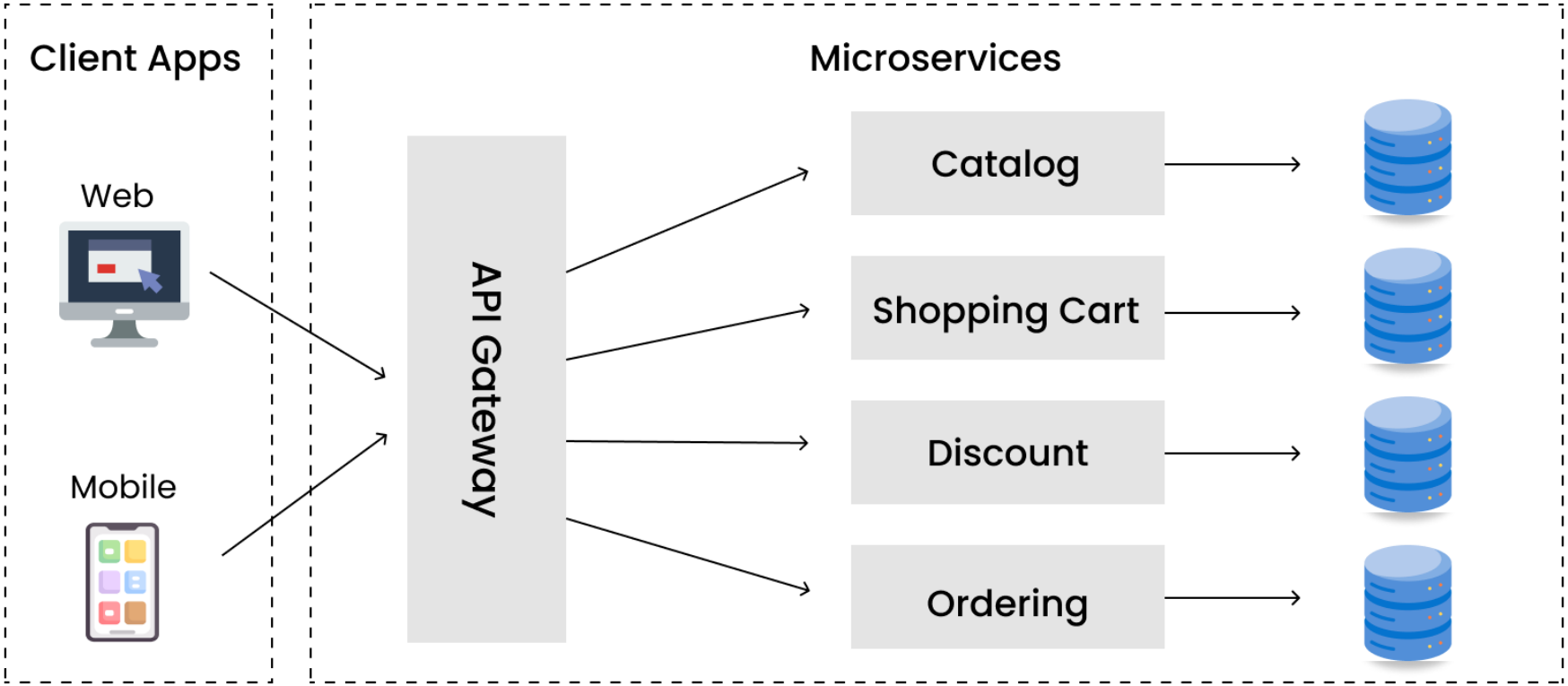
MIKROTEENUSTE ARHITEKTUURI MUSTRID

Mikroteenuste arhitektuuri näide



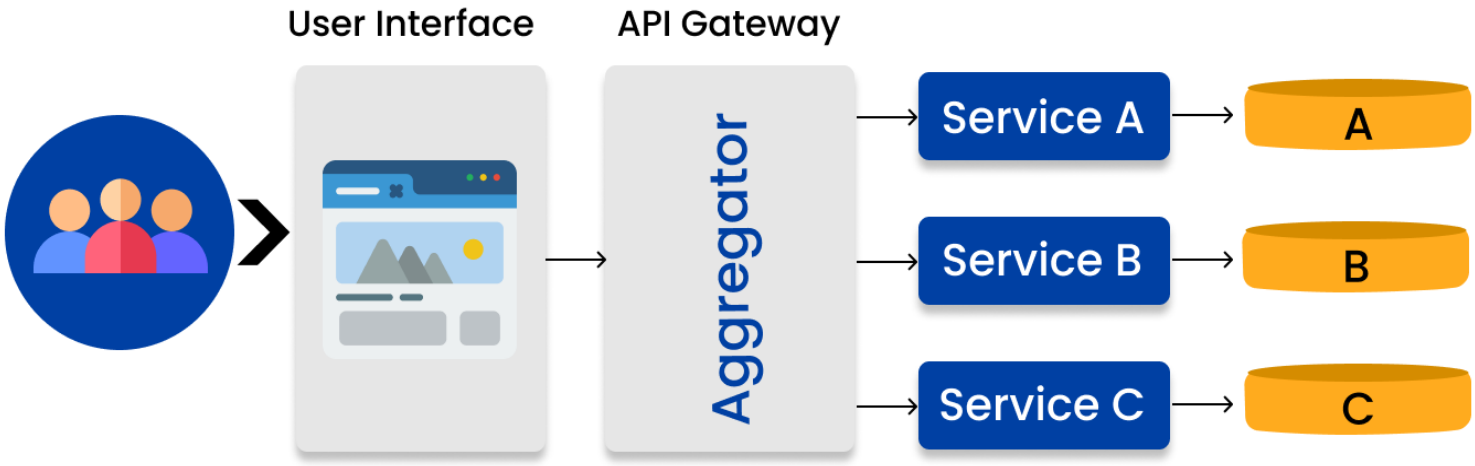
https://dzone.com/articles/microservice-architecture-learn-build-and-deploy-a

Teenuste marsruutimise muster



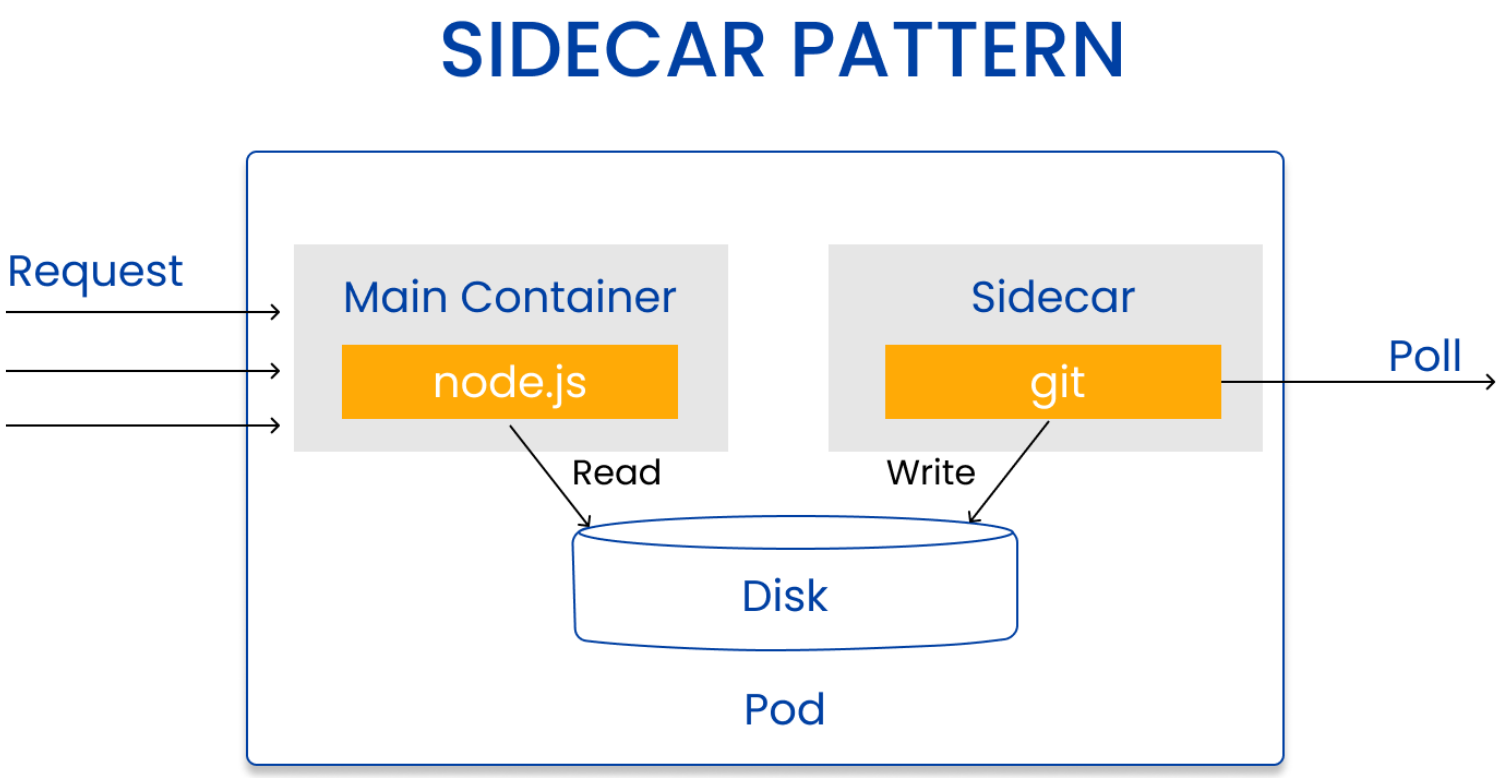
https://qentelli.com/thought-leadership/insights/selecting-microservices-design-patterns-your-business

Teenuse agregeerimise Muster



https://qentelli.com/thought-leadership/insights/selecting-microservices-design-patterns-your-business

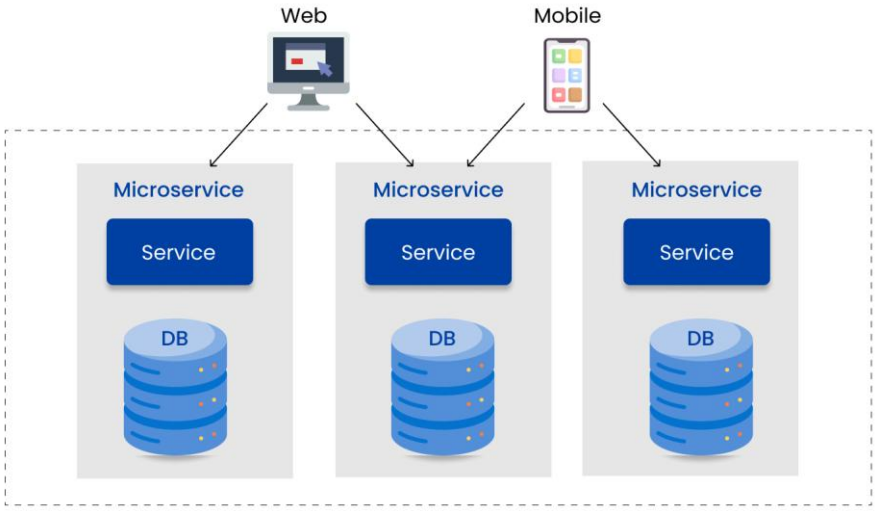
Kõrvalkäru muster



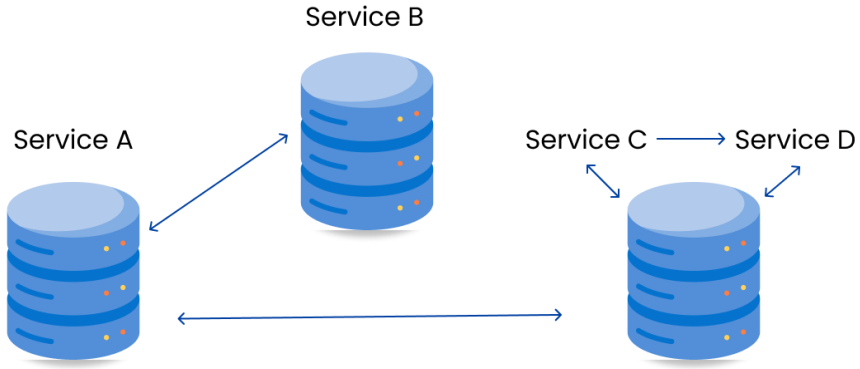
https://qentelli.com/thought-leadership/insights/selecting-microservices-design-patterns-your-business

Teenuste andmebaaside mustrid

Igal teenusel oma andmebaas



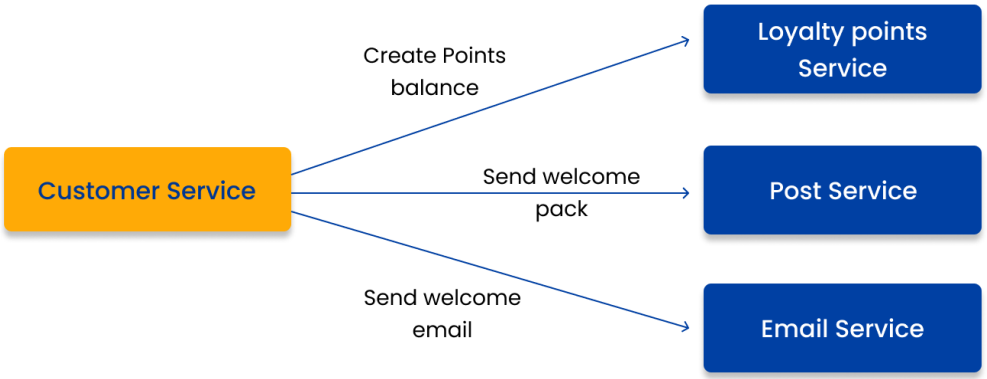
Osa andmebaase on jagatud teenuste vahel



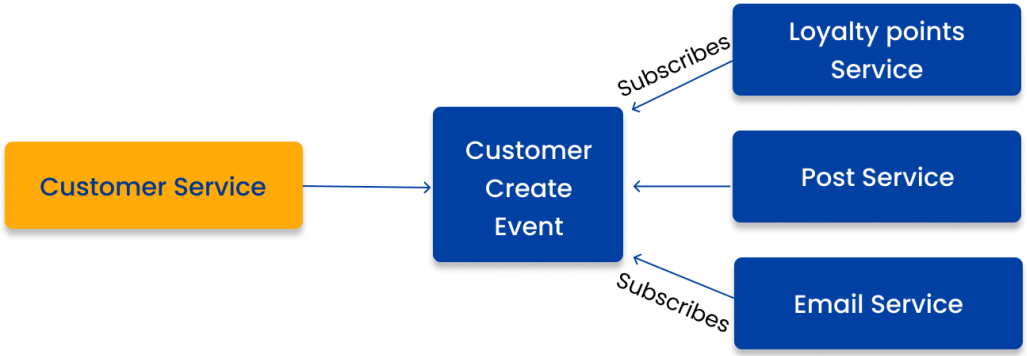
https://qentelli.com/thought-leadership/insights/selecting-microservices-design-patterns-your-business

Orkestreerimise vs Koreograafia muster

Orkestreerimine



Koreograafia

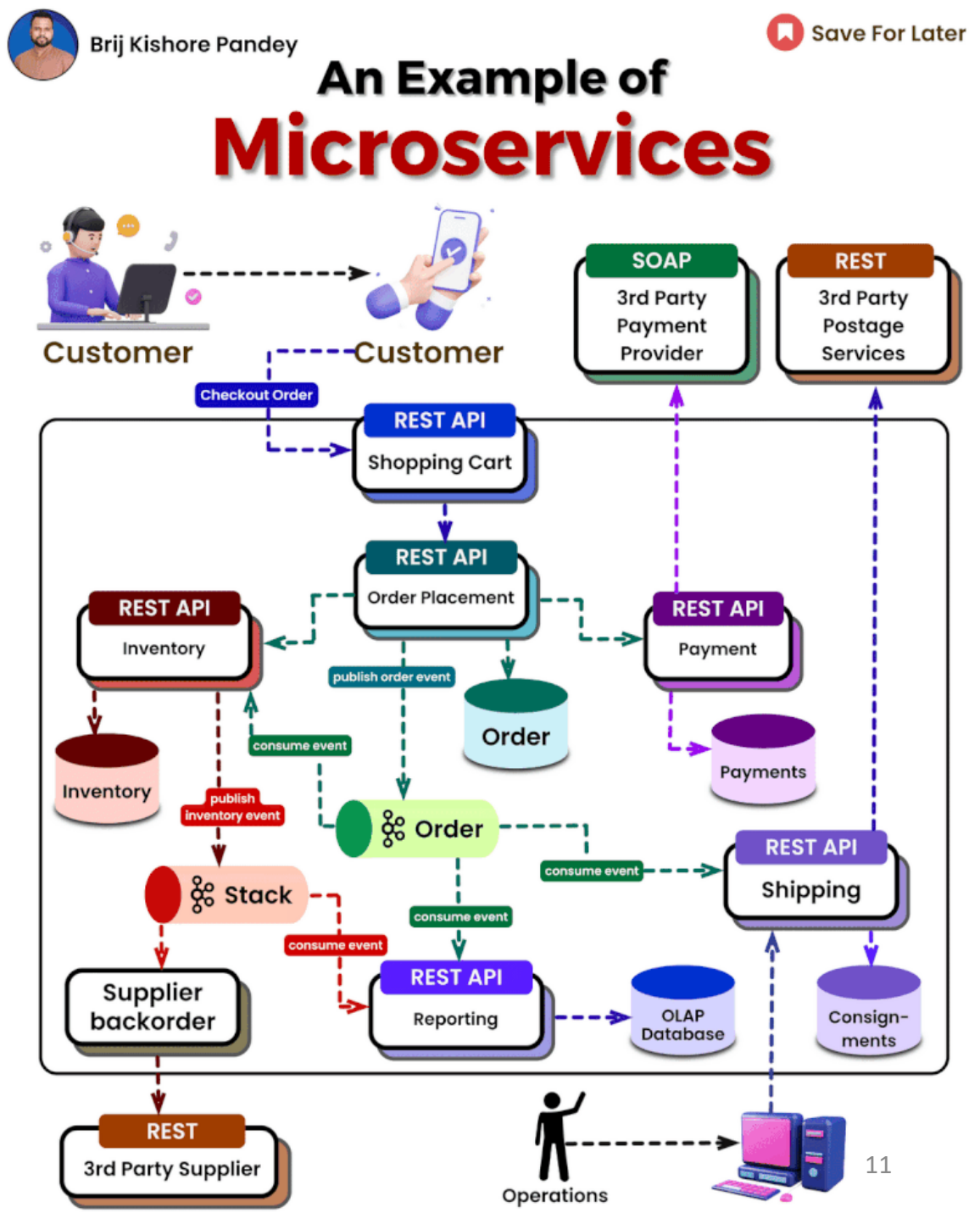


https://qentelli.com/thought-leadership/insights/selecting-microservices-design-patterns-your-business

MIKROTEENUSTE PÕHISTE RAKENDUSTE NÄITED

Mikroteenuste näide

Brij Kishore Pandey

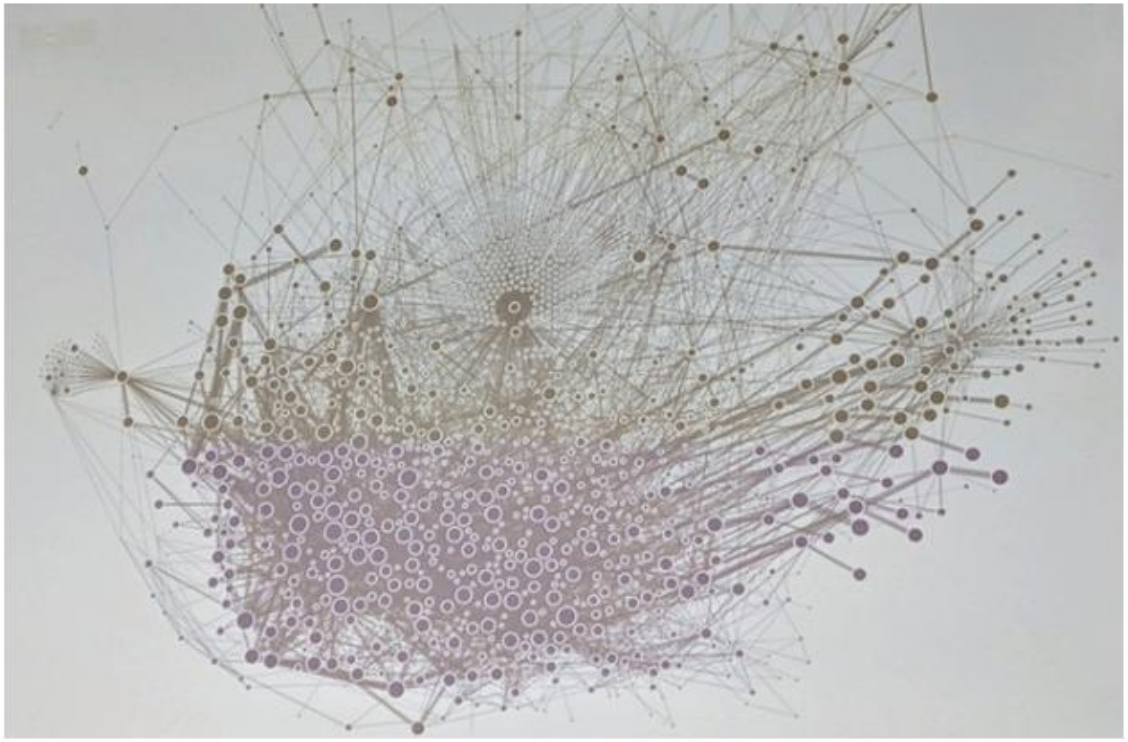


Uber Monoliit

https://nordicapis.com/4-examples-of-microservices-architectures-done-right/

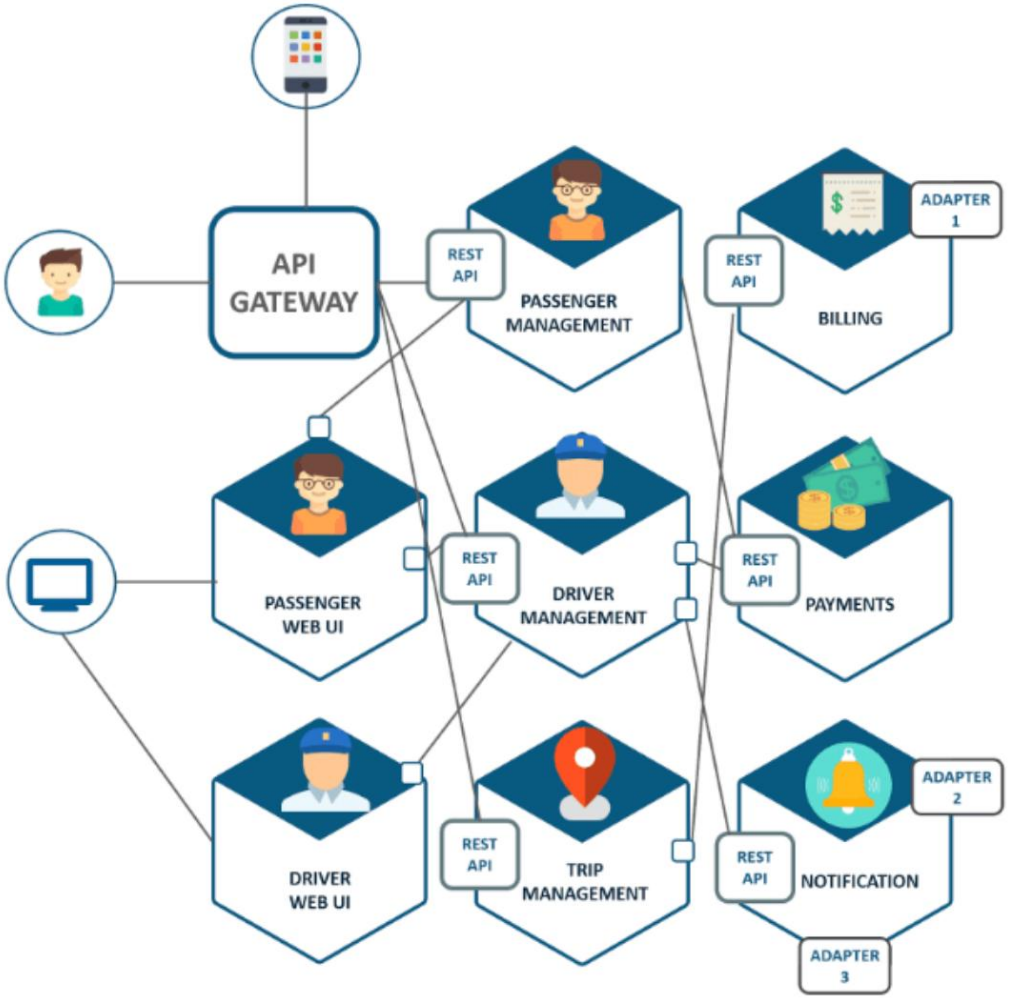
Uberi mikroteenuste suhtluse graafik (2019)

- Standardite puudumine ja mitte kasutamine tegi esialgse mikroteenuste arhitektuuri liiga keeruliseks
- Globaalsete standardite kasutuselevõtt tõi paremini isoleeritavuse ja vähem hapra süsteemi
  - Enam ei olnud vaja ühe mikroteenuse muutmisel ümber olevaid teenuseid muuta
- Hakati rangelt mõõtma ja jälgima mikroteenuste meetrikaid
  - Rikketaluvus, dokumentatsioon, jõudlus, stabiilsus, skaleeritavus
- Mikroteenuste Uuendused pidid rangelt parandama nende meetrikate väärtusi



https://nordicapis.com/4-examples-of-microservices-architectures-done-right/

Uber Mikroteenused



https://nordicapis.com/4-examples-of-microservices-architectures-done-right/

AINE TEEMADE KOKKUVÕTTED

Hajussüsteemide eelised

- Andmete ja arvutusvõimsuse laiali jagamine
- Detsentraliseeritus
- Lokaalsuse omadused ja efektiivsus
- Ressursside kaug kasutus
- Skaleeritavus
- Tõrkekindlus

Virtualiseerimise eelised

- Alakasutatud riist- ja tarkvararessursid
  - Serverist piisab tavaliselt rohkema kui ühe rakenduse jooksutamiseks
- Lihtsustab keskkondade kohandamist rakenduste jaoks
  - Igal rakendusel oma individuaalne keskkond
- Keskkondade ja rakenduste teisaldatavus ja taaskasutatavus
  - Tihti piisab failisüsteemi koopia ja süsteemi konfiguratsiooni liigutamisest
- Lihtsustab haldust
  - riistvara jälgimine, defektse riistvara asendamine
  - serveri seadistamine ja värskendused
  - Süsteemide varukoopiate tegemine

Virtualiseerimise puudused

- Jõudluse halvenemine
  - Lisab uue vahevara/kihi host ja guest süsteemi vahel
  - Rakenduse käivitamine VM sees on aeglasem
  - Iga isoleeritud rakenduse jaoks on vaja oma OS koopiat
- Virtualiseerimisega seotud turvaaugud ja uued ohud
- Näide: Virtualiseerimise puhul on võimalik, et pahatahtlikud programmid saavad end tööle panna enne, kui süsteemi OS käima pannakse.

Konteinerite eelised

- **Effektiivsus** - Vähem OS-i üldkulusid (overhead)
- **Mahutatavus** - Rohkem keskkondasid ühe serveri sisse (võrreldes VM'dega)
- **Teisaldatavus** - Vähem sõltuvusi platvormist = lihtsam liigutada infrastruktuuride vahel
- **Uuestikasutatavus** – valmistame ette üks kord, kasutame mitu korda
- **Kiirus** – Ei ole vaja OS'i üles seada

Konteinerite puudused

- Raske tagada täielikku eraldatust
  - VM'id on selles paremad
  - Reaalselt kasutatakse tihti konteinereid ja VM'e sama aegselt
- Konteinerite sees jooksvate muudatuste tegemine on tihti tülikas
- Tihti on keerulisem saada ülevaade, millised konteinerid on vajalikud, mis on ebavajalikud
  - Võrreldes VM'idega suureneb individuaalsete komponentide arv, mida peab haldama ja millest ülevaadet omama

Mikroteenuste eelised

- Võimaldab suurte rakenduste pidevat tarnimist ja juurutamist (CI/CD)
  - Parem hooldatavus
  - Parem testitatavus
  - Kiirem juurutatavus
- Iga teenus on "piisavalt" väike
  - Lihtsam aru saada, uutele töötajatele tutvustada
  - IDE'sse ei pea importima ülisuuri projekte
  - Teenus alustab jooksmist kiiremini
- Parem isoleeritavus vigade korral
- Vähendab tehnoloogia võlga, sõltuvust tehnoloogia valikutest
- Vabadus kasutada erinevaid tehnoloogiaid vastavalt vajadusele

Mikroteenuste puudused

- Raskem aru saada kogu süsteemi hajutatud arhitektuurist
  - Vaja tegeleda teenuste vahelise suhtlusega
  - Päringud, mis vajavad mitme teenuse välja kutsumist, või andmeid, on keerulisemad arendada ja testida
  - Teenuste vaheliste interaktsioonide testimine on keerulisem
- Kogu süsteemi korraga juurutada, üles seada on keerulisem
- Kogu süsteemi testimine, ja silumine (debug) on keerulisem
- Suurem ressursside (eriti mälu) kasutus, kui igal teenusel on oma (mitte jagatud) keskkond
- Monoliitsed rakendused võivad olla efektiivsemad
  - Jagatud mälu vs sõnumite saatmine
  - Lokaalse võrgu kasutamine konteinerite vahel on suhteliselt aeglane

Pilvepõhiste rakenduste eelised

- Pilved võimaldavad teenuseid ja ressursse tellida ja juurutada reaalajas
- Ettemaksud puuduvad ja võimalik kasutada tasuta kvoote
- Vähem halduskoormust
- Kasutuslihtsus ja mugavus
- Paljud kohandatud pilve teenused on kasutusvalmis
- Automaatne skaleeritavus – sageli hallatakse teenusepakkuja poolt taustal
- Saab juurutada teenuseid kasutajatele lähemale kogu maailmas

Pilvepõhiste rakenduste puudused

- Piiratud kontroll infrastruktuuri ja alusteenuste üle
- Kulusid võib olla raske ette hinnata
  - Kulude optimeerimine võib muutuda keerulisemaks
- Andmete konfidentsiaalsusekaotamise risk
- Suure arvu kasutajate juurdepääsupoliitikate haldamine muutub keeruliseks
- *Vendor lock-in*
- Mis juhtub, kui kellelgi õnnestub pääseda juurde teie pilvekontole?

EKSAMI KORRALDUS

Eksami ajad

- Eksam I:
  - 28 Mai - 12.00 - 13.45
  - Delta – ruumis 1022
- Eksam II:
  - 4 Juuni - 12.00 - 13.45
  - Delta – ruumis 1004
- Järeleksam:
  - 18 Juuni - 12.00 - 13.45

Eksami korraldus

- Eksamile kvalifitseerumiseks peab esitama 80% praktikumi lahendused
  - Aega esitada kuni eksami alguseni.
  - Kuni järeleksamini, kui ei ole kriteerium varem täidetud
- Aine hinde komponendid:
  - 50% - Praktikumi lahendused
  - 50% - Eksam
- Ainest läbi pääsemiseks: Vaja koguda > 50% punkte

Eksami reeglid

- Kirjalik, paberi eksam
- Avatud raamatu eksam
  - Võib kasutada Märkmeid, raamatuid, prinditud materjale
  - Sülearvutit, telefoni ei või kasutada
- Kestvus: 90 minutit
- 3 eksami küsimust

Näite küsimus I

- Hajussüsteemide ning mikroteenuste põhiste rakenduste disainimisel on võimalik kasutada erinevaid viise hajusate komponentide/sõlmede vahelise suhtluse implementeerimiseks.
  - **Küsimus I** (10 Punkti): Tooge välja ning kirjeldage **stsenaarium**, kus teie hinnangul on **sobivam või efektiivsem** kasutada hajusate komponentide vahel otse suhtlust üle **REST (HTTP) päringute** võrreldes **teadete järjekordade** kasutusele võtuga
    - Kirjeldage ning arutage **toetavaid argumente**, miks teie arvates on valitud lähenemine kõige sobivam **selles stsenaariumis.**

Näite küsimus II & III: Stsenaarium

- **Stsenaarium:** *Teid palkab idufirma, mille eesmärk on välja töötada ning käivitada tarkvara, mis võimaldab kasutajatel üles laadida **videosid** selleks, et neist välja otsida erinevaid pilte (stseene), milles leidub kasutaja enda nägu.*
- *Idufirma soovib lahendust luua pilvepõhise (**SaaS**) teenusena.*
- *Nende jaoks on tähtis **kokku hoida kulusid** alguses, kui kasutajate arv veel ei ole suur.*
- *Aga samas on soov, et rakendus **oleks võimeline skaleeruma**, kui kasutajate arv **kiiresti suureneb**.*

Näite küsimus II & III

- Küsimus II (10 Punkti): Millist tüüpi andmebaas (**Relatsiooniline, Võti-Väärtus, Dokumendi-põhine, Veergude perekonna**) oleks teie arvetes kõige mõistlikum kasutada selleks, et hoiustada selliseid videosid ning rakenduse tulemusena loodud pilte?
  - Arutage oma valiku kasutamise **eeliseid ja puudusi** selles stsenaariumis ning **põhjendage**, miks teie arvates on valitud lähenemine sobivam kui alternatiivid (**Võrrelge** vähemalt ühe alternatiiviga)

- Küsimus III (10 Punkti): Kas te soovitaksite kasutada **virtuaalmasinaid või konteinereid** rakenduse komponentide üles seadmiseks?
  - Arutage oma valikute kasutamise **eeliseid ja puudusi** selles stsenaariumis ning **põhjendage**, miks teie valitud lähenemine on sobivam kui alternatiiv.

TÄNUD KUULAMAST!