

30. MAJ 2024

4. SEMESTER PROJECT: GRØN OG OLSEN IT-ARKITEKTUR TRANSFORMATION

AUGUST MUNKHAUS, JEFF B STAMPE, WELAT SIDO, NICOLAJ O.
LAURIDSEN, JAIME CHRISTENSEN

THYSEVEN
Sønderhøj 30, 8260 Viby J

Indholdsfortegnelse

Indledning	2
Formål	2
Indhold og afgrænsning	2
Målbillede	3
Motivation	3
Arkitektur Vision	5
TO-BE Forretningsarkitektur	6
TO-BE IT-Arkitektur	8
IT-Governance & -Management	10
Transition	11
Roadmap	12
Udfordringer & udeståender	13
Brugerinddragelse og Design	14
Interview & Brugerrejse	14
Skitsering	15
Interaktionsdesign	16
Refleksion	18
Proces og metode	19
API Dokumentation & Test	23
Skalering	25
Beskrivelse af IT-Arkitektur og -infrastruktur	26
Load Balancing & Reverse Proxy	27
Monitorering med Loki, Grafana og Prometheus	28
Sikkerhed med HashiCorp Vault	28
RabbitMQ Integration	29
Continuous Deployment med Watchtower	29
Auction Core Services	30
Authorization & Authentication på services	31
Mail Service: Integrationen og Muligheder	32
Interoperabilitet	32
API Dokumentation	33
Model Dokumentation	33
Deployment	33
Access Points	35
Kildekode Repositories	35
Afrunding	36
Referencer	37
Bilag	38



Indledning

Formål

Formålet med denne rapport er at analysere Auktionshuset Grøn og Olsens nuværende IT-arkitektur med fokus på at identificere styrker, svagheder, muligheder og trusler ved at anvende en SWOT-analyse. Ved at evaluere virksomhedens eksisterende situation og foreslå en arkitekturvision, sigter rapporten mod at skitsere en fremtidig digital transformation, der muliggør en skalerbar, sikker og brugervenlig IT-infrastruktur. Målsætningen er at skabe en balanceret løsning, der både understøtter traditionelle fysiske auktioner og udvider virksomhedernes rækkevidde, gennem en ny online auktionsplatform, som sikrer deres fremtidige tilstedeværelse på auktionshusmarkedet.

Indhold og afgrænsning

Rapporten omfatter en grundig analyse af den nuværende AS-IS-situation for Auktionshuset Grøn og Olsen, herunder en detaljeret gennemgang af deres IT-systemer og infrastruktur, datastyring, socialt medie og online tilstedeværelse. Derudover præsenteres en arkitekturvision for TO-BE-situationen, som inkluderer en ny segmenteret løsningsarkitektur med en innovativ online auktionsplatform. Rapporten afgrænses til at fokusere på de interne IT-strukturer og processer samt integration af forskellige teknologier og services, med henblik på at optimere virksomhedens operationelle effektivitet og kundeoplevelser.



Målbillede

Motivation

En SWOT-analyse er blevet udført for at evaluere den nuværende situation:

Strengths	Weaknesses	Opportunities	Threats
Kendt auktionshus med godt omdømme	Meget manuel	At flytte auktionshusets brand til en ny form for aktioner	Konkurrence fra auktionshuset DAB DBA, "Gul og Gratis" og Facebook marketplace
Velkonsoliderede forretningsprocesser (manuelle)	Dårlig udnyttelse af IT		
God styring på (relativt få) data			

I den nuværende AS-IS-situation for Auktionshuset Grøn og Olsen er der flere aspekter af IT-arkitektur og enterprise-arkitektur, der kan identificeres:

IT-Systemer og Infrastruktur:

Auktionshuset anvender forældede IT-systemer, som primært håndterer registrering af effekter og afregning af salg. Disse systemer begrænser virksomhedens muligheder for digitalisering og udnyttelse af moderne teknologi til at forbedre processer og kundeservice.

Datastyring:

Der er en vis styring af data i virksomheden, men den er begrænset til manuelle processer. Der er ikke et integreret datasystem, der kan analysere og udnytte data til at optimere forretningsprocesser eller tilbyde forbedrede tjenester til kunderne.

Social Medie og Online Tilstedeværelse:

Virksomheden har en Facebook-side, men den bruges ikke aktivt. Dette er en savnet mulighed for at nå ud til potentielle kunder og markedsføre auktionshusets produkter og tjenester online.

Digital Kundeservice:

Under forsøg med online chat under Corona-nedlukningen viste det sig at være utilstrækkeligt. Dette indikerer en mangel på en integreret digital kundeserviceplatform, som kunne have forbedret kundeoplevelsen og muliggjort mere effektiv kommunikation med kunderne.

Behov for Digital Transformation:

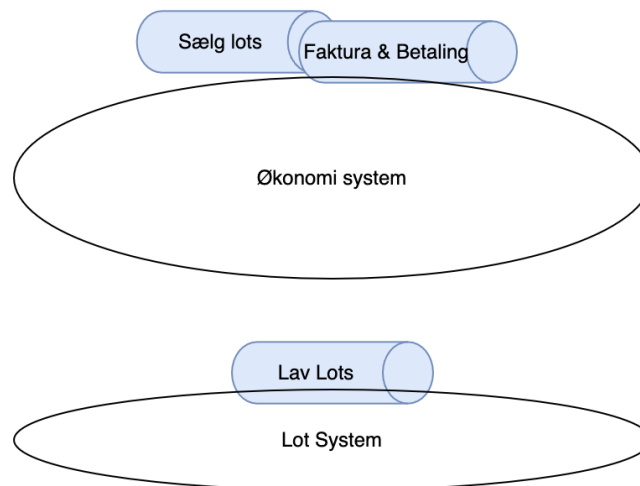
Den overvejende manuelle drift¹ og de forældede IT-systemer udgør en hindring for digitaliseringen af virksomheden. Dette er blevet anerkendt af både ledelsen og eksterne interessenter som den internationale investor, der trak sig tilbage på grund af disse udfordringer.

¹ Bilag 2



Samlet set illustrerer denne AS-IS-situation en virksomhed, der i øjeblikket har en Diversification Operating Model², er i stand til at opretholde sin forretning gennem traditionelle metoder, men som er bagud med hensyn til digitalisering og udnyttelse af moderne teknologi til at forbedre sine processer, udvide sin kundebase o

++++
++++g konkurrere effektivt i det moderne marked.



Figur M.1

² Enterprise Architecture as Strategy.



Arkitektur Vision

Grøn og Olsen står foran en transformation, der bygger videre på deres eksisterende IT-infrastruktur. Den nye løsningsarkitektur er designet som en segmenteret model, hvilket sikrer, at transformationen integreres sømløst uden at forstyrre de kundeoplevelser, hvor de nuværende allerede fungerer optimalt. Denne fremgangsmåde fører til en forfinet arkitektur, samtidig med at det tilføjes en innovativ online auktionsplatform. Denne udvikling bevarer i stor udstrækning de traditionelle kundeprocesser associeret med den fysiske auktionsmetode, samtidig med at det åbner for nye muligheder og effektivitet inden for virksomhedens operationelle rammer. Denne tilgang reflekterer en strategisk og velovervejet udvikling mod et målbillede for virksomhedens Enterprise Arkitektur, der balancerer mellem innovation og kontinuitet.

Vores vision er at skabe en IT-arkitektur der er skalerbar, sikker og brugervenlig, som agilt understøtter både det traditionelle auktionshus og vores nye online initiativer, ved at tilbyde en strømlinet oplevelse for både købere og sælgere af eksklusive genstande, altså lots. Denne arkitektur muliggør en intelligent integration af fysiske og digitale komponenter, som understøtter effektiv registrering, budafgivelse, fakturering og kundehåndtering, samtidig med at sikkerhed og brugernes tillid prioriteres højt.

Vores mål er at føre Grøn og Olsen Auktionshus ind i den digitale tidsalder ved at introducere en separat online platform, der opererer parallelt med vores eksisterende fysiske auktionshus. Vi stræber efter at bevare og styrke vores eksisterende kundebase og processer, samtidig med at vi åbner op for nye kundesegmenter og forretningsmuligheder på det digitale marked. Vores ambition er at skabe en agil og fleksibel arkitektur, der kan tilpasse sig hurtigt skiftende teknologiske og markeds-mæssige ændringer. Ved at være i stand til at forblive tilpasningsdygtige og reaktive over for fremtidige tendenser og krav, ønsker vi at fremtidssikre Grøn og Olsen og sikre dens fortsatte succes og relevans på lang sigt. Med en veldefineret arkitektur og klare principper vil virksomheden være rustet til at navigere succesfuldt gennem denne transformation og sikre sin position som et førende auktionshus i fremtiden.

Arkitektoniske principper

1. Separation af fysiske og online auktionshuse:
 - Udviklingen af IT-Arkitekturen skal ikke påvirke kundens process, & -oplevelse i den nuværende fysiske auktionshandel.
2. Flexibilitet og skalerbarhed:
 - Arkitekturen skal være fleksibel nok til at imødekomme skiftende krav og skalerbar nok til at håndtere både nuværende og fremtidige belastninger.
3. Sikkerhed og databeskyttelse:
 - Der skal implementeres omfattende sikkerhedsforanstaltninger for at beskytte både kunde- og virksomhedsdata mod cyberangreb og uautoriseret adgang.

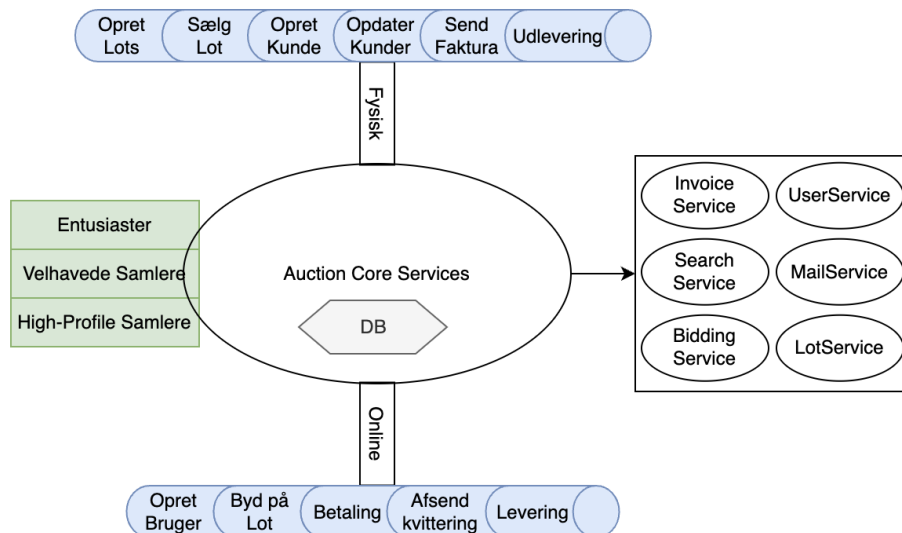


IT principper

1. Cloud-baseret infrastruktur:
 - Udnyttelsen af cloud-teknologi vil muliggøre skalérbarhed, fleksibilitet og omkostningseffektivitet i forhold til infrastruktur og ressourceallokering.
2. Microservices:
 - Opdeling af systemet i microservices, ved brug af Docker, vil lette vedligeholdelse, deployment og skalering af individuelle komponenter.
 - Kontinuerlig udrulning via GitHub Actions automatiserer frigivelse af software efter ændringer, hvilket sikrer hurtig og pålidelig opdatering af produktionsmiljøet - et princip for effektiv og fejlfri udvikling.
3. Repository Pattern:
 - Systemet skal være nemt integrerbart, dette sikres ved opbygning af et interface af repository pattern, for nemt skift af systemintegrationer.
4. Moderne webteknologier:
 - Brugen af moderne webteknologier som Blazor til frontend og ASP.NET til backend vil sikre en responsiv og brugervenlig online platform.

TO-BE Forretningsarkitektur

Forud for denne TO-BE, lavede vi en brugeranalyse ved brug af brugerrejser og interviews, se bilag 1. I disse interviews lagde vi især mærke til, at brugerne var utilfredse, når de skulle hente deres lots, fordi de ikke havde mulighed for at få deres lot sendt. Derfor har vi tænkt over hvordan vi kan løse dette problem i vores kommende forretningsarkitektur.



Figur M.2

Som vist i figur M.2, implementeres der en ny central arkitektur navngivet "Auction Core Services", som udgør hjertet af systemet. Dette område er designet til at håndtere al data og logik, der er afgørende for auktionsprocessen, understøttet af en robust database, som sikrer hurtig og sikker adgang til information. Omkring disse kerneydelser grupperes forskellige



brugerinteraktioner og funktionaliteter, som gør det muligt for os at levere en sømløs og intuitiv brugeroplevelse.

På den operationelle side indfører arkitekturen en række forbedrede processer for salg, kundeåndtering og logistik. Dette inkluderer effektive mekanismer for oprettelse af lots, salg, kundeoprettelse og -opdatering, fakturering og levering. Disse processer er nøje integreret med de centrale auktionstjenester for at sikre en strømlinet operation fra start til slut.

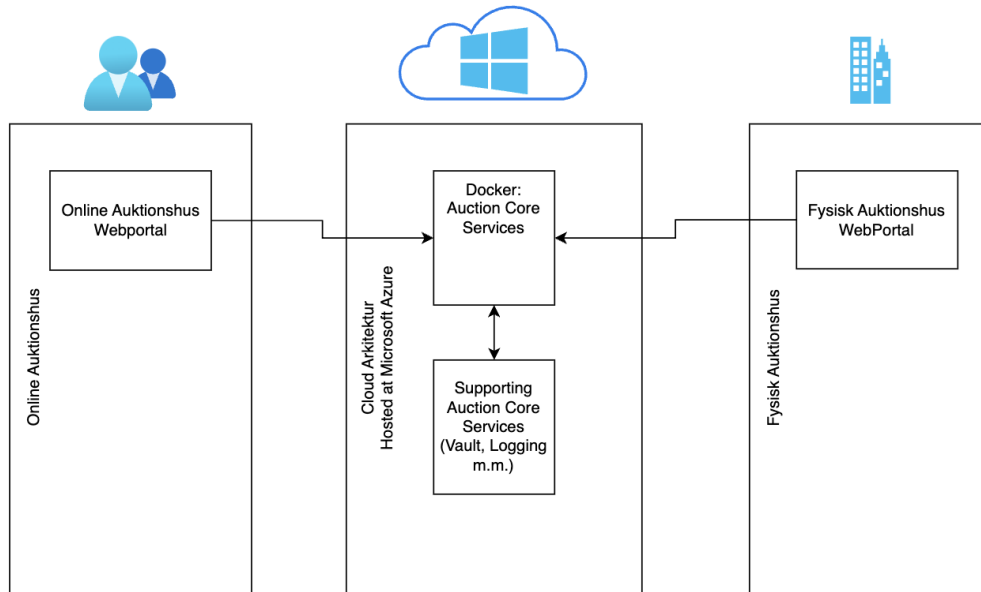
Online interaktioner, såsom brugeroprettelse, budgivning, betaling, samt søgning, er også optimeret for at fremme en nem og sikker brugerrejse. Ved at anvende moderne teknologi løsninger og -platforme, såsom Nginx og JWT, sikrer vi, at brugerne kan navigere i auktionsprocessen med minimal indsats og høj sikkerhed.

Endelig er vores arkitektur udvidet med en række specifikke tjenester, der inkluderer "Invoice Service", "UserService", "MailService", "SearchService", "BiddingService" og "LotService". Disse services er afgørende for at understøtte de varierede aspekter af auktionsplatformen, fra brugeradministration og søgefunktionalitet til budgivning og lot-håndtering.

I det store hele repræsenterer vores fremtidige forretningsarkitektur et fremadtænkende design, der bygger på principperne om separation, fleksibilitet og sikkerhed. Ved at fokusere på at forbedre og udvide kerneydelserne, samt strømline brugerinteraktionerne og operationelle processer, er vi godt positioneret til at levere en exceptionel service, der møder og overgår forventningerne fra Grøn og Olsens kunder.



TO-BE IT-Arkitektur



Figur M.3

Processerne i den nye implementering understøttes ved hjælp af flere services, der ligger under det centrale navn, Auction Core Services, og anvender principperne for microservices til at tilbyde skalerbare og effektive løsninger:

- Invoice service
 - Håndterer fakturering og betalingsprocesserne for købere og sælgere indenfor auktionsplatformen. Dette inkluderer generering, fremsendelse og betaling af fakturaer.
- User Service / Authentication Service
 - Administrerer brugeroplysninger, kontooprettelse, loginmekanismer, brugerprofilindstillinger og sikkerhedskontroller for at sikre, at brugerdata forbliver sikre. Authentication Service sikrer, at kun autoriserede brugere kan tilgå specifikke ressourcer og tjenester, og spiller en kritisk rolle i at beskytte systemet mod uautoriseret adgang.
- Bidding service
 - Ansvarlig for alle funktionaliteter relateret til budgivning. Dette omfatter at placere bud, sender overbuds notifikationer og tillade at lave et nemt gen overbud.
- Lot service
 - Styrer vareregistrering, kategorisering og oprettelse af auktionspartier. Denne service sørger også for opdatering og vedligeholdelse af lotinformation gennem hele auktionsprocessen, inklusiv forsendelse af lot.
- Mail Service
 - Håndterer sendelse af e-mails til brugere for notifikationer, bekræftelser, påmindelser og andre kommunikationsbehov, der er relateret til auktion aktiviteter.



Udover de specifikke services inkluderer Auction Core Services en bred vifte af core funktionaliteter såsom logging, authentication, database integration, queue teknologi, og load balancing. Disse grundlæggende teknologier understøtter ikke alene de enkelte services i at udføre deres opgaver effektivt, men også sikrer høj tilgængelighed, skalerbarhed og sikkerhed over hele platformen. Ved at anvende microservices arkitekturen kan Auction Core Services fleksibelt skalere individuelle komponenter efter behov og hurtigt implementere opdateringer, så platformen altid kører på det optimale niveau.

Adgang til Auction Core Services kan tilgås via 2 portaler:

- Online Auktionshus WebPortal:
 - Den gængse adgang til onlineauktionshuset
- Fysisk Auktionshus WebPortal
 - Personalets adgang til oprettelse, opdatering og salg af lots i det fysiske auktionshus.

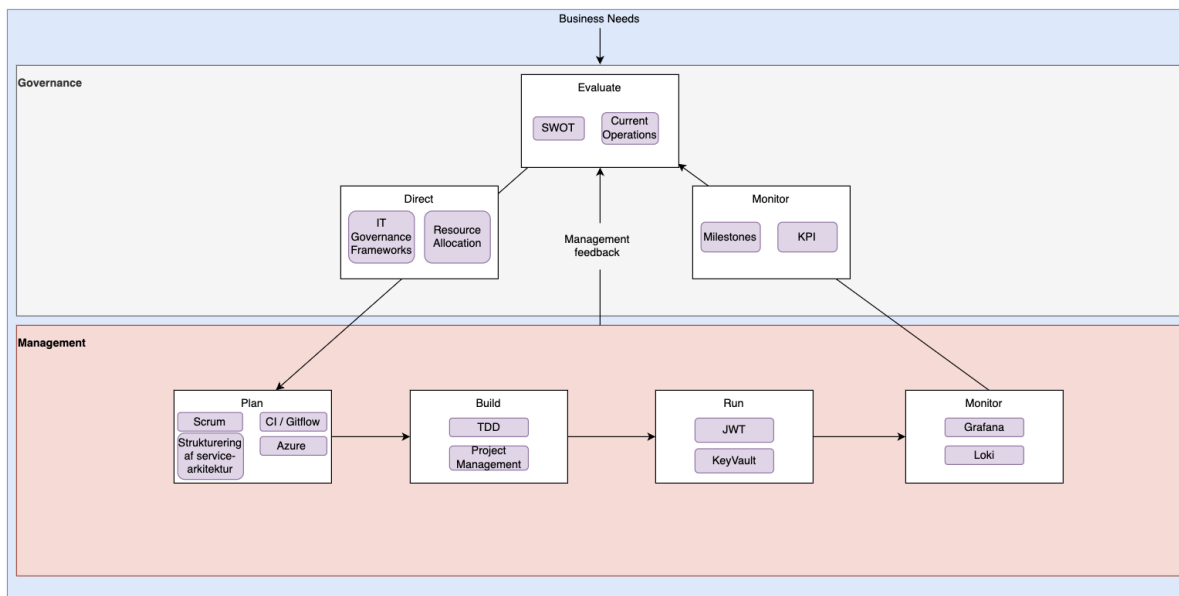
Ved udviklingen af systemet er det vigtigt at prioritere følgende egenskaber for at anvende microservice arkitekturen til det største potentiale³:

- Fleksibel.
- Sikkerhed.
- Skalerbarhed.
- Vedligeholdelsesvenlighed.
- Robusthed.
- Generalitet.
- Åbenhed.
- Pålidelighed.
- Tilgængelighed.

³ Newman, *Building Microservices*.



IT-Governance & -Management



Figur M.4

Grøn og Olsens vision om et fleksibelt, sikkert og brugervenligt IT-system, der understøtter både traditionelle og online auktionsfaciliteter, kræver et klart governance-framework. Dette framework skal håndtere arkitektoniske principper såsom separation af fysiske og online auktionshuse, fleksibilitet, sikkerhed og ressourceallokering.

For at realisere denne version er essentielle IT-management aktiviteter påkrævet, herunder anvendelse af agile metoder som Scrum og Continuous Integration/Continuous Deployment (CI/CD) principper ved udvikling og deployment. Projektledelse spiller en central rolle i at holde teamet fokuseret og opretholde mål og tidsfrister.

Under 'Build'-fasen anvendes Test-Driven Development (TDD) for at sikre kvalitet, mens "Run"-fasen kræver robust drift med sikker autentifikation og databeskyttelse.

Overvågning "monitor" er afgørende med værktøjer som Grafana og Loki for at sikre systemets sundhed og ydeevne.

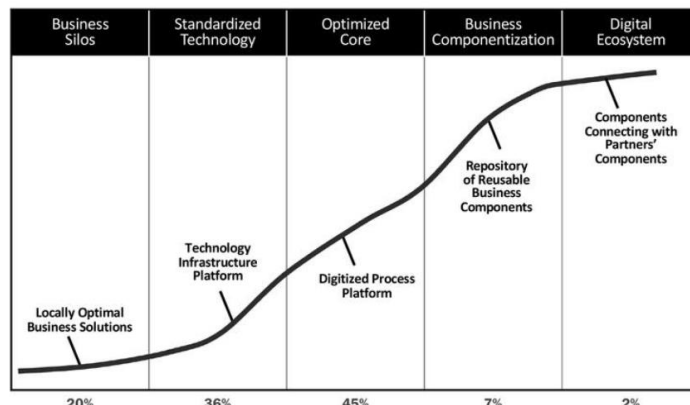
Ved at vedtage en microservices arkitektur vil systemet blive mere vedligeholdelsesvenligt, integrerbart og robust, hvilket giver bedre handlemuligheder og en mere strømlinet drift for både de fysiske og online auktionshuse.

Grøn og Olsens tilgang til IT-governance og -management er dynamisk og tilpasningsdygtig med fokus på fleksibilitet, sikkerhed og brugervenlighed for at trives i det digitale landskab.



Transition

I deres nuværende tilstand har Grøn og Olsen ikke prioriteret deres IT-arkitektur tilstrækkeligt, hvilket fremgår af et modenhedsniveau⁴, der indikerer en fragmenteret og silo-baseret systemstruktur. Grøn og Olsens arkitektur karakteriseres ved en ustruktureret tilgang, der resulterer i betydelige manuelle dataatastningsprocesser og ineffektiv håndtering af forretningsoperationer.



Figur M.5

Som vist i figur M.5, befinder Grøn og Olsens IT-arkitektur sig i en overgangsproces med målet om at opnå en optimeret core. Dette mål indebærer en bevægelse væk fra isolerede forretnings siloer og mod en mere integreret komponentbaseret arkitektur med fokus på standardiserede teknologier og optimerede platforme for digitalisering af forretningsprocesser.

For at realisere denne vision vil Grøn og Olsens arkitektur gennemgå en transformation mod en optimeret core-struktur gennem implementeringen af Auction Core Services. Denne transformation vil foregå i henhold til en veldefineret planlægningsproces, som skitseres i den fremadrettede vejledning – et 'roadmap'.

⁴ Ross, Weill, and Robertson, *Enterprise Architecture as Strategy*.



Roadmap

Grøn & Olsen har skabt et roadmap for at transformere deres traditionelle fysiske auktionshus til en moderne digital platform. Formålet med denne plan er at sikre en vellykket overgang ved at tage højde for både tekniske og brugerrelaterede aspekter gennem en struktureret tilgang.

I begyndelsen analyserer vi vores nuværende auktionsprocesser for at identificere nødvendige digitale tilpasninger. Dette hjælper med at bevare kundeforholdet og skabe en digitaliseringsstrategi, der sømløst integreres med de eksisterende operationer.

Dernæst etableres der et udviklingsteam med de nødvendige færdigheder og ressourcer.

En detaljeret projektplan udarbejdes for at guide overgangen til online auktion. Workshops afholdes for at sikre, at alle i teamet forstår de overordnede mål og strategier.

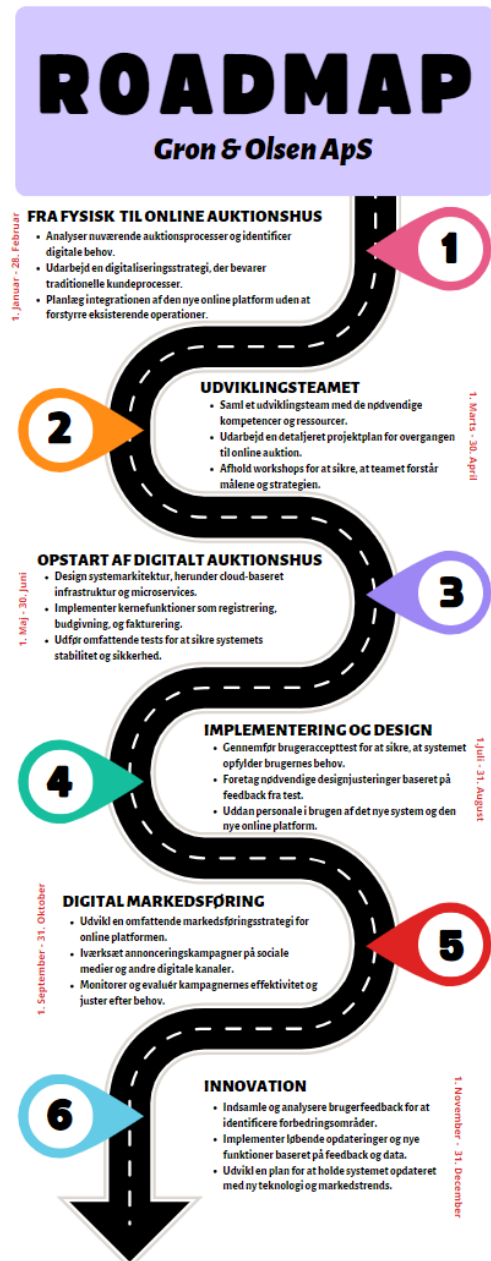
Opstarten af det digitale auktionshus indebærer design og implementering af systemarkitektur, herunder cloud-baseret løsninger. Vi introducerer kernefunktioner som registrering, budgivning og fakturering, og gennemfører omfattende tests for at sikre systemets stabilitet.

Implementerings- og designfasen fokuserer på brugertests for at sikre, at systemet opfylder kundernes behov. Justeringer foretages baseret på feedback, og personalet uddannes i at bruge den nye platform effektivt

For at tiltrække kunder udvikles der en omfattende markedsføringsstrategi. Annoncering Kampagner på sociale medier og andre digitale kanaler sættes i gang, og deres effektivitet overvåges og evalueres løbende

Endelig, i innovationsfasen, indsamler vi brugerfeedback for at identificere forbedringsområder. Løbende opdateringer og nye funktioner implementeres baseret på denne feedback, hvilket sikrer, at vores platform forbliver opdateret og relevant i forhold til teknologi og markedstrends.

Dette roadmap sikrer, at Grøn og Olsen succesfuldt kan transformere til en digital virksomhed, der er klar til fremtidens udfordringer og muligheder, samtidig med at de bevarer deres kerneværdier og kundetilfredshed.



Udfordringer & udeståender

Ved implementering af en ny online auktionshusløsning er der forbundet følgende risici og udfordringer:

- **Kunder**

- Implementeringen af et online auktionshus kan udgøre en udfordring med hensyn til at integrere de nuværende, etablerede budgivere i det nye online format. Derudover kan der være en risiko for, at deltagerne bliver skræmt væk, hvis Grøn og Olsen ophører med at afholde fysiske auktioner.

Mitigering:

Da de nuværende kunder stadig er vores fokusgruppe, vil auktionshuset ikke stoppe de fysiske auktioner, der fortsat er indtjenende.

- **Sikkerhed**

- At opdele en applikation i microservices skaber flere angrebsflader, hvilket øger behovet for robuste sikkerhedsforanstaltninger. Implementering af omfattende autentifikations- og autorisationsstrategier, sikring af kommunikationskanaler mellem services og beskyttelse af følsomme data bliver essentielle opgaver.

Mitigering:

Et særskilt godkendelsessystem, JWT Token Based Authentication, der sikrer adgang til diverse API-kald, og en central vault, der sikrer og styrer alle systemkoder.

- **Overvågning, logging og fejlfinding**

- Med mange små komponenter spredt over forskellige servere og infrastrukturer kan overvågning og logging blive en udfordring. Man skal have strategier på plads til at aggregere logs, spore transaktioner på tværs af services og hurtigt identificere og isolere fejl.

Mitigering:

Anvendelse af logging services, Loki og Grafana mitigere disse risici.

- **Teamkoordinering og versionsstyring**

- At opdele en applikation i flere services kan øge kompleksiteten af teamkoordinering, da forskellige teams kan arbejde på forskellige dele af systemet. Governance, management, versioneringsstrategier og tydelige kommunikationskanaler bliver afgørende for at sikre en koordineret indsats og konsistens i systemet.

Mitigering:

For at styrke kodebasens sikkerhed anvendes GitHub's versioneringskontrollsystemer.



Brugerinddragelse og Design

For at få indsigt i den nuværende situation for eksisterende online auktionshuse, begyndte vi processen med at interviewe og forhøre os hos tidligere og nuværende kunder af både fysiske og online auktionshuse. Ved at gøre dette kunne vi få en overordnet idé om, hvad der bidrager til en positiv oplevelse med et online auktionshus, samt hvad der skaber negative oplevelser.

Interview & Brugerrejse

Konklusioner fra interviews med tidligere auktions kunder

Positive Aspekter:

- **Aflysning af bestilling:** Muligheden for at annullere en bestilling skaber tryghed hos brugerne og giver dem en følelse af kontrol over deres køb.
- **Levering (hvis tilbudt):** Tilbud om levering bliver værdsat højt, da det sparer tid og besvær for kunderne.
- **Shopping følelse:** Online auktioner giver en lignende følelse som shopping, hvilket kan være underholdende og spændende.
- **Mulighed for at tjene/spare lidt penge:** Chancen for at finde gode tilbud eller at tjene penge på at sælge egne varer er attraktiv for mange brugere.
- **Følelse af at vinde (konkurrence):** Auktionens konkurrenceelement kan være spændende og tilfredsstillende, da det giver en følelse af at vinde noget værdifuldt.

Negative aspekter:

- **Angivne pris er før moms og salær:** Bruger forvirring og skuffelse opstår, når den angivne pris ikke inkluderer moms og salær, hvilket gør den endelige pris højere end forventet.
- **Forvirring omkring betaling:** Komplexiteten i betalingsprocessen kan være frustrerende og afskrækkende for kunderne.
- **Lots kan virke forvirrende:** Måden, hvorpå varer er opdelt i lots, kan være forvirrende, især for nye brugere, hvilket kan føre til fejl og utilfredshed.
- **Afhentning kan være upraktisk og forvirrende:** Manglende klarhed omkring afhentning procedurer og upraktiske afhentningssteder kan skabe betydelige udfordringer for kunderne.



Ud fra disse interviews konstruerede vi en brugerrejse⁵, som kortlagde de forskellige stadier, en bruger gennemgår, fra det øjeblik de besøger et online auktionshus, til de enten har vundet en auktion og modtaget varen, eller besluttet sig for at afbryde købsprocessen. Denne brugerrejse hjalp os med at identificere nøgleområder, hvor brugernes oplevelse kan forbedres, såsom i informationsformidlingen, betalingsprocessen og logistikken omkring afhentning og levering.

På baggrund af de konklusioner, vi kunne drage fra interviewene, samt brugerrejsen, kunne vi begynde at formulere konkrete designændringer. Vores designændringer fokuserer på at gøre priserne mere gennemsigtige ved at inkludere moms og salær i de viste priser, forenkle betalingsprocessen, forbedre informationen om lots, og gøre udleveringsprocedure mere brugervenlige. Ved at implementere disse ændringer forventer vi, at online auktionshuse kan tilbyde en mere strømlinet og tilfredsstillende oplevelse for deres brugere, hvilket vil øge både kundetilfredsheden og loyalitet.

Skitsering

Til opstart af designdelen, begyndte vi med at skitsere vores individuelle idéer, til hvordan vi forestillede os at en online auktionsside kunne se ud. Ud fra disse skitser⁶ kunne vi så diskutere hvilke elementer, for hver skitse, der havde potentiale og hvad vi eventuelt skulle prøve på at undgå. De 5 indledende skitser gav os en generel ide til platformens udseende, hvor de bedste ideer blev blandet sammen og videreudviklet i næste iteration.

Platformens design blev videreudviklet i et digitalt værktøj, hvor der til start blev arbejdet på platformens forside. Her udviklede vores skitser sig hurtigt til at blive mere high-fi. Vores fokus på udseendet var først og fremmest at indbringe vores Core services, så platformen ville afspejle systemets funktioner. Dette indebar auktionslisten og byd funktionen. Herefter gik overvejelserne på, hvad forsiden skulle indeholde. Her kom et slideshow med billeder på og en midlertidig liste over populære auktioner. Dette gav forsiden en bedre visuel udstråling, da midterdelen ikke stod tom, eller at det første møde med platformen ikke var et krav på at logge ind. Vi tilføjede kategorier i venstre side samt normale knapper i toppen til fx, about og nyhedsbrev. Dette var mest for at give platformen et hjemmeside-look ved at anvende en velkendt brugergrænseflade og intuitivt design.

Vi designede derefter siden for alle auktioner/lots og siden for en individuel auktion. Her tog vi igen en idé fra den første iteration og udviklede den. Tanken var at holde designet så simpelt som muligt, ved at undgå støjende elementer. På listen over auktioner tilføjede vi en søgefunktion og en filterfunktion. Her inddelte vi hver enkelt lot i sin egen boks for at forbedre brugervenligheden, ved at anvende klar mapping og tydelige signifiers, hvilket skaber et overskueligt og intuitivt layout. På byd siden fremvises en auktion med flere detaljer. Her blev der tilføjet informationer om pris, dato og lokation. Billeder og beskrivende tekst vises også. En liste over budhistorikken samt "afgiv bud" knappen blev også tilføjet. Et pop-up vindue til at udføre et bud blev til sidst designet.

⁵ Bilag 1: Brugerrejsen

⁶ Bilag 3, 4, 5, 6, 7: Skitser



Efter skitserings-fasen konkretiserede vi vores idé til platformen. Vi præsenterede layoutet og konceptet gennem en færdig mock-up af udvalgte sider på hjemmesiden. Selvom arbejdet ikke blev opdelt i så mange konkrete iterationer som planlagt, diskuterede vi vigtige overvejelser, såsom brugervenlighed og navigationsstruktur, under udviklingen af mock-uppen. Som det fremgår af vores skitser, blev de meget detaljerede, da vi havde brug for et fyldestgørende billede af, hvordan siden skulle se ud. Dette var nødvendigt for at kunne implementere og anvende vores viden fra interaktionsdesign, herunder principper om intuitiv brugerinteraktion og visuelt hierarki. Vi valgte specifikt at fokusere på hjemmesidens landingsside og produktside, da disse er centrale for brugeroplevelsen, samt førstehåndsindtryk og krævede særlig opmærksomhed.

Interaktionsdesign

I designet af vores auktionsplatform har vi nøje overvejet brugen af affordances, mapping, signifiers og feedback for at sikre en intuitiv brugeroplevelse. Ved at benytte signifiers, som de grønne knapper med tekst som "Afgiv bud" og "Alle auktioner", har vi gjort det klart, hvilke handlinger brugeren kan foretage sig, og vi har sørget for at disse knapper skiller sig ud ved at bruge en iøjnefaldende farve. Dette blev lavet på baggrund af spørgsmålet "Hvordan får vi brugeren til at klikke på Alle auktioner", da vi ønsker at vejlede dem hen til listen over auktioner.

Alle auktioner

Om auktionerne ▾

Om os ▾

Nyhedsbrev

Mapping er blevet brugt til at sikre, at elementerne på siden er placeret logisk og intuitivt. For eksempel er budhistorikken placeret til højre for beskrivelsen af auktionen, hvilket gør det nemt for brugeren at se tidligere bud uden at skulle scrolle for meget. Historikken er opdelt efter ID, antal kr. og tidspunkt, som gør det overskueligt at læse. Derudover er kategorimenuen til venstre organiseret på en måde, der afspejler brugerens forventning om, hvor de kan finde bestemte typer af auktioner.

Bud ID	Bud	Bud tidspunkt
200208	61.000 DKK	06:24:47 - 21.05.2024
200207	61.000 DKK	06:24:46 - 21.05.2024
200206	60.000 DKK	06:24:27 - 21.05.2024
200205	59.000 DKK	06:24:27 - 21.05.2024
200204	58.000 DKK	06:24:27 - 21.05.2024

Signifiers er tydeligt brugt i designet for at guide brugeren. For eksempel viser den lille pil til venstre og højre i slideshowet, at brugeren kan navigere gennem billederne. Derudover viser prikkerne, under billederne, hvor mange billeder der kan ses og hvor langt henne man er. Vi har også brugt visuelle signifiers som scrollbar til højre for budhistorikken, der indikerer, at der



er flere bud end dem, der vises umiddelbart. Pilene under visse knapper / tekster indikerer også at der kan vises flere valgmuligheder / informationer.

Feedback er en central del af vores design for at forsikre brugerne om, at deres handlinger har haft den ønskede effekt. Når en bruger vil afgive et bud, vises en pop-up hvor man kan lægge et nyt bud, hvilket giver øjeblikkelig feedback på deres handling. Desuden opdateres den nuværende pris straks på auktionens hovedside, hvilket hjælper brugeren med at holde sig informeret om status uden at skulle genindlæse siden. På listen over alle auktioner, lyser den valgte auktion op med en anden farve, hvor museikonet også skifter til den trykkende hånd, som indikerer at der kan trykkes på den. Kategori knappen har også en lille streg under teksten, når kategorierne er åbnet, for at vise at den er åben.

Vi har også taget hensyn til forskellige typer af brugerinteraktioner som request, demand, encourage, discourage, refuse og allow⁷. For eksempel, ved at have en grøn "Afgiv bud" knap, opfordrer vi (encourage) brugeren til at byde. Låseikonet ved login indikerer, at brugeren skal logge ind for at få adgang til visse funktioner. Her er der først tale om et request, Da ikonet indikerer at et login ville låse op for flere muligheder. Hvis man senere på siden forsøger at byde på en auktion uden et login, vil et demand fremkomme, at et login er nødvendigt for at fortsætte, hvis ikke der også er tale om refuse, som afslår en handling. Den klare struktur og opdeling af sektioner gør det nemt for brugerne at finde det, de søger (allow), mens en detaljeret budhistorik hjælper med at informere brugerne om tidligere aktiviteter, hvilket kan påvirke deres beslutningstagning (inform).

Et andet eksempel kunne være, at efter man har klikket på "Afgiv bud", så kommer der et pop-up, hvor man kan se den nuværende pris og afgive et nyt bud. For at lede brugeren i retningen af et nyt bud (encourage), har vi inkluderet en vejledende mindstepris, skrevet med lavere opacitet. Derudover vil byd-knappen også være af lavere opacitet (refuse), hvis ikke der er indtastet et bud i feltet. Hvis der gives et bud, under det mindste tilladte bud, vil der blive givet en fejlbesked til at informere brugeren (inform).



På hjemmesiden <https://gronogolsen.dk>, eller, <https://ui.gronogolsen.dk> kan workflowet / den foretrukken rejse gennemgås fra forsiden til og med at afgive et bud. Her er der dog antaget at brugeren er logget ind.

Alt i alt har vi arbejdet målrettet med disse designprincipper for at skabe en auktionsplatform, der er brugervenlig, intuitiv og funktionel. Fra brugerrejsen til et færdigt PoC, har vi undersøgt behovet og anvendelsen fra kundens perspektiv, analyseret os frem til de nødvendige funktioner og ønsker, samt designet et mock-up på baggrund af den bagvedliggende analyse.

⁷ Davis, *How Artifacts Afford*, 63.

Refleksion

Efter arbejdet med design-fasen samt analysen af brugerrejsen kan et par ting nævnes. Hvis vi tidligere i projektet havde udviklet flere brugerrejser med, personaer og user-stories, havde formålet med brugerrejsen givet mere gavn, i forhold til design-valg. Vores enkelte brugerrejse gav os et godt indblik i hvilke "services" vores auktionshus skulle have med, samt ting vi skulle gøre brugeren opmærksom på, fx levering/afhentning. Brugerrejsen blev lavet ud fra interviews vi gennemførte med 6 personer. Omkring halvdelen af de udsurgte havde reelle indsigter i brugen af auktionshuse. På bagkant havde flere gennemførte interviews med eksperter, været en fordel for at identificere mulige painpoints. Havde der derfor været mere arbejde med brugerrejsedelen, ville der kunne opdages flere mulige tiltag for designet af brugergrænsefladen.

Vores skitser og design fase blev startet løst op, uden en fastlagt plan for hvordan faserne skulle forløbe sig. Dette skete mest af alt på baggrund af travlhed med arbejdet på programmering, hvor vi delte gruppemedlemmernes fokus op, så det ikke var alle der arbejdede på det på samme tid. Vi var selvfølgelig alle inde over første iteration af skitser for at få lagt så mange mulige ideer på bordet. Efterfølgende blev arbejdet mere koncentreret på at få udarbejdet en "funktionel" mock-up af 3 af vores sider, der skulle illustrere vores services. Her gled vi hurtigt væk fra at opdele arbejdet i iterationer, hvilket gjorde det sværere at reflektere og dokumentere ændringer undervejs. Derimod har det med stor fordel været godt for designet at vi har dykket i detaljer og arbejdet, mod et næsten færdigt design, da en ordentlig visualisering har gjort det nemmere at overveje design valg, såsom placering af knapper, farver og overvejelser om hvordan brugerens opførsel ville forløbe sig på hjemmesiden. Generelt havde vi også forestillet os at skitsere login og betalings muligheder, men på baggrund af løs planlægning blev det ikke prioriteret.

Efter arbejdet med brugerrejsen og design, kan der sikkert konkluderes at en grundigere planlægning havde været til vores fordel. Samtidig kan der undres over om kreativiteten ville blive påvirket af for grundig planlægning, i form af fastlagte iterationer, da vores bedste inputs og ideer kom imens vi arbejdede fokuseret på mock-up designet med detaljerne for øje.



Proces og metode

Under transformationen af Grøn og Olsens online auktionshus har vi anvendt principper fra Extreme Programming⁸ (XP) for at sikre en effektiv og pålidelig udviklingsproces. Nogle af de væsentligste XP-principper, vi har integreret, inkluderer:

- **Parprogrammering (Pair Programming):** To udviklere arbejder sammen ved samme computer for at skrive kode, hvilket fremmer samarbejde og sikrer højere kvalitet af koden gennem kontinuerlig feedback og fælles problemløsning.
- **Testdreven udvikling (TDD):** Vi udvikler koden ved først at skrive tests, som definerer den ønskede funktionalitet, før vi implementerer den egentlige kode. Dette sikrer, at koden opfylder de specificerede krav og hjælper med at finde og rette fejl tidligt.
- **Brug af Moq-frameworket:** Dette framework anvender vi til at mocke (simulere) databaser og andre afhængigheder, så vi kan teste service-funktionalitet uden at skulle oprette en egentlig databaseforbindelse. Dette gør testene hurtigere og mere pålidelige.
- **Enhedstests (Unit Tests):** Vi anvender enhedstests til at verificere, at hver enkelt klasse og metode fungerer korrekt. Ved at isolere og teste små dele af systemet kan vi nemt identificere og rette fejl, hvilket resulterer i mere robust software.

Ved udviklingen af vores første service, InvoiceService, fulgte vi principperne for testdreven udvikling nøje. Vi startede med at oprette tests for de ønskede funktioner og derefter udvikle vores repository, InvoiceRepository, i overensstemmelse med interfacet IInvoiceRepository.

Ved at skrive tests før implementeringen kunne vi løbende verificere, at vores kode opførte sig som forventet, og derved undgå mange fremtidige fejl. Når der alligevel opstod problemer, selvom testsene var grønne (beståede), justerede vi testene for at inkludere de nye scenarier, der havde forårsaget fejlene. Dette gjorde det muligt at fange og rette mangler, som ikke tidligere var blevet opdaget, og dermed forbedre den overordnede kvalitet af systemet.

I vores projekt har vi valgt en Scrumban-tilgang, hvor vi kombinerer elementer fra både Scrum⁹ og Kanban. Hver morgen afholder vi en kort stand-up møde for at gennemgå, hvad hvert teammedlem har arbejdet på, hvad de planlægger at arbejde på, og om der er nogen hindringer. Vi planlægger vores arbejde i ugentligt sprint og holder retrospectives hver fredag for at evaluere vores fremskridt, diskutere udfordringer, og planlægge forbedringer.

Vi bruger Kanban-board til at visualisere og holde os konstant opdateret på arbejdsopgaver. Det hjælper os med at sikre, at arbejdet flyder jævnt gennem udviklingsprocessen.

Et vigtigt element i vores fremgangsmåde har været at sikre, at vores Kanban-board altid er opdateret, hvilket hjælper os med at holde styr på arbejdsopgaverne og deres status. Vi har også prioriteret teamets trivsel ved at inkludere energizers og pauser i vores daglige rutine.

I forhold til tidsestimering har vi valgt ikke at bruge dette værktøj, da det potentielt ville skabe mere arbejde end den værdi, det ville tilføre. Vi har fra starten været godt med og havde

⁸ Beck and Andres, *Extreme Programming Explained*.

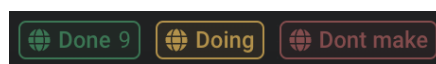
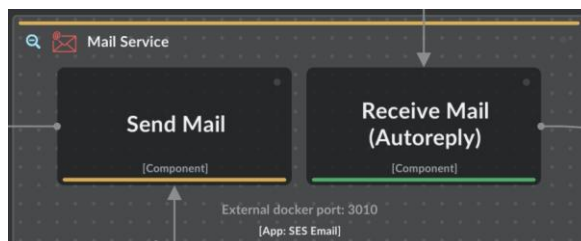
⁹ Kniberg, *Scrum and XP from the Trenches*.



planlagt processen gennem et internt roadmap. Dette roadmap har fungeret som en rettesnor for vores projektforsløb, hvilket har hjulpet os med at holde fokus og fremdrift.

Vi har ikke haft et dedikeret Scrum Master. I stedet har vi delt Scrum Master-opgaverne blandt teammedlemmerne, hvilket har fungeret godt, da vores team er relativt lille. Dette har givet os fleksibilitet og sikret, at alle i teamet har haft mulighed for at bidrage til projektets ledelse og koordinering.

I forbindelse med arkitekturmodellering har vi kontinuerligt opdateret vores arkitekturtegning for mikroservice-arkitekturen ved hjælp af IcePanel¹⁰. Ved brug af IcePanel har vi lavet en predefineret microservice IT-arkitektur, som vi har brugt som referencepunkt, hvilket har sikret, at alle teammedlemmer har haft en ensartet forståelse af systemets struktur og udviklingsproces. Derudover gjorde vi også brug af en indbygget funktion i IcePanel, som gjorde det muligt at lave egne "tags". Dette gjorde det muligt for os at lave en gruppe af tags, navngivet "Status", som kunne bruges til at tagge en service som værende "Done" eller "Doing". Da vi fandt ud af at ikke alle services, som vi startede ud med, skulle laves, lavede vi også "Dont Make".



Vores Continuous Integration¹¹ (CI) setup håndteres via GitHub Actions, hvilket har sikret en effektiv og automatiseret proces for kodeintegration og forhindrer, at ikke-bygbar kode bliver frigivet. Test-driven development (TDD) er blevet praktiseret og evalueret løbende for at sikre høj kvalitet i vores kodebase.

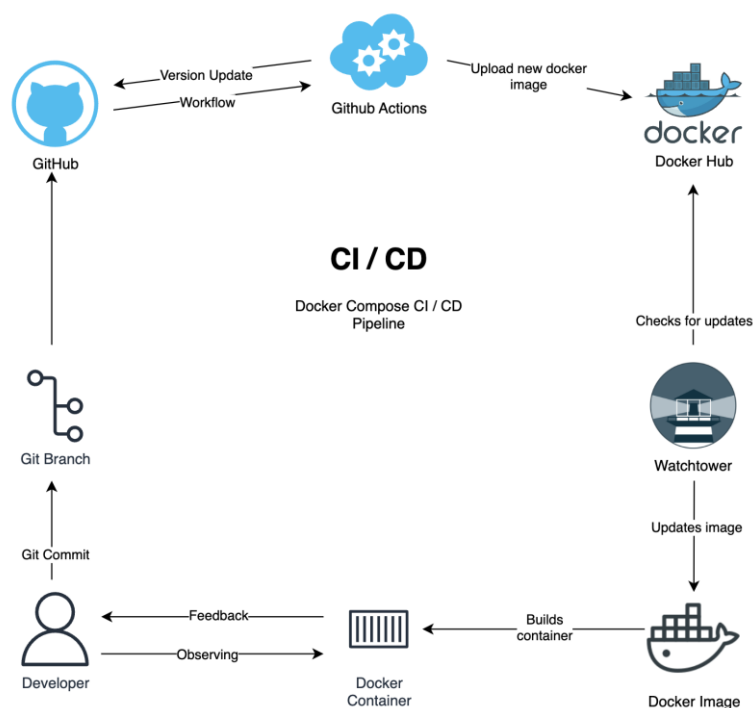
Dokumentation har været en integreret del af vores proces, hvor vi har skrevet og opdateret dokumentation i forbindelse med hver sprint retrospective. Dette har sikret, at vi har haft en ajourført og omfattende dokumentation af vores projekt.

Endelig har vi haft en struktureret tilgang til håndtering af arbejdsområdet og sørget for, at der er en ansvarlig for musik og lokale, hvilket har bidraget til et behageligt og produktivt arbejdsmiljø.

¹⁰ Se Beskrivelse af IT-Arkitektur og -infrastruktur

¹¹ Fowler, "Continuous Integration."





Vores CI/CD-pipeline, som vist i diagrammet ovenfor, er central for vores softwareudviklingsproces, hvor vi automatiserer integration (CI) og deployment (CD) for at sikre kontinuerlig levering af opdateringer og forbedringer til vores auktionsplatform. Her er en detaljeret gennemgang af pipeline-processen:

GitHub til Git Commit og Git Branch:

- Udviklerne starter med at lave ændringer i koden og committer disse ændringer til en specifik branch i vores GitHub-repository.
- Git-branches bruges til at isolere development og main, hvilket muliggør parallelt arbejde uden at påvirke hovedkoden.

GitHub Actions til CI/CD Workflow:

- Når et commit er lavet, trigges GitHub Actions, vores valg af CI/CD-platform, automatisk.
- GitHub Actions udfører en foruddefinerede workflow, herunder kørsel af byggescripts.
- Hvis alle tests er succesfulde, bygges en ny Docker-image af applikationen.

Docker Hub til Lagring af Docker Images:

- Den nybyggede Docker-image pushes automatisk til Docker Hub, vores centrale lager for Docker images.
- Dette sikrer, at den nyeste version af applikationen er tilgængelig for distribution og deployment.



Watchtower til Automatisk Opdatering (CD):

- Watchtower overvåger vores Docker-containere og Docker Hub for nye versioner af Docker images.
- Hvis et ny image er tilgængelig på Docker Hub, opdaterer Watchtower automatisk de relevante containere med den nyeste version.
- Dette understøtter vores kontinuerlige deployment (CD), hvor nye funktioner og rettelser hurtigt og automatisk rulles ud til vores produktionsmiljø.

Docker Container til Runtime Miljø:

- De opdaterede Docker-containere indeholder de nyeste ændringer og kører applikationen i produktionsmiljøet.
- Udviklere observerer applikationens performance og får feedback fra runtime-miljøet, som hjælper med at identificere og rette eventuelle problemer.

I løbet af udviklingen af vores auktionsplatform har vi brugt Moq-frameworket som en essentiel del af vores teststrategi. Moq er et mocking-bibliotek for .NET, der hjælper os med at oprette mock-objekter til enheder, der er svære at teste isoleret. Her er, hvordan vi har brugt Moq i vores udviklingsproces:

Under udviklingen af vores mikroservices, såsom InvoiceService, har vi brugt Moq til at mocke afhængigheder som repositories og eksterne services. Dette har gjort det muligt for os at isolere den funktionalitet, vi ønskede at teste, og sikre, at vores tests var fokuserede og ikke påvirkedes af eksterne faktorer.

Et konkret eksempel på brug af Moq i vores projekt er testmetoden GetAll_ReturnsAllInvoices. Denne metode tester, at GetAll-metoden i IInvoiceRepository korrekt returnerer alle fakturaer. Ved hjælp af Moq kan vi konfigurere en mock af IInvoiceRepository til at returnere en liste over fakturaer, når GetAll kaldes, og derefter verificere, at metoden opfører sig som forventet.

```
[TestMethod]
public void GetAll_ReturnsAllInvoices()
{
    var mockService = new Mock<IInvoiceRepository>();
    var invoices = new List<InvoiceModel>
    {
        new InvoiceModel { Description = "Test Invoice 1", Price = 100.0 },
        new InvoiceModel { Description = "Test Invoice 2", Price = 200.0 }
    };

    mockService.Setup(expression: s:IInvoiceRepository => s.GetAll()).Returns(invoices);

    var result:IEnumerable<InvoiceModel> = mockService.Object.GetAll();

    Assert.IsNotNull(result);
    Assert.AreEqual(expected: 2, actual: result.Count());
    mockService.Verify(expression: s:IInvoiceRepository => s.GetAll(), Times.Once);
}
```



I dette eksempel oprettes en mock af InvoiceRepository, og Setup-metoden bruges til at definere, at GetAll skal returnere en foruddefineret liste over fakturaer. Testen kalder derefter GetAll på mock-objektet og verificerer, at resultatet ikke er null, altså at det indeholder det forventede antal elementer, og at GetAll-metoden blev kaldt præcis én gang.

Fordele ved Brug af Moq:

- **Isolation af Testede Enheder:** Moq gør det muligt for os at isolere den enhed, der testes, fra dens afhængigheder. Dette sikrer, at testen kun fejler, hvis der er et problem med den enhed, der testes, og ikke på grund af afhængighedernes tilstand eller adfærd.
- **Forbedret Testdækning:** Ved at kunne simulere forskellige scenarier og edge cases kan vi opnå en højere testdækning og sikre, at vores kode håndterer forskellige situationer korrekt.
- **Øget Produktivitet:** Moq forenkler oprettelsen af mocks, hvilket reducerer den tid, det tager at skrive tests. Det forbedrer også vedligeholdelsen af testkoden ved at gøre den mere læsbar og mindre kompleks.
- **Tidlig Fejlfinding:** Ved at identificere potentielle fejl tidligt i udviklingscyklussen kan vi hurtigt rette dem og forhindre, at de når produktionen.

API Dokumentation & Test

For at sikre, at vores serviceplatforme er nemme at integrere med, har vi lagt stor vægt på grundig API-dokumentation. Formålet med vores API-dokumentation er at hjælpe udviklere med at forstå, hvordan de kan interagere med vores systemer og de tilgængelige endpoints.

Vores API-dokumentation starter med at beskrive de grundlæggende endpoints, som vores services er tilgængelige på. Basis-URL'en for vores api tjenester er <https://api.gronogolsen.dk>.

Et eksempel på et endpoint er det, der anvendes til at oprette en ny bruger. Dette endpoint, der anvender HTTP-metoden POST, er tilgængeligt på /user/create og er designet til at oprette en ny bruger i systemet. Dokumentationen beskriver, hvordan udviklere kan sende en POST-anmodning til dette endpoint og inkluderer detaljer om de forventede inputparametre og det output, der returneres, hvilket er en brugerkonto.

For at sikre, at vores API'er fungerer som forventet, har vi anvendt Postman til at teste API-kald. Postman har hjulpet os med at bekræfte, at vores API-kald gennemføres korrekt. I vores dokumentation inkluderer vi alle de tilgængelige endpoints i de services, vi har testet, samt en beskrivelse af deres funktionalitet og en guide til, hvordan disse API-kald kan testes i Postman. Vi specificerer de forskellige HTTP-metoder (GET, POST, PUT, DELETE) samt de nødvendige inputparametre. Dette gør det nemt for udviklere at forstå, hvordan de kan sende anmodninger til vores API og hvad de kan forvente som svar. Ved at inkludere beskrivelser af output sikrer vi, at dokumentationen er klar og brugervenlig.



I vores testdokumentation har vi disse endepunkter i bilag 13, 14, 15, og 16, som kan testes af. Derudover kan de også blive testet på vores testing hjemmeside på <https://gronogolsen.dk>

Fordele ved API-dokumentation

- **API-dokumentation giver udviklere klarhed:** De behøver ikke at gætte sig frem til, hvordan de skal bruge forskellige API-endpoints.
- **Det sparer tid:** Udviklere kan hurtigt forstå, hvordan API'en fungerer, hvad de skal sende, og hvad de kan forvente at få tilbage.
- **Mindre tid spildes på fejlfinding:** Med klar dokumentation behøver udviklere ikke at rode rundt og lede efter svar.

Ved at have lettilgængelig dokumentation kan udviklere bruge mere tid på at udvikle applikationer i stedet for at bruge tid på at finde ud af API'en.



Skalering

I vores overvejelser omkring projektstyring og teamstruktur har vi diskuteret, hvordan vores tilgang ville være, hvis vi var et team på 50 personer i virksomheden. I den forbindelse har vi valgt at gå med et koncept med et fullstack-align-team, som ville være passende til vores større skala.

Et fullstack-align-team tilbyder nemlig en række fordele, der ville have en afgørende betydning for effektiviteten af vores udviklingsproces i en større organisation. Ved at have både front-end og back-end udviklere i det samme team gør at vi optimerer kommunikationen og samarbejdet mellem teammedlemmerne, samtidig med at vi sikrer at der fuld kontrol over end-to-end funktionalitet i vores applikation.

Den kontakt vi får med slutbrugeren, som et fullstack team muliggør, ville give os en mulighed for at forstå og imødekomme brugernes behov og krav på en hurtig og effektiv måde. Dette er afgørende for at sikre, at vores løsninger er brugervenlige og opfylder de ønskede krav.

Derudover ville et fullstack-align-team ikke blot give os muligheden for at drage fordel af specialister inden for forskellige områder, men det ville også åbne op for en mere omfattende tilgang til udviklingsprocessen. Ved at integrere specialiserede færdigheder på tværs af alle lag af udviklingen, kan vi bedre håndtere kompleksiteten i vores systemer og microservice-arkitekturen, især når vi opererer i en større og mere dynamisk organisatorisk kontekst

Et fullstack-align-team sikre også, at vi kan arbejde uafhængigt og træffe beslutninger internt, uden behov for omfattende koordination med andre teams. Dette ville fremme hastighed og agilitet i vores udviklingsproces og sikre, at vores tilgang kunne tilpasses og fungere effektivt, uanset hvor stort et team kommer til at blive i fremtiden.

Desuden muliggør et sådant team en mere effektiv håndtering af eventuelle tekniske udfordringer, da vi har ekspertise både inden for front-end og back-end udvikling, hvilket resulterer i en mere sømløs og sammenhængende produktudvikling.

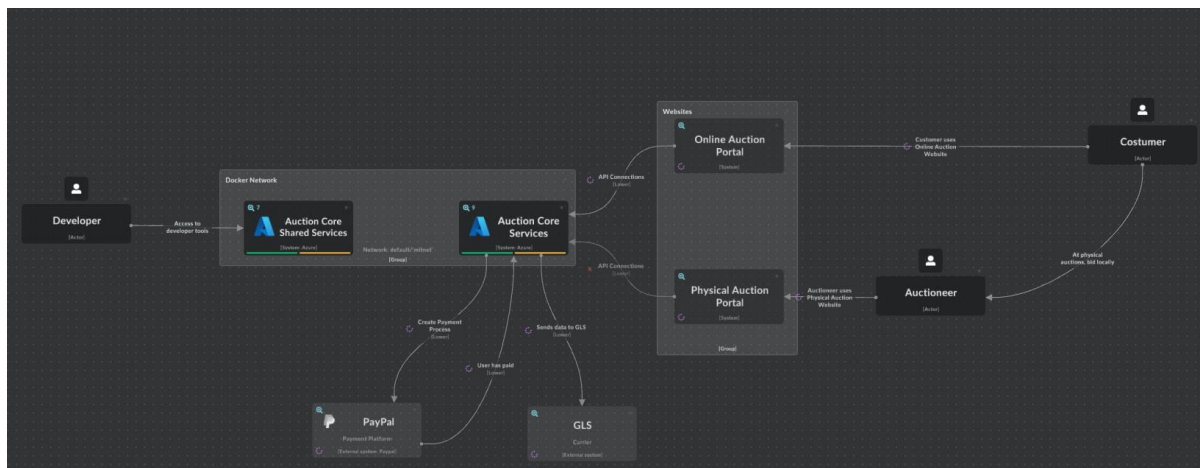
Alt i alt, hvis vi tænker over det, virker det som om et fullstack-align-team ville være det smarteste valg for vores projekt, hvis vi var en virksomhed på 50 mennesker. Det ville gøre det nemmere for os at håndtere al den kompleksitet og skabe gode løsninger, som alle kan bruge, på tværs af hele organisationen. Med alle de forskellige eksperter samlet og alle arbejdsopgaverne koordineret, ville vi være godt rustet til at tackle alt det, der måtte komme vores vej i fremtiden.



Beskrivelse af IT-Arkitektur og -infrastruktur

Auktionshusets IT-Arkitektur er tegnet ved brug af C4-modellen¹², den kan tilgås via <https://gronogolsen.dk>, eller direkte på en interaktiv onlineportal, IcePanel: <https://s.icepanel.io/mB4kr95xX1FRKO/RXg8>

- Eller alternativt i bilag 12

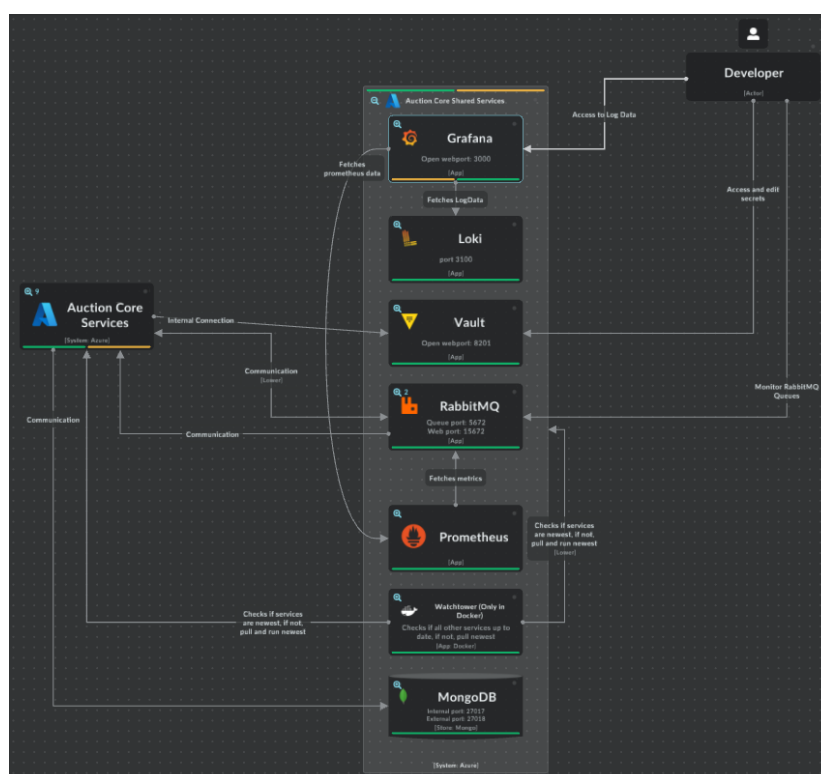


Overordnet har vi 2 servicepakker, Auction Core Shared Services og Auction Core Services¹³.

Auction Core Shared Services er en Container Stack indeholdende:

- Grafana
- Loki
- Vault
- RabbitMQ
- Prometheus
- Watchtower (i docker)
- MongoDB

Denne PoC leverer en funktionel pakke af anvendelige løsninger til vores Auction Core Services.



¹² "The C4 Model for Visualising Software Architecture."

¹³ Se TO-BE IT-Arkitektur i målbillede for nærmere forklaret Auction Core Services



I denne Proof of Concept (PoC) er valget af database faldet på MongoDB i en container for at undgå komplikationer forbundet med integration af ny og ukendt teknologi på de valgte platforme. Ved den endelige udvikling vil det være fordelagtigt at anvende en SQL-database og implementere nye repositories på de udviklede interfaces, da de gemte data i høj grad, er relationelle snarere end ikke-relationelle.

RabbitMQ er blevet anvendt for at sikre en robust messaging-løsning inden for et containeriseret miljø. For den endelige implementering på Azure vil det være fordelagtigt at migrere til Azure Message Queues for at drage nytte af den forbedrede opetid, kompatibilitet og skalerbarhed. Denne ændring kan desuden medføre, at Prometheus bliver overflødig.

Ligeledes er HashiCorp Vault valgt til sikker opbevaring og administration af hemmeligheder i PoC-fasen. Ved den endelige udvikling vil det være fordelagtigt at benytte Azure Key Vault for en mere integreret og omkostningseffektiv løsning, som også tilbyder forbedret opetid og kompatibilitet med Azure-platformen.

Endelig anvender vi Nginx som en omvendt proxy og til belastningsbalancering i det nuværende setup. Ved den endelige implementering ville det være fordelagtigt at overgå til Azure Application Gateway for at drage fordel af den dybere Azure-integration og de avancerede sikkerhedsfunktioner, som forbedrer opetid og skalerbarhed.

Load Balancing & Reverse Proxy

For at holde vores forskellige services adskilt og samtidig kunne servere dem gennem det samme front-end system, har vi implementeret Nginx som vores reverse proxy.

Eksternt anvender vi HTTPS for at sikre krypterede forbindelser mellem klienter og vores servere, hvilket beskytter data i transit mod aflytning og manipulation. Når først trafikken når vores eksterne Nginx-proxy, dekrypteres den HTTPS-forbindelse, og trafikken videreføres internt over HTTP til vores forskellige services. Dette setup reducerer kompleksiteten og overhead af kryptering internt i netværket, hvilket fører til bedre ydeevne samtidig med at opretholde sikkerhed eksternt.

Denne Nginx-konfiguration er designet med skalerbarhed for øje, hvilket gør det muligt både nemt og hurtigt at opskalerer vores nuværende løsning ved at tilføje flere instanser af den samme service. Hver instans specificeres i konfigurationsfilen, hvilket gør load balancing og high availability let og ukompliceret.

For eksempel kan vi, ved at tilføje flere upstream-servere under en specifik serviceblok i konfigurationen, distribuere indkomne anmodninger jævnt mellem flere service-instanser. Dette hjælper ikke kun med at balancere belastningen på vores systemer, men også med at øge fejltolerancen, hvis en instans skulle gå ned.

Desuden tillader Nginx fleksibilitet i håndteringen af forskellige typer af trafik og dataformater, hvilket yderligere understøtter en robust og skalerbar arkitektur. Med Nginx som vores proxy-løsning opnår vi således en høj grad af modularitet, som er essentiel for en moderne, mikroservicebaseret arkitektur.



Monitorering med Loki, Grafana og Prometheus

Ud over vores effektive Nginx-baserede load balancing og reverse proxy-løsning, har vi forbedret vores system monitorering ved at implementere Loki og Grafana til at overvåge vores Auction Core Services. Loki er en meget skalerbar, letvægts log aggregator, der er designet til at arbejde i tæt samarbejde med Grafana. Dette setup gør det muligt for os at samle og visualisere logdata, hvilket hjælper os med hurtigt at identificere og løse potentielle problemer i vores services.

For at indsamle og rapportere data fra RabbitMQ ind i Grafana, anvender vi Prometheus. Prometheus er en metrics samler, der hurtigt og effektivt kan hente data fra forskellige kilder og eksponere det i et format, der er let at forstå. Ved at kombinere Prometheus med Grafana, kan vi skabe detaljerede dashboards, der giver os dyb indsigt i systemets ydeevne og tilstand.

Denne kombination af værktøjer sikrer, at vi har en omfattende og realtids orienteret monitorerings løsning, der holder os informeret om alle aspekter af vores Auction Core Services. Med en sådan løsning kan vi ikke kun reagere på hændelser hurtigere, men også proaktivt identificere og adressere potentielle problemer, inden de påvirker slutbrugerne.

Grafana kan ses på: <https://grafana.gronogolsen.dk>

Med credentials:

Brugernavn: eksamen

Password: eksamen

Sikkerhed med HashiCorp Vault

For at sikre håndteringen af sensitive data på tværs af vores tjenester, har vi implementeret HashiCorp Vault som vores centraliserede løsning til sikkerhedshåndtering. HashiCorp Vault giver os mulighed for at lagre, administrere og kontrollere adgang til secrets såsom API-nøgler og adgangskoder fra et enkelt centralt system.

Vi gemmer også vores JWT (JSON Web Token) secrets i denne Vault. Ved at anvende et centralt system til håndtering af secrets kan vi nemt rotere nøgler og adgangskoder, hvilket reducerer risikoen for kompromittering af vores sikkerhed. Vault gør det muligt for os at beskytte vores data ved at kryptere secrets både i transit og i hvile, hvilket sikrer, at kun autoriserede applikationer og brugere har adgang til dem.

Integrationen af HashiCorp Vault i vores arkitektur betyder, at vi kan opnå en højere grad af sikkerhed, automatisering og compliance, hvilket er afgørende for at beskytte følsomme oplysninger og sikre integriteten af vores systemer. Denne centraliserede tilgang til secret management gør vores infrastruktur sikkerhed mere robust og nemmere at administrere i takt med, at vores services skalerer.

Med HashiCorp Vault sikrer vi således en høj standard for sikkerhed og compliance, som er essentiel for vores moderne, microservicebaserede arkitektur.



RabbitMQ Integration

For at sikre essentielle beskeder mellem systemerne benytter vi også RabbitMQ som vores primære messaging queue system. RabbitMQ gør det muligt for os at håndtere og transmittere beskeder og data mellem de forskellige microservices på en pålidelig og effektiv måde.

RabbitMQ er essentielt for opdatering af budstatus i realtid og sikrer, at notifications og andre vigtige beskeder kommer frem til de korrekte services. Det fungerer som et mellemlid, der gemmer og sender beskeder, hvilket hjælper med at afkoble produktionen og forbrug af beskeder mellem systemkomponenter, hvilket igen forbedrer systemets robusthed og skalerbarhed.

RabbitMQ integrationen i vores arkitektur hjælper med at sikre fleksibilitet og pålidelig kommunikation mellem services som BiddingService, LotService og MailService. Ved at bruge RabbitMQ kan vi opnå høj ydeevne og minimere mængden af data tab i tilfælde af fejl, hvilket er kritisk for en gnidningsløs operation af vores auktionsplatforme.

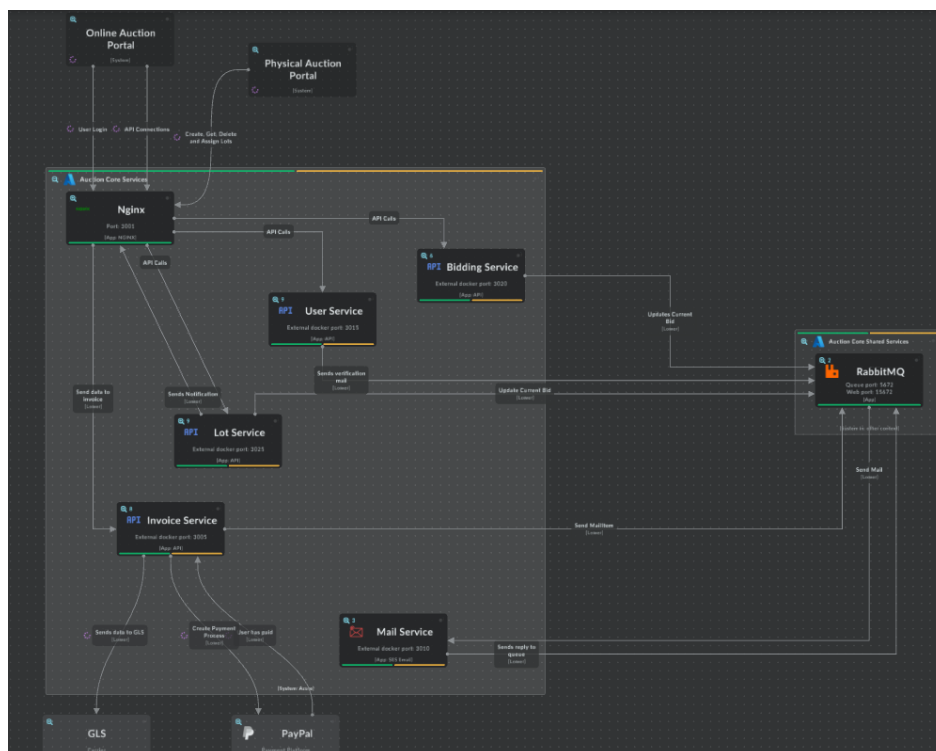
Continuous Deployment med Watchtower

For at sikre en automatiseret og kontinuerlig deployment af vores tjenester, bruger vi Watchtower i vores Docker-baserede løsning. Watchtower overvåger vores Docker-containerer og opdaterer dem automatisk, når nye versioner af billederne er tilgængelige. Dette fremmer en problemfri kontinuerlig deployment (CD) ved hele tiden at holde vores miljø opdateret med de nyeste ændringer uden manuel intervention.

Brugen af Watchtower i vores Docker-setup garanterer, at vores images altid er opdateret med vores nyeste kode og konfigurationer, hvilket minimerer nedetid og understøtter vores CI/CD pipeline.



Auction Core Services



Vores Auction Core Services indeholder flere integrationer, der tilsammen sikrer en effektiv og robust drift af vores auktionsplatforme. Diagrammet herover viser en række essentielle integrationer mellem de forskellige services, som samlet set understøtter de forskellige funktioner på tværs af platformen.

User Service

Denne service håndterer brugeradministration, herunder login, verificering og brugeroplysninger. Den kommunikerer med både Online Auction Portal og Physical Auction Portal for at opdatere brugernes informationer og status.

Lot Service

Lot Service håndterer alle opgaver relateret til auktions lots, fra oprettelse og tildeling af lots til at sende notifikationer vedrørende bydning på lots. Den kommunikerer primært med User Service og Invoice Service for at sikre en sammenhængende oplevelse på tværs af brugerinteraktion og fakturering.

Bidding Service

Denne service er ansvarlig for håndteringen af bud på de forskellige partier. Den modtager bud fra brugerne og sørger for at opdatere den aktuelle budstatus. Bidding Service opdaterer også RabbitMQ for at sikre, at alle bid-opdateringer er transmitteret korrekt til de relevante tjenester.



Invoice Service

Invoice Service står for fakturering og sender de nødvendige data til betalings- og logistiksystemer som GLS og PayPal. Når en bruger har betalt, bliver information om betalingen opdateret i systemet.

Mail Service

Mail Service sørger for at sende diverse notifikationer og bekræftelser via e-mail til brugerne. Den modtager send anmodninger fra tjenester som Lot Service og Invoice Service, og sørger for levering af e-mails.

Denne omfattende integration af Auction Core Services sikrer, at alle dele af systemet fungerer gnidningsløst sammen, hvilket giver en høj grad af driftssikkerhed og brugervenlighed i vores auktionsportaler.

Authorization & Authentication på services

Vi benytter JWT Token Authorization for essentielle API-kald for at sikre, at kun autoriserede brugere kan udføre kritiske handlinger som at placere bud og gennemføre betalinger, imens mere sensitive kald såsom at opdatere brugeroplysninger er begrænset mere ved at sammenligne sendte JWT token med hvilken brugers opdateringer der bliver opdateret. Dermed sikrer vi at det kun er den specifikke bruger, der kan handle på den specifikke brugers oplysninger. For mere almene kald, som f.eks. at hente information om auktions-lots, har vi åbne API-endepunkter, der ikke kræver autorisation. Dette reducerer byrden på systemet og forbedrer brugeroplevelsen, idet almindelige forespørgsler kan udføres hurtigt og nemt.

Derudover anvender vi en intern authentication-mekanisme, som gør det muligt for vores services at kommunikere indbyrdes uden at skulle udstede og verificere JWT tokens. Denne interne kommunikation sker gennem en instanceret WebManager, som bruger en HttpClient med en speciel header kaldet X-Internal-ApiKey.

Denne header er initialiseret med en secret value, som hentes fra Vault. Når en service skal kalde en anden intern service, tilføjer WebManager denne header med den korrekte secret value. Den modtagende service kontrollerer headeren og sammenligner den med sin egen kopierede secret value fra Vault. Hvis de matcher, bypasses behovet for JWT tokens, og kaldet får lov at eksekvere.

Denne tilgang sikrer både høj sikkerhed og effektivitet, da vi reducerer overheadet ved token-håndtering internt i systemet, samtidig med at vi opretholder sikre og autoriserede forbindelser mellem vores services.



Mail Service: Integrationen og Muligheder

Vores Mail Service er designet til at håndtere udsendelse af e-mails såsom brugeroprettelse, fakturaer og systemnotifikationer til brugerne. På nuværende tidspunkt anvender vi Gmail's SMTP-servere til at sende disse e-mails. Denne løsning er praktisk og nem at konfigurere, hvilket muliggør hurtig og pålidelig e-mailkommunikation med vores brugere.

Selvom Gmail er vores nuværende valg, giver vores system også mulighed for at opsætte en egen SMTP-server. Dette kan være en fordel, hvis man ønsker at sende e-mails fra virksomhedens eget domæne, såsom gronogolsen.dk, hvilket vil styrke Grøn og Olsens branding og professionelle image. En egen SMTP-server kan konfigureres til at fungere med vores Mail Service ved at justere SMTP-indstillingerne og sikre, at den nødvendige sikkerhed og autentifikation er på plads.

Alternativt er det også muligt at integrere/tilkøbe Google Workspace hvis der ønsker at bruge Googles effektive infrastrukturen, men samtidig sende e-mails fra deres eget domæne. Dette vil inkludere fordele som høj pålidelighed, forbedret branding og avancerede administrative funktioner.

Denne fleksibilitet i vores Mail Service sikrer, at vi kan tilpasse os forskellige behov og præferencer, alt efter hvad der passer bedst til vores brugere og virksomhedernes krav.

Interoperabilitet

Auction Core Services sikrer fuld interoperabilitet i overensstemmelse med kravene i "Kontrolsystem af Online Auktionshuse" gennem en række nøgleservices:

- Nginx
 - Anvendes til præcis fordeling af forespørgsler på korrekte endepunkter og services.
- UserService
 - Tilbyder følgende endpoints:
 - /api/legal/login til autentificering af brugere.
 - /api/legal/users/{userid} til hentning af brugerinformation.
- LotService
 - Ansvarlig for interoperabiliteten ved auktionskald og administrerer kommunikation mellem forskellige auktionskomponenter.
 - Tilbyder følgende endpoints:
 - /api/legal/auction/{auctionid} til hentning af specifik auktionsinformation.
 - /api/legal/auctions til at liste alle auktioner.
 - Inkluderer en AuctionMapper, som omstrukturerer Lots til Auctions i overensstemmelse med specifikationens krav.

Denne strukturerede tilgang til IT-arkitektur og infrastruktur sikrer, at alle komponenter fungerer sømløst sammen og opfylder de nødvendige standarder for en effektiv online auktionsplatform.



API Dokumentation

Alle vores API kald er dokumenteret via OpenAPI, alle disse kan ses på github på:

<https://github.com/ThySeven/AuctionCoreServices/tree/main/OpenAPI>

Eller på

<https://gronogolsen.dk>, under "OpenAPI List"

Disse er opdelt i services:

- BiddingService.json
- InvoiceService.json
- LotService.json
- UserService.json

Alle endepunkter kan testes jf. Testdokumenter der ligger i bilag 13, 14, 15 og 16, eller alternativt kan de testes med fordel på <https://gronogolsen.dk> under:

- UserTests
- LotTests
 - Inklusiv BiddingTests
- InvoiceTests
- InteroperabilityTests

Model Dokumentation

Alle vores anvendte skemaer i systemet kan findes på vores GitHub-side:

<https://github.com/ThySeven/AuctionCoreServices/tree/main/Schema>

De er også tilgængelige på vores testside:

<https://gronogolsen.dk>, under "Schema List"

Deployment

Auction Core Shared Services og Auction Core Services kan let køres ved hjælp af Docker Compose. Vejledningen til dette findes på GitHub:

<https://github.com/ThySeven/AuctionCoreServices>.

Efter opstarten af Auction Core Shared Services i Docker, er det afgørende at initialisere secrets ved hjælp af en VaultInitializer i vores Proof of Concept (PoC) setup. Disse secrets er essentielle for den korrekte funktionalitet af Auction Core Services. VaultInitializer køres automatisk ved brug af startscriptet. Alternativt er rækkefølgen for deployment nøje dokumenteret på GitHub.

Versionering af microservices håndteres ved at oprette en VERSION-fil i kildekoden. Denne fil læses under CI-processen, hvorefter versionen deployes, incrementeres og gemmes.

Med nogle mindre justeringer i Docker Compose-filerne kan disse omdannes til en Docker Swarm Service Stack. Dette gør det muligt at implementere dynamisk og automatisk skalering



af services, afhængig af belastning og eventuelle servicefejl, hvilket forbedrer systemets robusthed og fleksibilitet.

Selvom Azure Bicep forenkler deployment af Azure-ressourcer, har vi i vores nuværende projekt oplevet flere udfordringer under selve deployment-processen. En af de mest fremtrædende problematikker har været kompleks fejlfinding. Vi har ofte stødt på fejlmeddelelser, der var vage eller ikke direkte pegede på årsagen til problemet, hvilket har krævet en dybere undersøgelse af Bicep-koden for at identificere og løse problemerne. Fejlene har typisk været relateret til syntax, uoverensstemmelser i afhængigheder eller problemer med ressourcer, der blev forsøgt oprettet i forkert rækkefølge.

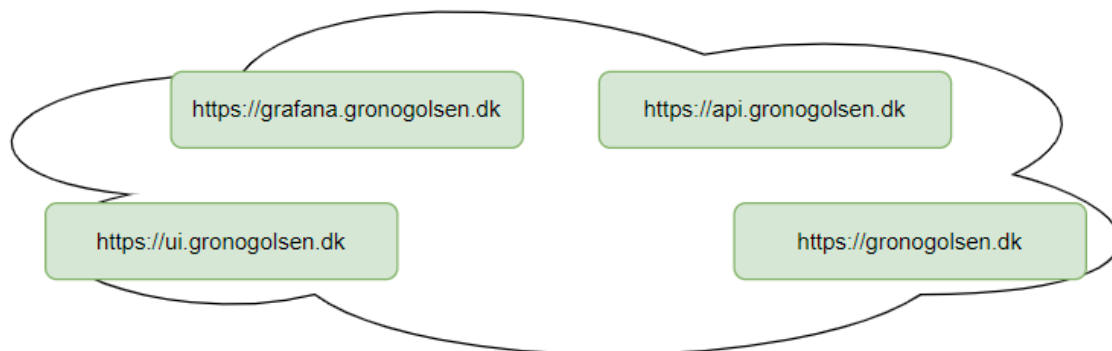
For at afhjælpe dette har vi anvendt Visual Studio Code med Bicep-udvidelsen, som tilbyder syntaksfremhævnning og fejlmarkering. Dette har hjulpet med at reducere fejlfindingstiden. Afhængighedsproblemer har også kompliceret vores deployment-processer, især når afhængigheder mellem ressourcer ikke var korrekt defineret, eller ressourcer ikke blev oprettet i den forventede rækkefølge. Dette har ført til deployment-fejl og inkonsistente miljøer. For at løse dette har vi nøje testet vores Bicep-filer i mindre, isolerede miljøer, før de blev anvendt i produktionsmiljøer. Denne fremgangsmåde har hjulpet os med at identificere og rette afhængighedsproblemer i løbet af udviklingsprocessen.

Som en backup-løsning har vi implementeret vores deployment-løsninger på en alternativ platform ved hjælp af Docker. Containerisering med Docker har gjort det muligt for os at opsætte konsekvente og isolerede miljøer, hvilket har reduceret risikoen for miljøspecifikke fejl. Ved at containerisere applikationer og deres afhængigheder har vi sikret, at applikationen kører ensartet uanset den underliggende infrastruktur. Denne tilgang har været særligt nyttig, når vi har stødt på problemer med Bicep-deployments, da det har givet os en robust fallback-metode til at håndtere deployment-kompleksiteter.

I sidste ende har kombinationen af Bicep og Docker-containerisering gjort det muligt for os at opnå en mere fleksibel og pålidelig deployment-proces i vores projekt. Dette har ikke kun forbedret vores evne til at fejlsøge og rette problemer, men har også styrket vores overordnede deployment-strategi.



Access Points



Følgende endepunkter med grænseflade kan tilgås på:

<https://gronogolsen.dk>

- Dokumentation & test af API-endepunkter

<https://ui.gronogolsen.dk>

- Sketch af brugergrænseflade

<https://grafana.gronogolsen.dk>

- DevOps monitoreringsværktøj af Auction Core Services

<https://api.gronogolsen.dk>

- API Endpoint til Auction Core Services

Kildekode Repositories

Alle kildekoder til vores services kan findes på GitHub via nedenstående links:

- AuctionCoreServices: <https://github.com/ThySeven/AuctionCoreServices>
- InvoiceService: <https://github.com/ThySeven/InvoiceService>
- UserService: <https://github.com/ThySeven/UserService>
- LotService: <https://github.com/ThySeven/LotService>
- BiddingService: <https://github.com/ThySeven/BiddingService>
- MailService: <https://github.com/ThySeven/MailService>
- VaultInitializer: <https://github.com/ThySeven/VaultInitializer>



Afrunding

Grøn og Olsens forestående digitale transformation sigter mod at implementere en skalerbar, fleksibel og sikker IT-arkitektur, der understøtter både traditionelle og online auktioner. Gennem nøje planlagte arkitekturprincipper, integrationer og moderne teknologier som microservices, cloud-løsninger og robust sikkerhed vil auktionshuset være godt rustet til fremtidens udfordringer. Ved at følge den detaljerede roadmap og inddrage brugernes feedback, stræber virksomheden efter at opnå en seamless og tilfredsstillende brugeroplevelse, samtidig med at de bevarer deres kerneværdier og konkurrencedygtighed i markedet. Dette grundlag giver Grøn og Olsen Auktionshus mulighed for at fortsætte sin succes og relevans, både på kort og lang sigt.

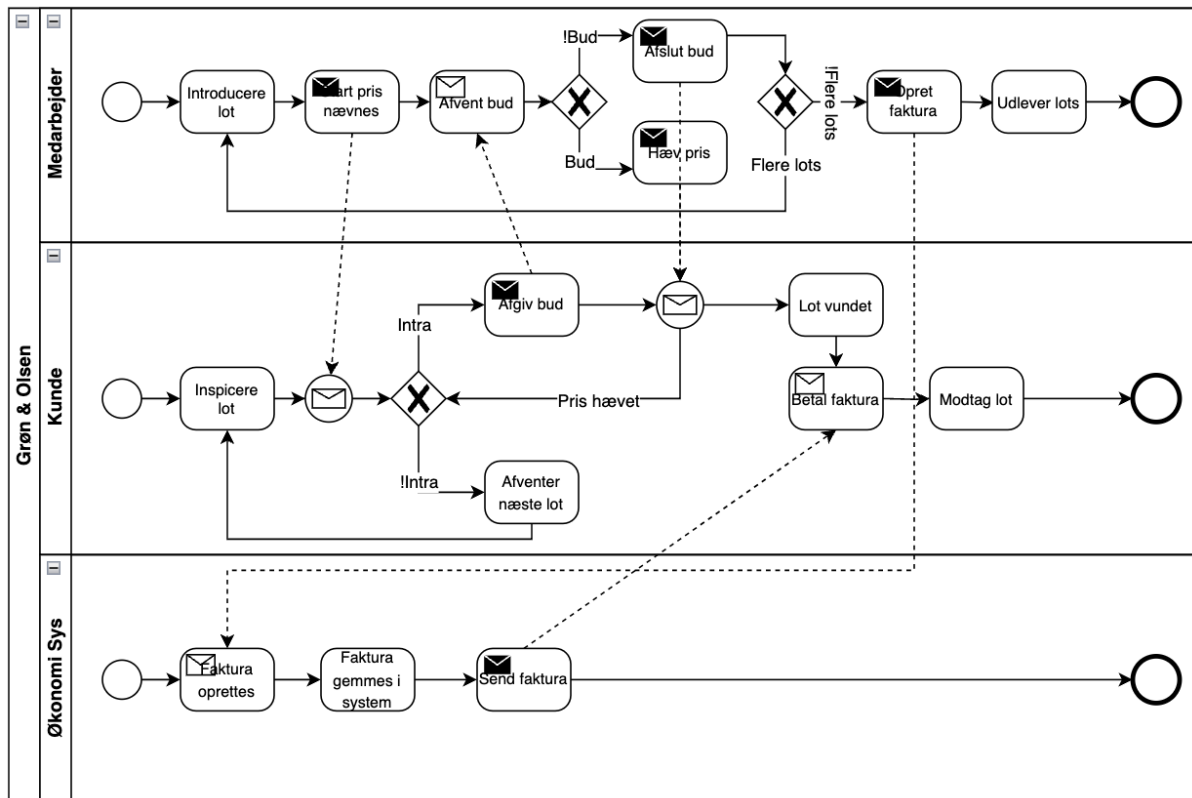


Referencer

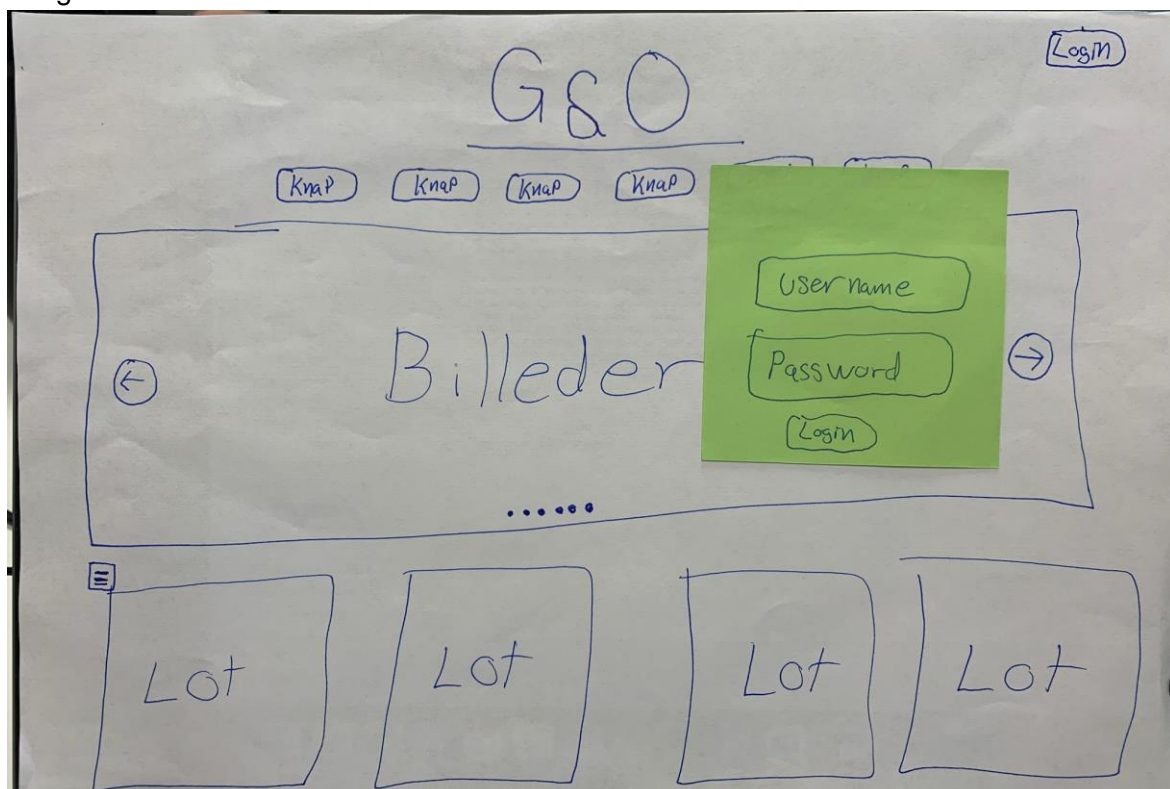
- Newman, Sam. *Building Microservices*. Second. O'Reilly Media, Inc., 2022.
- Ross, Jeanne W., Peter Weill, and David Robertson. *Enterprise Architecture as Strategy: Creating a Foundation for Business Execution*. Nachdr. Boston, Mass: Harvard Business School Press, 20.
- Beck, Kent, and Cynthia Andres. *Extreme Programming Explained: Embrace Change*. 2nd ed. Boston, MA: Addison-Wesley, 2005.
- Davis, Jenny L. *How Artifacts Afford: The Power and Politics of Everyday Things*. Design Thinking, Design Theory. Cambridge, Massachusetts: The MIT Press, 2020.
- Fowler, Martin. "Continuous Integration," January 18, 2024.
<https://martinfowler.com/articles/continuousIntegration.html>.
- Kniberg, Henrik. *Scrum and XP from the Trenches: How We Do Scrum*. Malmö: MTM, 2024.
- "The C4 Model for Visualising Software Architecture," n.d. <https://c4model.com/>.



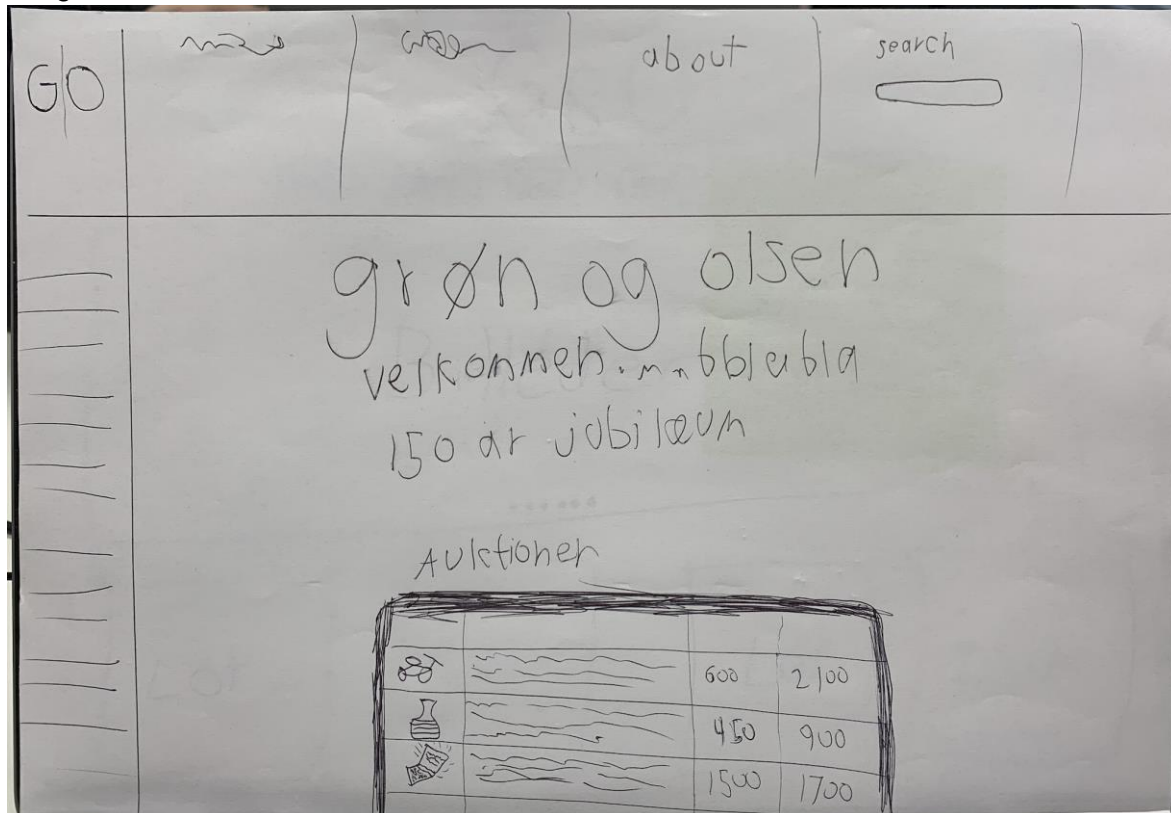
Bilag 2: BPMN, AS-IS



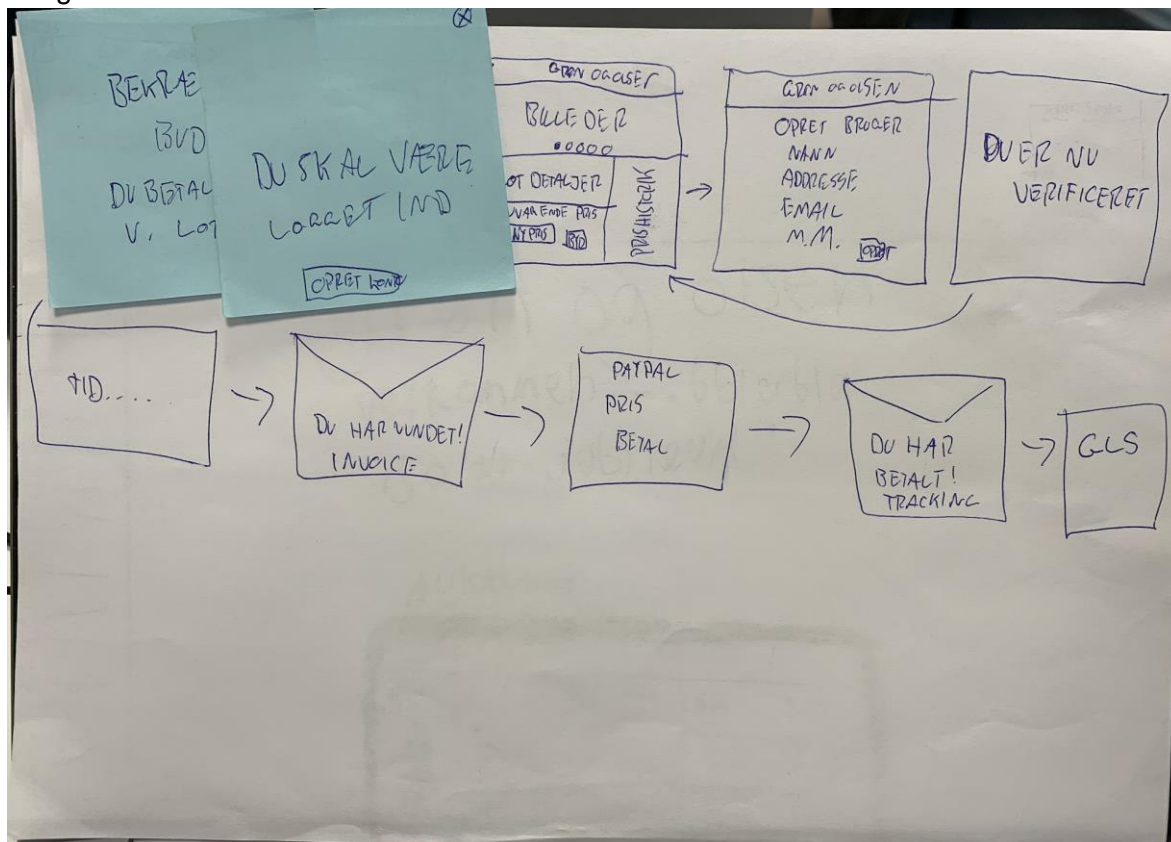
Bilag 3: Skitse



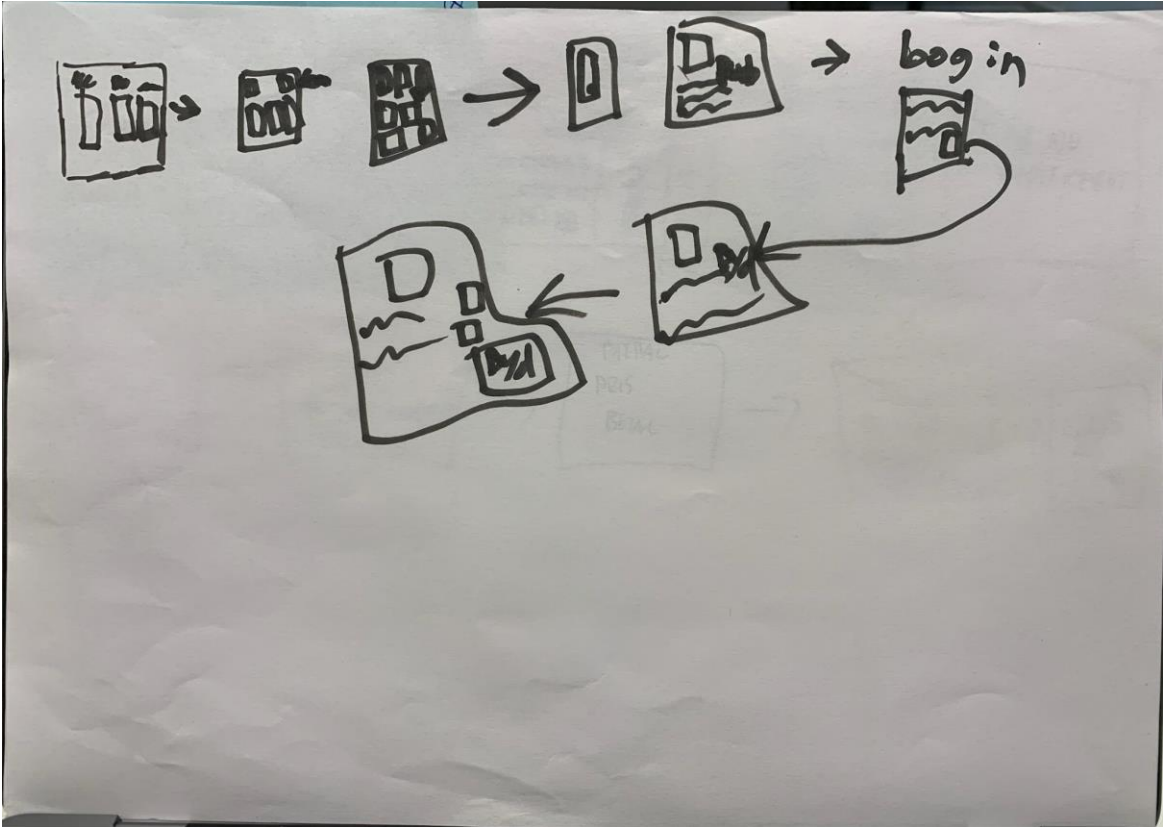
Bilag 4: Skitse



Bilag 5: Skitse



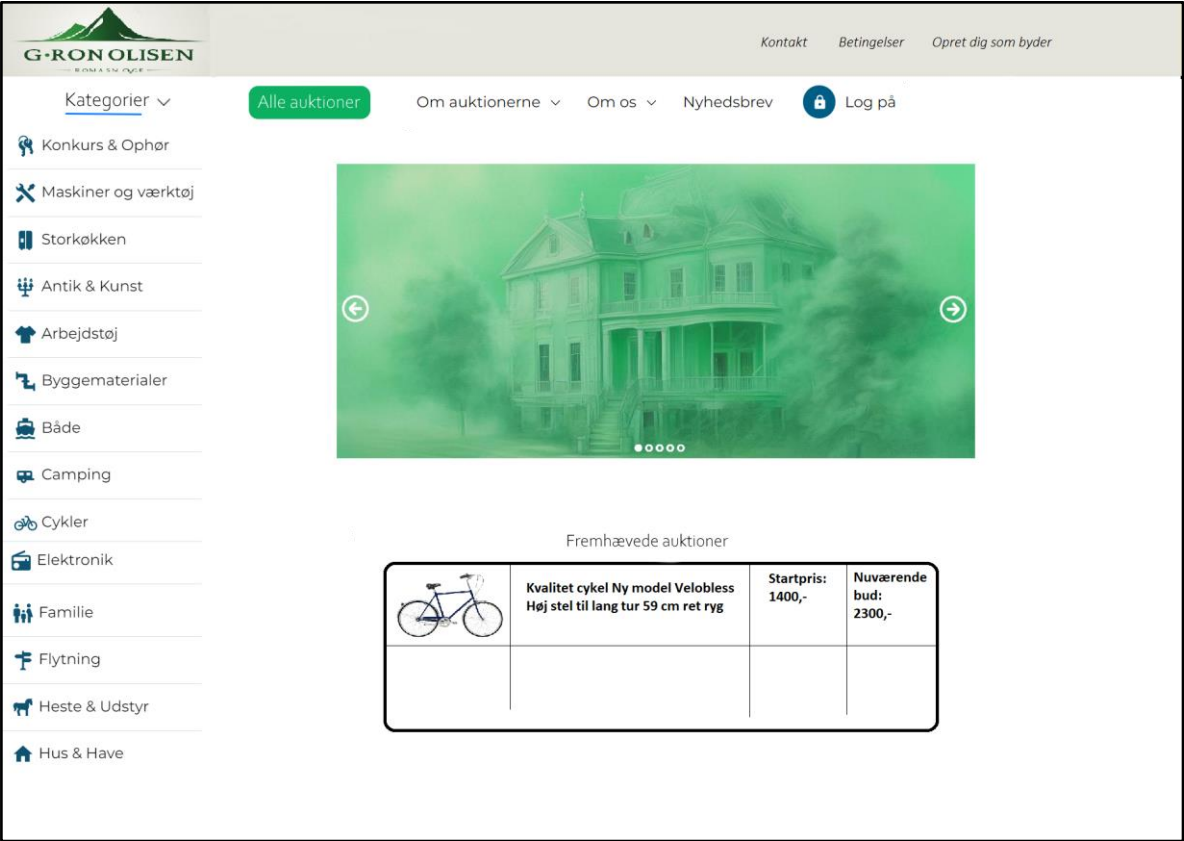
Bilag 6: Skitse



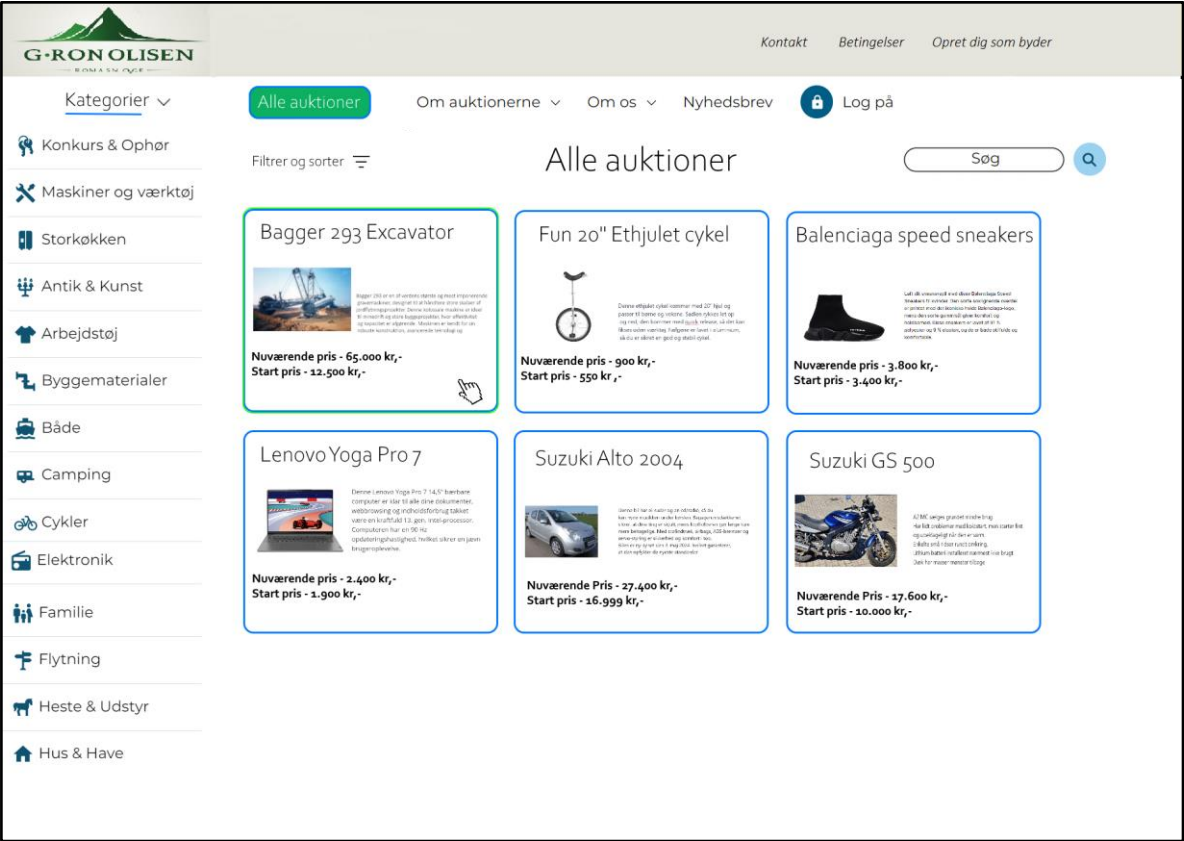
Bilag 7: Skitse

The sketch shows a web page layout. At the top left is a logo. To its right is a navigation menu with 'Aktionshuset', 'login', and 'Min Profil'. The main content area is divided into four columns: 'Kategori', 'filter', '* Lotliste', and '** Lotliste'. The 'Kategori' column has a list of items. The 'filter' column has a 'Selot' button. The '* Lotliste' column has a 'x lot' button. The '** Lotliste' column has a list of items. The right sidebar contains a 'Kontakt' section with links to 'Information', 'Introduktion', 'Veiledning', and 'QA'.

Bilag 8: Skitse



Bilag 9: Skitse



Bilag 10: Skitse

The screenshot shows the G-RON OLSEN auction website. The header includes the logo and navigation links: Kontakt, Betingelser, and Opret dig som byder. A sidebar on the left lists various categories like Konkurs & Ophør, Maskiner og værktøj, Storkøkken, etc. The main content area displays a listing for a 'Bagger 293 Excavator'. It includes three images of the excavator, its current price (61,000 DKK), start price (12,500 DKK), minimum price (25,000 DKK), start and end times of the auction, and the location (Tyskland, Stuttgart). A green button 'Afgiv bud' is visible. Below the listing, a table shows the bidding history.

Bud ID	Bud	Bud tidspunkt
200208	61.000 DKK	06:24:47 - 21.05.2024
200207	61.000 DKK	06:24:46 - 21.05.2024
200206	60.000 DKK	06:24:27 - 21.05.2024
200205	59.000 DKK	06:24:27 - 21.05.2024
200204	58.000 DKK	06:24:27 - 21.05.2024

Bilag 11: Skitse

Bilag 12: ThySevenGronOgOlsenAuctionCoreServices.pdf

- <https://github.com/ThySeven/Aflevering>

Bilag 13: "UserTests/UserService Test Doc.pdf"

- <https://github.com/ThySeven/Aflevering>

Bilag 14: "UserTests/LotService Test Doc.pdf"

- <https://github.com/ThySeven/Aflevering>

Bilag 15: "UserTests/InvoiceService Test Doc.pdf"

- <https://github.com/ThySeven/Aflevering>

Bilag 16: "UserTests/BiddingService Test Doc.pdf"

- <https://github.com/ThySeven/Aflevering>

