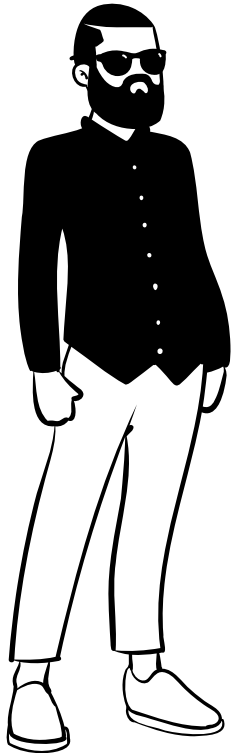


# OEPFELBAUM TECH ASSESSMENT

# TEAM SETUP UND ABLAUF

**Yves**  
Business Analyst



**Lukas Yu**  
Lead Developer



**Du**  
Software Engineer



Sprint Kick-Off



**03.10.2025 (30m)**

Sprint



**5 Tage**  
zur freien Verfügung

Sprint Review



**08.10.2025 (1.5h)**

Interner Review



Feedback



**~ 10.10.2025**  
ad-hoc

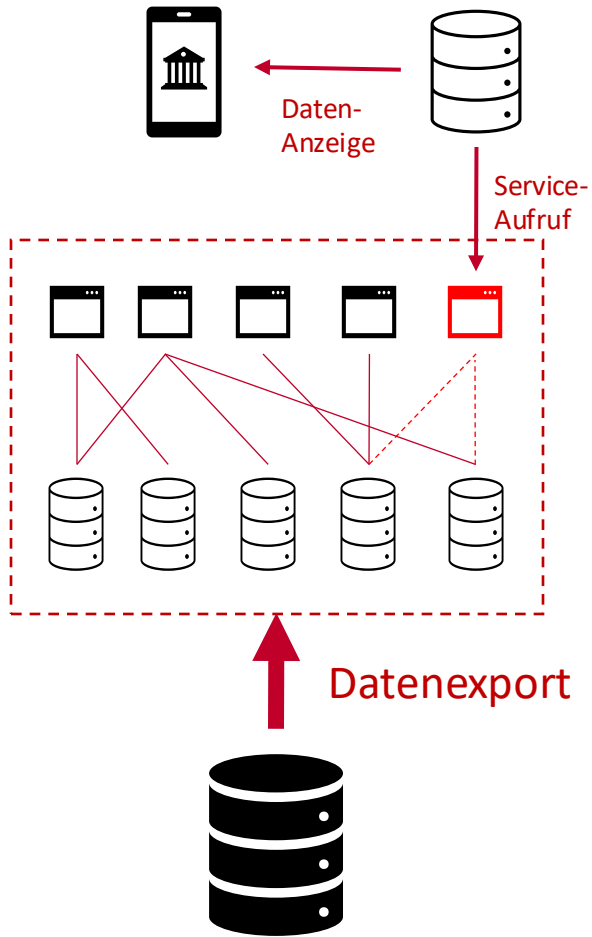
# PROJEKT – KREDIT SERVICE

## Ausgangslage:

- Unser Kunde hat ein Frontend-System gekauft, mit dem Kredite den Bankkunden angezeigt werden können.
- Das Frontend-System soll “out-of-the-box” verwendet werden, das heisst, an den Funktionalitäten im Frontend darf nichts verändert werden.
- Das Frontend-System ruft die Kredit-Informationen durch einen REST-API Call ab.
- Das Frontend-System besitzt eine eigene Datenmodellierung für Bankkredite, die sich vom dem des Kernbankensystems unterscheidet.



# ARCHITEKTUR-ÜBERSICHT



## Frontend System (FS)

Das FS ruft den benötigten Service für die Anzeige der Kredite aus dem SL auf und stellt die ermittelten Daten im UI dar.

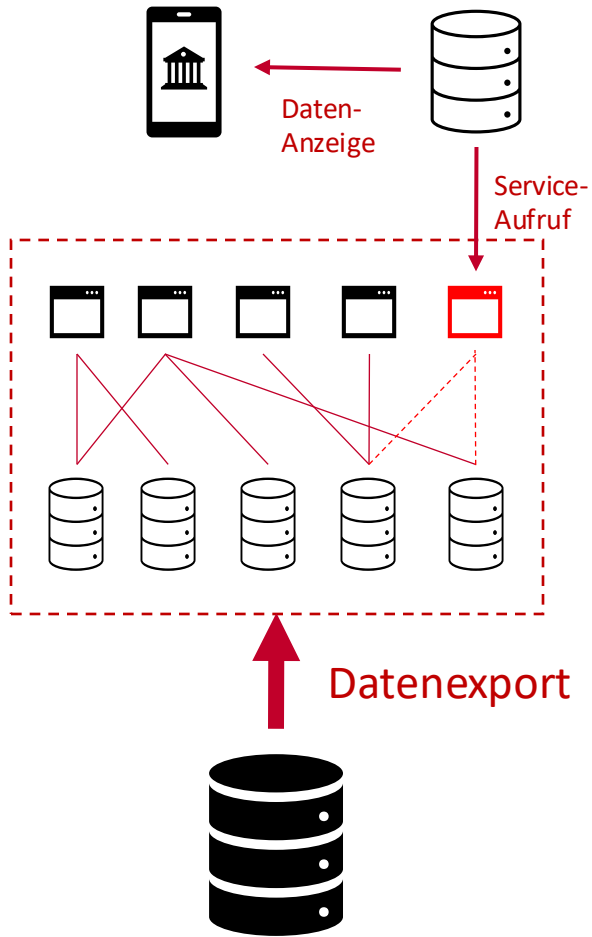
## Service Layer (SL)

Das SL besteht aus verschiedenen Datenbanken und Services, welche Daten aus dem KBS transformieren und für Konsumenten wie bspw. dem Frontend-System zur Verfügung stellen.

## Kernbankensystem (KBS)

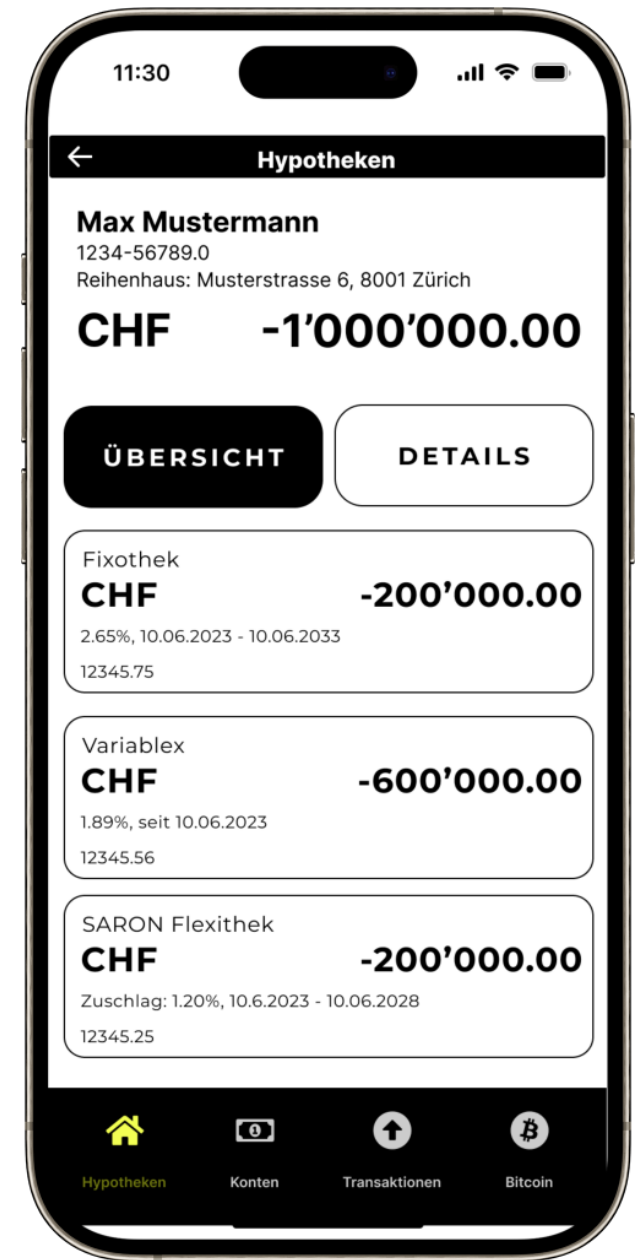
Das KBS fungiert als Master-System und beinhaltet alle Stamm- und Transaktionsdaten der Bank. Die Daten werden dem SL event-basiert zur Verfügung gestellt.

# AUFGABE



## Aufgabe:

- Erstelle einen Service, welcher die Daten vom KBS empfängt, sammelt und sie für den Aufruf durch das Frontend-System aufbereitet.
- Der Service muss in der Lage sein, die Anfragen vom Frontend-System zu empfangen und diese gemäss Frontend-API-Spezifikation mit den Daten aus dem KBS zu beantworten.



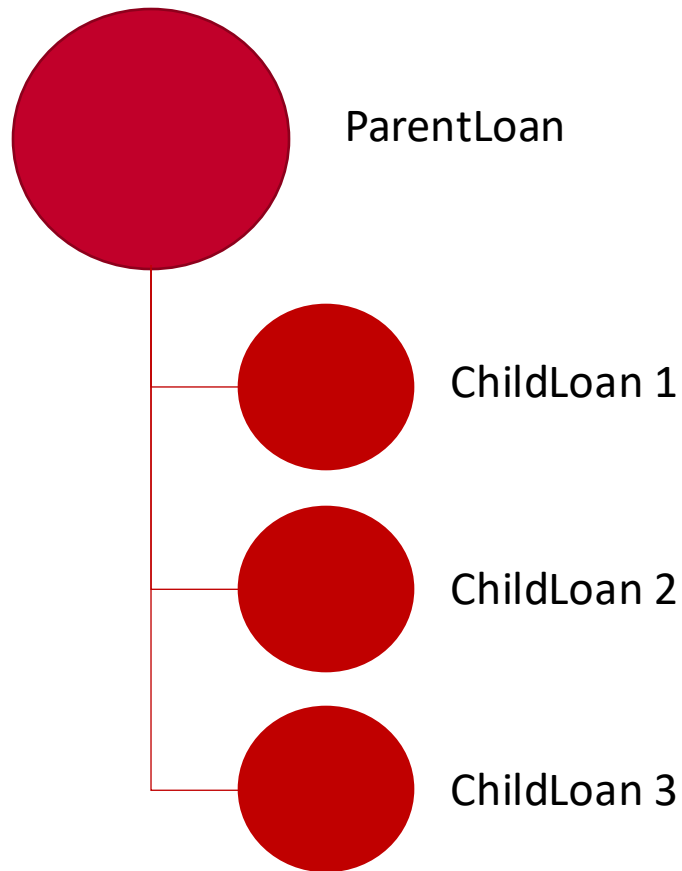


# DATENMODELL KBS

- Im KBS bestehen Kredite aus 3 Entitäten.
- Die Hauptentität ist das **Financing Object** welches alle Entitäten zusammen hält und den Kredit mit dem Bankkunden verknüpft.
- Jedes Financing Object hat genau ein **Limit** welches das Kreditlimit und die hinterlegten Sicherheiten abbildet.
- Ein Financing Object enthält ein oder mehrere **Products** welche jeweils einen Zins und eine gewisse Laufzeit definieren.

Financing Object	Limit	Product
<ul style="list-style-type: none"><li>- id: long</li><li>- Owner: Obj<ul style="list-style-type: none"><li>o id: long</li><li>o Name: String</li></ul></li><li>- limit: long</li><li>- products: long[]</li><li>- status: String (enum inactive/active)</li></ul>	<ul style="list-style-type: none"><li>- id: long</li><li>- name: String</li><li>- type: String (enum LimitTypes)</li><li>- limitAmount: double</li><li>- amortisationAmountAnnual: double</li><li>- agreedAmortisationFrequency: int</li><li>- contractNumber: String</li><li>- realSecurities: Obj[]<ul style="list-style-type: none"><li>o type: String</li><li>o address: String</li><li>o collateralValue: long</li><li>o currency: String (enum ISO3 Code)</li><li>o nextRevaluationDate: Date</li></ul></li></ul>	<ul style="list-style-type: none"><li>- id: long</li><li>- name: String</li><li>- type: String (enum ProductTypes)</li><li>- amount: long</li><li>- currencyCode: String (enum ISO3 Code)</li><li>- interestRate: double</li><li>- startDate: Date</li><li>- endDate: Date</li><li>- productNumber: String</li><li>- defaultSettlementAccountNumber: String</li><li>- interestDue: double</li><li>- isOverdue: boolean</li><li>- interestPaymentFrequency: int</li></ul>

# DATENMODELL FE



- Das FE bildet Kredite mit einer **Loan** Entität ab.
- Die Hierarchie wird hier über die Felder "loanType" und "parentLoanId" abgebildet.
- Ein Kredit vom loanType "ParentLoan" hat ein oder mehrere Loans mit loanType "ChildLoan".
- Die genauen Datenfelder und Werte sind in der OpenAPI Definition ersichtlich.

# UNTERLAGEN

- Diese Präsentation
  - User-Stories
- OpenAPI-Spezifikation
  - OutboundIntegrationAPI\_LoansService.yaml
- Testdaten von Kernbankensystem
  - FINANCING\_OBJECT.json
  - LIMITS.json
  - PRODUCTS.json



# DELIVERABLES

- Java/Kotlin Projekt
  - Git Repository mit History (Link zu Github o.Ä. reicht)
  - Lokal lauffähige Applikation
  - OpenAPI-Spezifikation implementiert
  - User Stories implementiert und mit Unit-Tests abgedeckt
  - "Getting started" im README.md
  - Frameworkauswahl offen
- Kleine Demo / Präsentation beim Sprint Review

# HINWEISE

- Komm gerne jederzeit auf uns zu wenn du Unklarheiten oder Fragen hast.
- Du darfst gerne Annahmen treffen, damit du nicht zu viel Zeit mit Rückfragen verlierst.
- Dokumentiere deine Annahmen im Code oder im README.
- Aufwand: ½ bis 1 Tag

Kontakte:

Lukas Yu – [lukas.yu@oepfelbaum.com](mailto:lukas.yu@oepfelbaum.com)

# USER STORIES

- Die User Stories in den folgenden Folien definieren (unter anderem) komplexere Mappings von einzelnen Feldern.
- Bei allen anderen Feldern kann ein (nahezu) 1:1 Mapping angenommen werden, falls du eine andere Annahme triffst dokumentiere diese.

ID: S1	Status: Open
Title: Provide loans for userId	
Story: As a Online Banking user, I want to be able to see any loans with the bank I'm the owner of, so I can stay on top of my lendings.	
Acceptance Criteria: <ul style="list-style-type: none"> <li>• The API endpoint /service/v1/loansByUser/{userId} is implemented according to the openAPI contract</li> <li>• The list of loans returned only contains loans owned by the user with the provided userId</li> </ul>	

ID: S2	Status: Open
Title: Sum up outstanding amounts of child loans in parent loan	
Story: As a Online Banking user, I want to be able to quickly see the total amount of my mortgage, so I don't have to add the amount of each product within my mortgage.	
Acceptance Criteria: GIVEN a financing object with two products "prodA" & "prodB" AND "prodA" outstandingAmount = 120'000 AND "prodB" outstandingAmount = 85'000 WHEN the data is returned from the API THEN the parentLoan outstandingAmount is set to 205'000 AND the two childLoan ousstandingAmounts are set to 120'000 and 85'000 respectively	

ID: S3	Status: Open
Title: Find lending date range based on product date ranges	
Story: As a Online Banking user, I want to be able to see the overall start and end dates of my lending, so I can easily see when my lendings will end.	
Acceptance Criteria: GIVEN a financing object with two products "prodA" & "prodB" AND "prodA" startDate = 15.12.2020, endDate = 15.12.2030 AND "prodB" startDate = 01.11.2020, endDate = 01.11.2025 WHEN the data is returned from the API THEN the parentLoan startDate = 01.11.2020, endDate = 15.12.2030 AND the two childLoan startDate and endDate are set to startDate = 15.12.2020, endDate = 15.12.2030 and startDate = 01.11.2020, endDate = 01.11.2025 respectively	

ID: S4	Status: Open
Title: Identify if any part of a lending is overdue	
Story: As a Online Banking user, I want to be able to see if one of my lendings is overdue, so I can easily see if I've missed a payment.	
<p>Acceptance Criteria:</p> <p>#1</p> <p>GIVEN a financing object with two products "prodA" &amp; "prodB"</p> <p>AND "prodA" isOverdue = true</p> <p>AND "prodB" isOverdue = false</p> <p>WHEN the data is returned from the API</p> <p>THEN the parentLoan isOverdue = true</p> <p>AND the two childLoan isOverdue is set to true and false respectively</p> <p>#2</p> <p>GIVEN a financing object with two products "prodA" &amp; "prodB"</p> <p>AND "prodA" isOverdue = false</p> <p>AND "prodB" isOverdue = false</p> <p>WHEN the data is returned from the API</p> <p>THEN the parentLoan isOverdue = false</p> <p>AND the two childLoan isOverdue is set to false for both</p>	



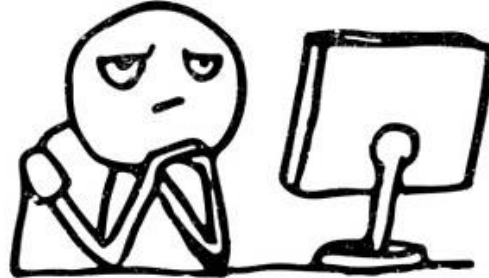
ID: S5	Status: Open
Title: Calculate next amortisation payment amount for lending	
Story: As a Online Banking user, I want to be able to see my next amortisation payment amount and the yearly frequency, so I can easily see my overall repayment strategy.	
Acceptance Criteria: GIVEN a financing object with a limit "LimA" AND "limA" amortisationAmountAnnual = 5000 AND "limA" agreedAmortisationFrequency = 4 WHEN the data is returned from the API THEN the parentLoan amortisationPaymentAmount = 1250 (5000/4) AND the parentLoan paymentFrequency = 4 AND the childLoan(s) amortisationPaymentAmount = null and paymentFrequency = null	

ID: S6	Status: Open
Title: Identify when the next interest payment is due	
Story: As a Online Banking user, I want to be able to see my next interest payment due date, so I can plan my payments.	
Acceptance Criteria: GIVEN a financing object with two products "prodA" & "prodB" AND "prodA" interestDue = 01.01.2024 AND "prodB" interestDue = 01.03.2024 WHEN the data is returned from the API THEN the parentLoan interestDue = 01.01.2024 AND the childLoans interestDue is set to 01.01.2024 and 01.03.2024 respectively	

ID: S7	Status: Open
Title: Show interest rate on the child loan	
Story: As a Online Banking user, I want to be able to see my next interest payment rates and frequency per product, so I can see my interest strategy for each product.	
Acceptance Criteria: GIVEN a financing object with two products "prodA" & "prodB" AND "prodA" interestRate = 2.5, interestPaymentFrequency = 2 AND "prodB" interestRate = 1.2, interestPaymentFrequency = 6 WHEN the data is returned from the API THEN the parentLoan interestRate = null, interestPaymentFrequency = null AND the childLoans interestDue is set to interestRate = 2.5, interestPaymentFrequency = 2 and interestRate = 1.2, interestPaymentFrequency = 6 respectively	

# Q&A

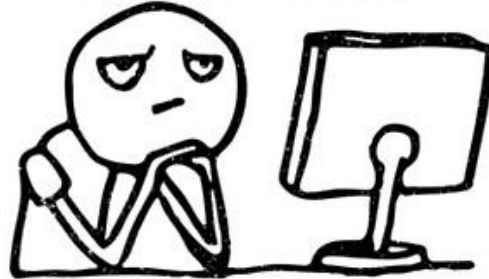
THE CODE DOESN'T WORK



WHY?

---

THE CODE WORKS



WHY?