*Lukas Orluff Knudsen*

[*eaaluok@students.eaaa.dk*](mailto:eaaluok@students.eaaa.dk) *| EAALUOK*

*Systemintegration Forår 2023*

EKSAMENSPROJEKT – DMU-21T

Indholdsfortegnelse

[Opgave 1 - Integration af forretningsprocesser (15 point) 2](#_Toc134652767)

[Registrering af tilstedeværelse ved hjælp af sit nye studiekort 2](#_Toc134652768)

[Opgave 2 - Service integrations arkitektur (15 point) 3](#_Toc134652769)

[Event Description Table 3](#_Toc134652770)

[Service Category Table 3](#_Toc134652771)

[Opgave 3 - Informations integrations arkitektur (10 point) 4](#_Toc134652772)

[Metadata Model – C-READ 4](#_Toc134652773)

[Metadata Model – PAYS 4](#_Toc134652774)

[Opgave 4 - Enterprise integrationsmønstre (10 point) 5](#_Toc134652775)

[Generelt 5](#_Toc134652776)

[C-READ 5](#_Toc134652777)

[PAYS 6](#_Toc134652778)

[Opgave 5 - Enterprise integrationsmønstre (30 point) 7](#_Toc134652779)

[Opgave 6 - Enterprise Integration Patterns (20 point) 8](#_Toc134652780)

[Generelt 8](#_Toc134652781)

[Kode 8](#_Toc134652782)

[Router 8](#_Toc134652783)

[Simuleret sender 9](#_Toc134652784)

[XML Datafil 10](#_Toc134652785)

[Output 11](#_Toc134652786)

[STAS 11](#_Toc134652787)

[SS 11](#_Toc134652788)

[LMS 11](#_Toc134652789)

[PAYS 12](#_Toc134652790)

[Out 12](#_Toc134652791)

[RabbitMQ Management 13](#_Toc134652792)

[Overview 13](#_Toc134652793)

[Eksempel 13](#_Toc134652794)

# Opgave 1 - Integration af forretningsprocesser (15 point)

## Registrering af tilstedeværelse ved hjælp af sit nye studiekort

![A picture containing text, diagram, screenshot, number

Description automatically generated]()

Figur 1: Aktivitetsdiagram af en C-READ forretningsproces

Figur 1 beskriver et forløb over registrering af tilstedeværelse, ved brug af det nye kortlæsningssystem (C-READ). Der antages at der er installeret en kortlæsningsterminal i hvert klasselokale. Dette øger installationsomkostninger, men forhindre misbrug af systemet. Det er også antaget at C-READ indeholder CPR, navn og kortnummer for hver studerende, derudover ved C-READ hvilket klasselokale kortlæsningsterminalen er installeret i.

Flowet følger en iterativ proces. Dette betyder at C-READ kun anmoder om data fra de andre systemer, og har derfor ALT ansvaret for processen. Dette kaldes høj samhørighed og lav kobling, da hvert system står for udførslen af KUN deres egne opgaver. Forløbet ender med en fejl- eller succesmeddelelse.

# Opgave 2 - Service integrations arkitektur (15 point)

## Event Description Table

|  |  |  |  |
| --- | --- | --- | --- |
| # | Business event | Description | Responses |
| 1 | Registrering af tilstedeværelse ved hjælp af sit nye studiekort | Studerende registrerer sin tilstedeværelse ved at scanne sit studenterkort på kortlæsningsterminalen i klasseværelset | 1. Verificere studenterkort 2. Verificere at den studerende er aktiv 3. Verificere at den studerende er i det korrekte klasseværelse 4. Registrerer tilstedeværelse i LMS 5. Bekræftelse af registrering |
| 2 | Køb af mad i kantinen ved at betale med sit nye studiekort | Studerende betaler ved at scanne sit studenterkort på kortlæsningsterminalen i kantinen | 1. C-READ verificerer studenterkortet 2. Tjekker om der er tilstrækkelige saldo på studenterkortet 3. Fratrækker mængde fra saldo 4. Bekræftelse af betaling |

Figur 2: Event Description Table af et par events bade i C-READ og PAYS

## Service Category Table

|  |  |  |  |
| --- | --- | --- | --- |
| Response | Description | Category | Systems |
| 1.1 Verificere studenterkort | Tjekke at studenterkortet er tilknyttet en studerende | Studenterkort | C-READ |
| 1.2 Verificere at den studerende er aktiv | Tjekke at studerende er studieaktiv | Studenterkonti | STAS |
| 1.3 Verificere at den studerende er i det korrekte klasseværelse | Tjekke om der eksisterer en lektion for den givne klasse, i det givne klasseværelse, på den givne dato og tidspunkt | Skemastyrring | SS |
| 1.4 Registrerer tilstedeværelse i LMS | Send anmodning om at registrere den givne studerende som tilstede | Tilstedeværelse | LMS |
| 1.5 Bekræftelse af registrering | Meld tilbage til studerende at de er blevet registreret som tilstede | Tilstedeværelse | C-READ |

Figur 3: Service Category Table af den tidligere beskrevet event I C-READ

# Opgave 3 - Informations integrations arkitektur (10 point)

## Metadata Model – C-READ

|  |  |  |  |
| --- | --- | --- | --- |
|  | Data element name | <StudenterID> | <Studenterkortnummer> |
| Basic metadata | Data source | <IDM udtrækker løbende data fra CPR-registret. Studenter ID er direkte relateret til den studerendes CPR> | <C-READ autogenerere ét unikt studenterkortnummer for hver student> |
| Description | <Studenter ID bruges som den studerendes unikke ID. Dette er direkte relateret til den studerendes CPR. Studenter ID’et bruges til at tilknytte ét studenterkortnummer til den korrekte student> | <Studenterkortnummer bruges til at tilknytte ét studenterkort til den korrekt student. Studenterkortnummer identificere et studenterkort> |
| Format and data type | <DDMMYYXXXX – String> | <XXXXYYYYZZZZWWWW – String> |
| Canonical name | <Studenter ID> | <Studenterkortnummer> |
| Canonical format | <DDMMYYXXXX> | <XXXXYYYYZZZZWWWW> |
| Transformation rules | <DDMMYY-XXXX til DDMMYYXXXX> | <XXXX-YYYY-ZZZZ-WWWW til XXXXYYYYZZZZWWWW> |
| Interface | <Web service, API, adapter, MQ, SQL> | <Adapter, MQ, SQL> |
| Semantic metadata | Integrity rules | <Skal være unikt og må kun være tilknyttet én person> | <Skal være unikt og der må kun være ét studenter ID, studenterkortnummer par> |

Figur 4: Metadata Model af studenter ID og studenterkortnummer i C-READ

## Metadata Model – PAYS

|  |  |  |
| --- | --- | --- |
|  | Data element name | <Saldo> |
| Basic metadata | Data source | <Når en student opretter en PAYS konto, bliver saldo initialiseret til decimaltallet 0,0. Hvis en studerende indsætter eller fratrækker (betaler) kroner på/fra deres PAYS konto, bliver saldo opdateret med beløbet> |
| Description | <Saldo er mængden af kroner der er tilgængelig på den studerendes PAYS konto. Dette er kroner de kan bruge i kantinen> |
| Format and data type | <X.Y – Double> |
| Canonical name | <Saldo> |
| Canonical format | <X.Y> |
| Transformation rules | <> |
| Interface | <MQ, SQL> |
| Semantic metadata | Integrity rules | <Saldo er et decimaltal der repræsenterer antal tilgængelige danske kroner på den studerendes PAYS konto> |

Figur 5: Metadata Model af saldo i PAYS

# Opgave 4 - Enterprise integrationsmønstre (10 point)

## Generelt

Jeg vælger at sender beskeder i XML-formatet, da det er nemt at sende i message queueing systemer og vi har arbejdet med det i timerne.

## C-READ

A screen shot of a computer

Description automatically generated with medium confidence

Figur 6: Eksempel på besked format ved registrering af tilstedeværelse i C-READ

Når en studerende vil registrere deres tilstedeværelse, sender kortlæsningsterminalen en payload til C-READ med følgende dataelementer:

* Dato og tid
  + For at kunne indsende tidspunkt for registrering og validering af lektion.
* Klasseværelse
  + For validering af lektion / at den studerende er i det korrekte klasseværelse.
* Terminal ID
  + For at gøre potential fejlfinding lettere senere hen.
* Studenterkortnummer
  + For at identificere den studerende i C-READ.

Bemærk at i figur 6, kan du se den første payload kortlæsningsterminalen sender. Efter alt relevant data er modtaget fra STAS og SS, ser payloaden ud som i figur 7 på næste side.

A screen shot of a computer program

Description automatically generated with low confidence

## PAYS

Når en studerende vil betale for deres mad/varer i kantinen, sender kortlæsningsterminalen en payload til PAYS med følgende dataelementer:

* Dato og tid
  + For registrering af tidspunkt for køb.
* Terminal ID
  + For at gøre potential fejlfinding lettere senere hen.
* Studenterkortnummer
  + For at identificere den studerende i C-READ og dermed også deres PAYS konto
* Mængde
  + Til brug af PAYS for validering/opdatering af saldo.

A screen shot of a computer code

Description automatically generated with low confidence

Figur 7: Eksempel på besked format ved betaling i PAYS

# Opgave 5 - Enterprise integrationsmønstre (30 point)

A picture containing diagram, plan, line, screenshot

Description automatically generated

Figur 8: Flow diagram af C-READ processer

Hvad du ser i figur 8, er et flowdiagram over hvordan en proces kan flow igennem de forskellige systemer. Lad os gå igennem et eksempel med registrering af tilstedeværelses.

1. Den studerende scanner sit studenterkort på terminalen. Terminalen sender en payload, som der bliver adapteret til vores canonical dataformat
2. Payloaden bliver smidt ind i ”C-READ In” køen, hvor den bliver opfanget af ”C-READ Logic”
3. ”C-READ Logic” validere studenterkortnummeret og sender payloaden videre til ”C-READ Out”. Bemærk at payloaden godt allerede nu kan indeholde en fejlmeddelelse
4. Payloaden gør så igennem en ”Content Filter”, som tager unødvendige data ud af payloaden, f.eks. ”TerminalID”
5. Payloaden går igennem en ”Content Based Router”, som sender payloaden til den korrekte kø baseret på payloadens indhold. Hvis payloaden endnu ikke indeholder ”Class”, så bliver den sendt til ”STAS In”
6. Payloaden bliver først håndteret af ”STAS Logic” og bliver derefter sendt bliver ”Enriched”, med i det her tilfælde, ”Class”, og bliver smidt i ”C-READ Reply”. Bemærk at for at spare plads har jeg placeret ”Content Enricher”en efter køen, men den skulle egentligt være før køen for alle systemer der sender payloaden tilbage
7. Nu ender payloaden tilbage ved ”C-READ Logic”, hvor payloaden endnu engang bliver håndteret og sendes videre
8. Payloaden går igennem samme proces, men den gang til ”SS Logic”, hvor den både får appendet ”Course” og ”Teacher”
9. Beskeden bliver sendt tilbage til ”C-READ Logic” og bliver nu sendt til ”LMS Logic”
10. ”LMS Logic” håndtere registrering af tilstedeværelse, ”C-READ Logic” sender bare nødvendige data
11. Samtidig bliver en besked sendt tilbage til kortlæsningsterminalen, som enten er en succes-eller fejlmeddelelse

# Opgave 6 - Enterprise Integration Patterns (20 point)

## Generelt

Jeg har valgt at implementere en ”Content Based Router”, da det er specifik type af ”Message Router” og fordi det er det jeg har brugt til min flowdiagram. Jeg har gjort mig umage med kommentar, så jeg håber det forklare koden nok til at kunne forstå det. Jeg bruger RabbitMQ, da jeg arbejder på MacOS og har derfor ikke adgang til MSMQ.

## Kode

### Router

A screenshot of a computer program

Description automatically generated with medium confidence

### Simuleret sender

A screenshot of a computer program

Description automatically generated with medium confidence

### XML Datafil

A screenshot of a computer program

Description automatically generated with medium confidence

## Output

### STAS

A screenshot of a computer

Description automatically generated

### SS

A screenshot of a computer

Description automatically generated

### LMS

A screenshot of a computer

Description automatically generated

### PAYS

A screenshot of a computer

Description automatically generated

### Out

A screenshot of a computer

Description automatically generated

## RabbitMQ Management

### Overview

A picture containing text, screenshot, number, font

Description automatically generated

### Eksempel

A screenshot of a computer

Description automatically generated with medium confidence