

Sachbericht zum Verwendungsnachweis

WIR! - Blockchain - Verbundvorhaben

**UNiversal Token COLLateralisation System
UNICO**

Teilprojekt 1:
Entwicklung einer Web-Anwendung zur Emission, Verwaltung
und Abwicklung dezentraler Sicherungsgegenstände

**Dr. Ronny Kunz
Matthias Kramer**

splainX GmbH
Bahnhofstr. 32
09648 Mittweida



**Projektträger:
Projektträger Jülich**

Förderkennzeichen: 03WIR1326A
Zuwendungsbescheid: 20.09.2023
Bewilligungszeitraum: 01.11.2023 bis 31.01.2025

Gefördert durch:



Bundesministerium
für Forschung, Technologie
und Raumfahrt

Mittweida, 26.07.2025

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

1. Kurzbericht

1.1 Aufgabenstellung

Das Projekt UNICO beschäftigte sich mit der Tokenisierung und Abwicklung zustandsabhängiger Zahlungsansprüche basierend auf Statusangaben zu emittierten Security Token Dritter. Gegenstand war die systemunabhängige Einbindung von dezentral verwalteten, tokenisierten Kreditsicherheiten in den Begebungs- und Abwicklungsprozess tokenisierter Verbindlichkeiten, sogen. Debt Tokens. Dabei sollte die automatisierte Verwertung bzw. Inanspruchnahme tokenisierter Kreditsicherheiten im Falle einer vom Emittenten eines Debt Token zu verantwortenden Leistungsstörung, d.h. eines Credit Events, ermöglicht werden. Die Aufgabenstellung des Projekts bestand in der Entwicklung eines prototypischen Demonstrators einer Systemlandschaft, in dem Emittenten oder Inhaber von Debt Token diese unabhängig von einem Tokenisierungsnetzwerk (Emissionsplattform) nach Bedarf mit Kreditsicherheiten ausstatten können. Ferner sollte das System die automatisierte Verwertung bzw. Inanspruchnahme der Kreditsicherheiten im Falle des Eintritts eines Credit Events ermöglichen. Dies setzte die Formalisierung der Definitionen für Credit Events, Mechanismen zur Messung des Eintritts dieser sowie den Aufbau eines (dezentralen) Systems zur Verwaltung von Einträgen zu Credit Events voraus. Die prototypische Systemlandschaft soll im Anschluss an die Förderprojektphase im Rahmen einer Produktentwicklung an die Gegebenheiten und Erfordernisse potenzieller Anwender angepasst und durch die splainX GmbH wirtschaftlich vermarktet werden. Arbeitsziel 1 umfasste dabei die *Erhebung von für die Tokenisierung geeigneten Kreditsicherheiten*. Die *Konzeption und der Aufbau eines dezentralen Registers* waren Gegenstand des zweiten Arbeitsziels. Arbeitsziel 3 bestand in der *Konzeption und Entwicklung eines dezentralen Protokolls*.

1.2 Wissenschaftlicher und technischer Ausgangsstand

Der wissenschaftliche und technische Ausgangsstand des Projekts UNICO basiert auf den Entwicklungen der Blockchain-Technologie und ihrer Anwendung im Finanzsektor. Seit der Einführung von Bitcoin im Jahr 2009 hat sich die Blockchain zu einer etablierten Lösung für die dezentrale Verwaltung und Verifikation digitaler Transaktionen entwickelt. Insbesondere im Finanzbereich kommen Distributed-Ledger-Technologien (DLT) zunehmend zum Einsatz, um tokenisierte Finanzinstrumente effizient zu emittieren und abzuwickeln. Mittels regelbasierter Funktionen und Abläufen in sogen. Smart Contracts können klassische, papierbasierte Verträge, insb. Finanzinstrumente aber auch Kreditsicherheiten, digital abgebildet und automatisiert ausgeführt werden. Aufgrund der Validität und Authentizität der dezentral verwalteten Daten sind zentrale Instanzen, bspw. Market Maker, nicht mehr notwendig. Es kann beobachtet werden, dass der Markt für tokenisierte Fremdkapitalinstrumente, wie Schuldverschreibungen und Schuldscheine, stetig wächst. So wird für das Jahr 2026 ein Emissionsvolumen sogen. Debt Token i.H.v. ca. EUR 220,0 Mrd. prognostiziert (ca. EUR 50,0 Mrd. in 2022). Während durch Smart Contracts darlehensähnliche Zahlungsansprüche automatisiert dargestellt und verarbeitet werden können, besteht weiterhin ein Ausfallrisiko der Emittenten. Um dieses Risiko zu steuern, sind Kreditsicherheiten ein bewährtes Mittel, deren Tokenisierung einen logischen nächsten Schritt darstellt. Durch die Verknüpfung von tokenisierten Kreditsicherheiten mit Debt Tokens sowie die systemunabhängige, automatisierte Verwertung dieser Sicherheiten im Falle eines Credit Events entsteht eine innovative Lösung zur Effizienzsteigerung der Finanzmarktprozesse. UNICO adressiert diese Herausforderung durch die Entwicklung einer prototypischen Systemlandschaft, die eine dezentrale Verwaltung von Sicherheitsleistungen ermöglicht und damit neue Standards in der tokenisierten Fremdkapitalfinanzierung setzt.

1.3 Ablauf des Vorhabens

Das Vorhaben wurde mit dem Ziel geplant, den prototypischen Demonstrator innerhalb der Projektlaufzeit von 15 Monaten zu entwickeln. Dafür wurde das Gesamtvorhaben in sieben Arbeitspakete (AP) gegliedert und wie folgt durchgeführt:

AP 1: Pilotierung dezentrales Credit Event-Register

- Definition relevanter Credit Events
- Konzeption von Prüfroutinen zum Eintritt von Credit Events
- Automatisierte Einträge dezentrales Credit Event-Register
- Funktionstest Credit Event-Register
- Datenschutz

AP 2: Repräsentation geeigneter Kreditsicherheiten als Security Token

- Identifikation rechtlicher Anforderungen für die Tokenisierung von Kreditsicherheiten
- Bewertung der Rechtsposition aus tokenisierten Kreditsicherheiten

- Marktanalyse zu geeigneten tokenisierten Kreditsicherheiten
- Konzeption standardisierter Formularverträge Kreditsicherheiten
- Schaffung technisches Umfeld zur nutzerorientierten Tokenisierung von Kreditsicherheiten

AP 3: Pilotierung dezentrales Besicherungsregister

- Konzeption
- Implementierung dezentrales Besicherungsregister

AP 4: Dezentrales Protokoll zur Initiierung der Inanspruchnahme & Verwertung von Kreditsicherheiten

- Konzeption
- Integration Credit Event-Register und Kreditsicherheit
- Entwicklung des dezentralen Protokolls
- Funktionstest des dezentralen Protokolls

AP 5: Aufbau Demonstrator

- Analyse und Auswahl geeigneter Technologien und Frameworks
- Aufbau eines physischen Simulationsnetzwerkes
- Implementierung

AP 6: Sicherstellung der Nachhaltigkeit

- Evaluation der Projektergebnisse
- Systemdokumentation
- Übertragbarkeit der Projektergebnisse

AP 7: Projektkoordination und Öffentlichkeitsarbeit

1.4 Wesentliche Ergebnisse

Innerhalb der Projektlaufzeit erfolgte entsprechend der Zielstellung der Aufbau eines prototypischen Demonstrators einer Systemlandschaft, mittels dessen die Tokenisierung von Kreditsicherheiten, deren Zuordnung zu Debt Token, die Überwachung dieser im Hinblick auf den Eintritt eines Credit Events und die automatisierte Abwicklung des dezentralen Besicherungsprotokolls möglich ist. Die nachstehende Abbildung visualisiert den Prozessablauf des entwickelten UNICO-Systems, das die Abwicklung von Krediten und Sicherheiten (Collaterals) unterstützt. Zunächst wird ein Kreditvertrag zwischen dem Debitor (Schuldner) und dem Creditor (Gläubiger) abgeschlossen. Dieser Schritt erfolgt außerhalb des Systems. Anschließend werden die Kreditinformationen durch den Creditor in das UNICO-System übermittelt. Der Debitor hinterlegt daraufhin die vereinbarten Sicherheiten im System. Im nächsten Schritt bestätigen sowohl der Debitor als auch der Creditor den Kredit sowie die hinterlegten Sicherheiten. Danach sperrt das System das hinterlegte Collateral, sodass es ausschließlich im Zusammenhang mit dem Kredit verwendet werden kann. Anschließend erstellt der Creditor die notwendigen Zahlungsinformationen, die für die Abwicklung des Kredits benötigt werden. Sobald die Zahlung fällig wird, wird sie vom System UNICO automatisch ausgelöst. Der Status der Zahlung – ob erfolgreich oder fehlgeschlagen – wird anschließend im System dokumentiert. Bei einem Zahlungsausfall (Failure) zieht das System das hinterlegte Collateral ein und aktualisiert die Kreditinformationen. Im Falle eines erfolgreichen Zahlungsvorgangs (Success) können weitere Zahlungen erfolgen, bis der Kredit vollständig zurückgezahlt ist. Nach Abschluss aller Zahlungen wird das verbleibende Collateral dem Debitor zurückerstattet. Das System UNICO fungiert dabei als zentrale Plattform, die alle Schritte koordiniert, Daten verwaltet, die entsprechenden Transaktionen abwickelt sowie sämtliche Vorgänge automatisiert und überwacht.



Abbildung 1: Prozessablauf der UNICO-App

1.5 Zusammenarbeit mit anderen Forschungseinrichtungen

Das Vorhaben wurde in Zusammenarbeit zwischen der splainX GmbH und der Hochschule Mittweida University of Applied Sciences (HS Mittweida) durchgeführt. Während der Projektlaufzeit übernahm die splainX GmbH die Rolle des Verbundkoordinators.

2. Eingehende Darstellung

2.1 Erzielte Ergebnisse

Im folgenden Abschnitt werden für alle Arbeitspakete des Projektes UNICO die in der Vorhabenbeschreibung formulierten Ziele den bei Projektabschluss im jeweiligen Arbeitspaket erzielten Ergebnissen gegenübergestellt. Für jedes Arbeitspaket wird dabei ein Fazit gezogen, welches den Grad der Zielerreichung beschreibt.

2.1.1 AP 1: Pilotierung dezentrales Credit Event-Register

Ziele

AP 1 diente der Pilotierung des dezentralen Credit Event-Registers. Das Ziel dieses AP unter Leitung der splainX GmbH bestand in dem technischen Aufbau des SOLL-Zustands des dezentralen Credit Event-Registers (Schnittstellen und Protokolle) sowie dem semantischen Aufbau und der Erstellung von Transformations-schemata der Daten. Zudem zielte das AP auf das Design der System- und Prozesslandschaft auf Basis passender Modellierungssprachen ab. Ferner sollte eine Übersicht über zu beachtende Richtlinien bezüglich des Datenschutzes und der Datensicherheit erstellt sowie deren Einhaltung gewährleistet werden. Das AP beinhaltete ebenfalls die Entwicklung des Prototyps für das dezentrale Credit Event-Register in einer Demonstrator-Umgebung.

Ergebnisse

Im Rahmen der Pilotierung eines dezentralen Credit Event-Registers stand zunächst die Definition und technische Modellierung relevanter Kreditereignisse im Fokus. Ziel war es, Credit Events zu identifizieren, die für den Collateralisation-Prozess eine zentrale Rolle spielen, und diese in ein standardisiertes Datenmodell zu überführen.

Eine umfassende Recherche und Analyse zeigten, dass die Integration verschiedener Kreditereignisse und deren Erfassung über externe, öffentlich zugängliche Schnittstellen mit erheblichen technischen Herausforderungen verbunden ist. Insbesondere erwies sich die Standardisierung solcher Informationen als schwierig. Aus diesem Grund wurde entschieden, sich im Prototyp auf Zahlungsausfälle als einziges relevantes Credit Event zu konzentrieren. Diese Entscheidung basiert auf der hohen praktischen Relevanz von Zahlungsausfällen und ihrer vergleichsweise einfachen technischen Umsetzbarkeit.

Für die praktische Erfassung für das Credit Event Zahlungsausfall wurden die Schnittstellen des splainX-Systems genutzt, das geeignete Zahlungsinformationen bereitstellt. Zur Modellierung wurde ein Blockchain-basiertes System entwickelt, in dem Zahlungsausfälle als spezielle Token repräsentiert werden. Diese enthalten die wesentlichen Merkmale eines Zahlungsausfalls, darunter den Zahlenden, den Zahlungsempfänger, die Fälligkeit sowie den Betrag. Bei dem Token, das auf der Blockchain entsteht, handelt es sich also um ein Zahlungsinformations-Token mit den genannten Informationen. Die Blockchain ermöglicht es, automatisch Ereignisse zu generieren, sobald ein Zahlungs-Token entsteht oder die erwartete Zahlung ausbleibt.

Das Ergebnis dieses Prozesses ist ein robustes und technisch umsetzbares Datenmodell, das Zahlungsausfälle präzise erfasst und die Grundlage für die weitere Entwicklung des dezentralen Credit Event-Registers bildet.

Die nachstehende Abbildung 2 zeigt die ursprünglich anvisierte, ideelle Anwendungslandschaft, welche externe Anwendungen inkludiert. Bedauerlicherweise ist die vorgesehene Integration solcher Anwendungen (3rd parties), welche weitere Credit Events auf die Blockchain schreiben, nicht gelungen, da sich das öffentliche Abgreifen solcher Informationen sowie deren Standardisierung als nicht realisierbar erweist.

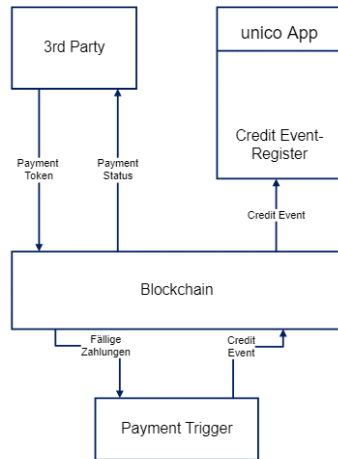


Abbildung 2: Ideelle Anwendungslandschaft

Die nachfolgende Abbildung 3 visualisiert die tatsächlich umgesetzte Anwendungslandschaft, in welcher statt einer 3rd party auf die vorhandene splainX-Anwendung zurückgegriffen wurde.

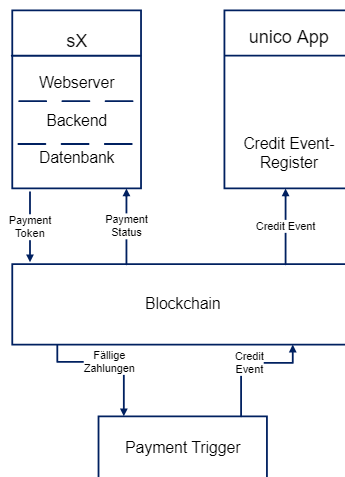


Abbildung 3: Realisierte Anwendungslandschaft

Die Anwendungslandschaft illustriert den Daten- und Prozessfluss innerhalb des dezentralen Credit Event-Systems und besteht aus vier zentralen Komponenten: der Blockchain, der vorhandenen splainX-Anwendung, der entwickelten unico-App und dem Payment Trigger. Die Blockchain dient als zentraler Infrastrukturbaukasten. Sie verarbeitet die von sX übermittelten Payment Tokens und registriert deren Status, d.h. ob Zahlungen erfolgreich waren oder fehlschlagen sind. Bei einem Zahlungsausfall wird ein Ereignis (Payment Failed) ausgelöst, das an die unico-App übermittelt wird. Die splainX-Anwendung, bestehend aus Webserver, Backend und Datenbank, fungiert als Datenquelle, die Informationen über Zahlungstransaktionen in Form von Payment Tokens bereitstellt. Diese Tokens enthalten relevante Zahlungsdetails, die an die Blockchain weitergeleitet werden. Die unico-App dient schließlich als zentrales Credit Event-Register. Sie empfängt die von der Blockchain gesendeten Kreditereignisse (Zahlungsausfälle) und speichert und verwaltet diese in einem zentralen Credit Event-Register, das den Nutzern als Informationsquelle dient. Über eine direkte Anbindung an die Blockchain erhält die App Ereignismeldungen und aktualisiert den Status der registrierten Kreditereignisse in Echtzeit. Der Payment Trigger (A2A-EBICS) überwacht die Zahlungsprozesse und prüft fällige Zahlungen. Sobald eine Zahlung fällig ist, initiiert der Trigger einen entsprechenden Prozess auf der Blockchain, um den Status der Zahlung zu prüfen (erfolgreich oder fehlgeschlagen).

Der Prozessablauf lässt sich wie folgt beschreiben:

1. **Erstellung eines Payment Tokens:**
Das sX-System erzeugt einen Payment Token mit relevanten Informationen (z. B. Zahler, Zahlungsempfänger, Betrag, Fälligkeitsdatum) und überträgt diesen an die Blockchain.
2. **Verarbeitung in der Blockchain:**
Die Blockchain nimmt das Token entgegen, prüft den Status der Zahlung und initiiert bei Bedarf ein Ereignis ("Payment Failed").
3. **Auslösung eines Zahlungsausfalls:**
Der Payment Trigger erkennt, ob eine Zahlung fehlschlägt, und meldet dieses Ereignis an die Blockchain.
4. **Übertragung an die unico-App:**
Die Blockchain sendet Kreditereignisse (Zahlungsausfälle) an die unico-App, die diese Informationen für den Endnutzer bereitstellt.

1. Technischer Aufbau des SOLL-Zustands

Im Rahmen von AP 1 wurde ferner eine Infrastruktur- und Schnittstellenanalyse zu den Credit-Events insb. zu dem speziellen Credit-Event Zahlungsausfall durchgeführt. Aus diesen resultierte die Anforderungsanalyse, auf deren Basis im Anschluss die Entwicklung und der technische Aufbau des SOLL-Zustands erfolgte. Das Entwicklerteam arbeitete dabei in einer containerisierten Umgebung (Docker). Die Server-Architektur in der Docker-Umgebung besteht als Ergebnis dieses Schritts aus den folgenden Komponenten:

- Private Blockchain-Node(s)
- Datenbank-Server
- Backend-Server
- Frontend-Server
- Container zur Abwicklung der A2A-EBICS-Zahlungen (Teil der splainX-Anwendung)

Die nachstehende Abbildung veranschaulicht die Software-Architektur.

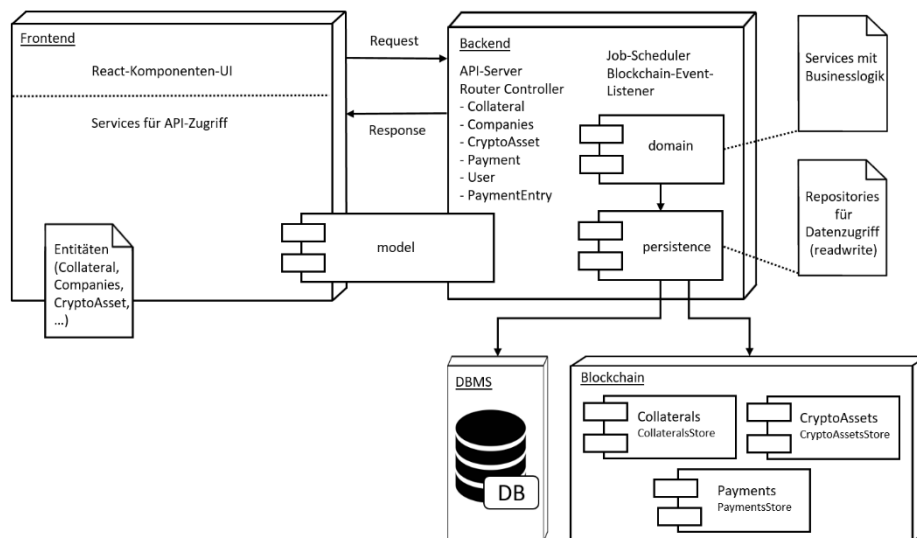


Abbildung 4: Software-Architektur

2. Semantischer Aufbau und Transformationsschemata der Daten

Der aus der Anforderungsanalyse entstandene semantische Aufbau lässt sich als Entity Relationship Modell (ERM) darstellen. Dieses wurde für die Bereiche Benutzerverwaltung, Zahlungsmanagement, Credit Events und Collaterals erstellt. Das entwickelte ERM umfasst zum einen die innerhalb der Anwendung verwendeten Entitäten, d.h. Benutzer, Unternehmen (Kreditor oder Debitor), Kredit, Collateral (Besicherung) und Rückzahlungsrate (Zahlungsplan, bestehend aus Zinsen und Tilgung) mit ihren spezifischen Attributen. Der Entität Benutzer sind bspw. die Eigenschaften *Anmeldename*, *Passwort* und *Blockchain-Account-ID* zugeordnet. Kredite werden u.a. durch die Merkmale *Kreditbetrag*, *Zinsrate* und *Endfälligkeit* charakterisiert. Zum anderen dient das ERM der Darstellung der Beziehungen der Entitäten unter- bzw. zueinander. So sind einem Kredit mehrere Rückzahlungsraten zugeordnet.

Hinsichtlich der zu erstellenden Transformationsschemata der Daten kann als Ergebnis der entwickelte Event Handler benannt werden. Der Event Handler stößt eine Funktion im Smart Contract an, indem er ein Event registriert und daraufhin eine Aktion auslöst. Hinsichtlich der Credit Events wird der eingetretene Zahlungsausfall vom Payment Trigger gemeldet. Dieser Ausfall muss als Status an den Payment Token hinterlegt und im Token gespeichert werden. Diese Aufgabe wird von einem bestimmten Smart Contract übernommen. Nach diesem Prinzip sollen künftig neben dem Zahlungsausfall auch weitere Credit Events verarbeitet werden.

3. System- und Prozesslandschaft auf Basis passender Modellierungssprachen

Im Rahmen des Scrum-Entwicklungsprozesses wurden User Stories entwickelt. Eine User Story beschreibt einen Teilprozess detailliert aus Sicht einer konkreten User-Rolle, d.h. des Kreditors oder des Debtors und liefert den Entwicklern Vorgaben für die softwaretechnische Umsetzung. In der User Story werden Rolle, Funktionalität, erwarteter Nutzen sowie eindeutig definierte Akzeptanzkriterien zur Überprüfung der Implementierung einer User Story dargestellt. Die folgende Abbildung 5 illustriert beispielhaft eine der User Stories. Die erstellten User Stories dienen gleichzeitig als Ausgangspunkt für automatisierte Tests, in welchen die definierten Akzeptanzkriterien umgesetzt werden. Dadurch wird die exakte Abbildung der Prozesse in der Anwendung sichergestellt.

Integrate Loan Information into the UNICO system

+ Add @ Apps

Description

As a **creditor, I, Charles**, want to integrate loan agreement details into the UNICO system after concluding a loan agreement with a debtor, so that the credit information is systematically recorded and accessible.

Description:

After concluding a loan agreement with a debtor named Daniel, I, Charles, want to create a new entry in the UNICO system to store the credit information. The required information includes:

- **Name:** string
- **Amount:** number
- **Interest Rate:** number
- **Debtor Name:** string
- **Debtor IBAN:** string
- **Initial Amortization Rate:** number (optional)
- **Emission Date:** Date
- **Due Date:** Date (optional)
- **Token Amount:** number (default 1)
- **Asset Type:** AssetType
- **Loan Type:** LoanType
- **Interval Type:** IntervalType

Tasks:

- **Design the Form:**
 - Create the form layout with all specified fields.
 - Implement input validation for each field.
- **Backend Integration:**
 - Develop API endpoints to save and retrieve loan information.
 - Ensure data is correctly stored in the database.
 - Add loanStatus: unconfirmed, confirmed, active, failed, completed
- **Frontend Development:**
 - Implement the form in the UNICO system UI.
 - Create a tab for Create Credit → only show when a creditor is logged in
 - Create a Page Component
 - Title and optional extra prop.
 - Content Wrapper below
 - Create the loan details page to display all relevant information.
- **Testing:**
 - Write unit tests for form validation.
 - Perform end-to-end testing to ensure the entire flow works as expected.

Dependencies:

- Database schema updates to accommodate new loan-related fields.
- Frontend framework for form and page rendering.
- Backend services for data storage and retrieval.

Acceptance Criteria:

- 1. Form Creation**
 - A form should be designed to capture all the specified fields.
 - The form should have clear labels and input validation to ensure data accuracy.
 - The "Token Amount" field should default to 1.
- 2. Data Storage**
 - Upon submission, the form data should be saved into the database accurately.
 - The system should handle optional fields correctly, allowing them to be left blank.
- 3. Separate Page Display**
 - There should be a dedicated page for loan entries.
 - This page should display all relevant information clearly and comprehensively.
- 4. Navigation and Accessibility**
 - The system should allow easy navigation to the loan details page from the navbar.
 - The layout of the form and details page should be user-friendly and accessible.
- 5. Validation and Error Handling**
 - Implement appropriate validation for each field (e.g., correct date format, numerical values for amount and interest rate).
 - Provide meaningful error messages for invalid input.

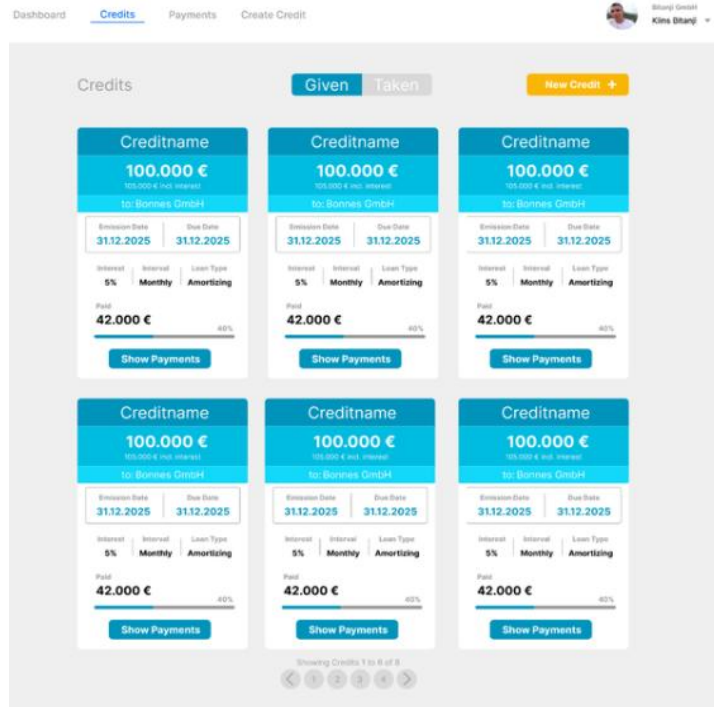
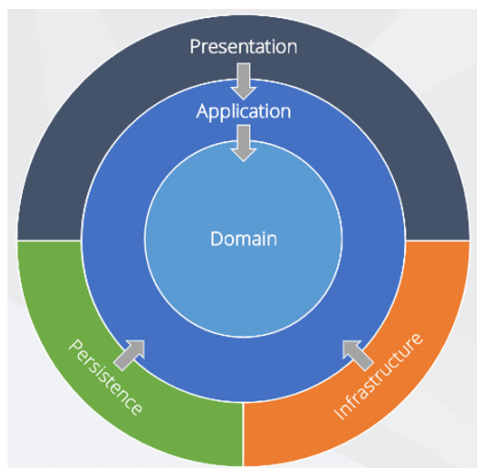


Abbildung 5: Beispiel der User Story "Integrate Loan Information into the UNICO system"

Basierend auf den User Stories wurden die noch nicht vollständig definierten SOLL-Prozesse entwickelt und modelliert. Abbildung 1 veranschaulicht bereits den modellierten Hauptprozess, der durch weitere Prozesse in Form von User Stories ergänzt wurde. Die Systemarchitektur wurde unter Berücksichtigung der Anforderungen modelliert und folgt dem Prinzip des Domain-Driven-Design (DDD). Die folgende Abbildung zeigt die Architektur sowie eine detaillierte Beschreibung ihrer einzelnen Bestandteile.



Domain

- Independent npm-package
- Typescript
- Currently integrated in frontend, but will become backend

Infrastructure/Persistence

- Independent npm-package for demo data
- Typescript
- Currently quite empty, needs to provide access to blockchain

Frontend

- React/Typescript-App
- Currently serves as prototype demo
- Supposed to evolve to „real“ application



Monorepository – plan to manage with Rush

Abbildung 6: Architektur nach dem Prinzip des Domain-Driven-Design

Die System- und Prozesslandschaft basiert auf einer modernen, komponentenbasierten Architektur, die durch geeignete Modellierungssprachen und Technologien umgesetzt wird. Im Frontend kommt TypeScript als typsichere Programmiersprache zum Einsatz, kombiniert mit React als Bibliothek für die Entwicklung wiederverwendbarer UI-Komponenten. Diese Kombination ermöglicht eine skalierbare und wartungsfreundliche Benutzeroberfläche. Das Backend wird durch einen Express-Server auf Basis von Node.js realisiert. Die gesamte Backend-Logik ist ebenfalls in TypeScript entwickelt, um eine konsistente und robuste Codebasis zu gewährleisten. Auf der Blockchain-Seite basiert die Architektur auf der Ethereum-Blockchain. Smart Contracts werden in Solidity entwickelt, um dezentrale und unveränderbare Geschäftslogiken sicher und transparent abzubilden. Für das entwickelte Blockchain-Netzwerk wurden die Entwicklungs-Blockchain Ganache, welche das Ethereum-Netzwerk simuliert, und das Werkzeug Truffle-Suite ausgewählt.

4. Übersicht über zu beachtende Richtlinien bzgl. Datenschutz und Datensicherheit, Einhaltung des Datenschutzes und der Datensicherheit

Weiterhin erfolgte in diesem AP die Analyse der Prozessimplikationen auf die technische Umsetzung. Im Zuge dieser Analyse wurden hohe Anforderungen an Datenschutz und Datensicherheit konstatiert.

So wurde ein umfassendes Konzept zur Sicherstellung des Datenschutzes entwickelt. Dieses beinhaltet u.a. einen Passwortschutz für die UNICO-App. Zugriff auf die App erhalten lediglich Benutzer mit gültigem Passwort. Nach Aufruf der Anwendung erscheint die Aufforderung zur Eingabe des Benutzernamens (E-Mail-Adresse) sowie des Passworts. Nur bei gültiger Kombination wird eine Session eröffnet, mit der die Anwendung bedient werden kann. Bei Anmeldung wird der Session ein JWT (json web token) zugeordnet, welcher die Session eindeutig identifiziert. Das Token hat eine Gültigkeitszeit von 60 Minuten, d.h. nach einer Stunde ohne Aktivität wird eine Session beendet, der Nutzer wird auf die Anmeldeseite geleitet und muss sich erneut anmelden. Benutzername und Passwort werden dabei vom Administrator eingerichtet.

Weiterhin wurden über ein umfassendes Rollen- und Rechtekonzept Daten- und Funktionszugriffe entwickelt. In der Anwendung existieren die zwei Rollen Debitor und Kreditor. Jeder Benutzer ist dabei eindeutig einer dieser Rollen zugeordnet und entsprechende Daten werden je nach Rolle gefiltert. So sind sowohl für den Debitor als auch den Kreditor Kreditverträge sichtbar, jedoch jeweils mit den für die jeweilige Rolle relevanten Inhalten, insb. in Bezug auf den zum jeweiligen Zeitpunkt gültigen Status. Solange sich bspw. der Kredit im Status „Awaiting Collateral“ (Warte auf Collateral) befindet, wird dies dem Kreditor unter Unconfirmed Lendings durch den Hinweis „No Collateral“ bei dem entsprechenden Kredit angezeigt, wohingegen dem Debitor unter Unconfirmed Borrowings eine Funktion „Provide Collateral“ für den Kredit angeboten wird, über die er das Collateral bereitstellen kann. Die nachstehenden Abbildungen zeigen die unterschiedliche Ansicht des Kreditgebers (Abb. 7) und -nehmers (Abb. 8) in der Anwendung, wenn sich ein Kredit im Status „Awaiting Collateral“ befindet.

Unconfirmed Lendings

Active Unconfirmed

Car Loan

35.000,00 €

77.450,00 € incl. interest

To: S+T Versicherungen AG

Emission Date	Due Date
08 Jan 2025	08 Jun 2025

Interest	Interval	Loan Type
7%	Monthly	Term

No Collateral

Awaiting Collateral

Expansion Loan

220.000,00 €

226.600,00 € incl. interest

To: Computer Service Halle GmbH

Emission Date	Due Date
01 Dec 2021	01 Dec 2026

Interest	Interval	Loan Type
3%	Annually	Annuity

No Collateral

Awaiting Collateral

Abbildung 7: Status "Awaiting Collateral" aus Kreditorensicht

Unconfirmed Borrowings

Active Unconfirmed

Expansion Loan

220.000,00 €

226.600,00 € incl. interest

By: Finsol Bank S.A.

Emission Date	Due Date
01 Dec 2021	01 Dec 2026

Interest	Interval	Loan Type
3%	Annually	Annuity

No Collateral

Provide Collateral

Infrastructure Development

250.000,00 €

256.875,00 € incl. interest

By: splainK GmbH

Emission Date	Due Date
01 Jan 2023	01 Jan 2028

Interest	Interval	Loan Type
2.75%	Quarterly	Term

No Collateral

Provide Collateral

Abbildung 8: Status "Awaiting Collateral" aus Debitorensicht

Für einen Kreditoren sind in der Anwendung jederzeit alle Kredite, die er an verschiedene Debitoren ausgegeben hat, unter Active Lendings sichtbar (Abb. 9). Ein Debitor hingegen sieht alle Kredite, die er bei verschiedenen Kreditgebern aufgenommen hat unter Active Borrowings (Abb. 10).

Active Lendings

Active Unconfirmed

Warehouse Development Asset

300.000,00 €

306.000,00 € incl. interest

To: Computer Service Halle GmbH

Emission Date	Due Date
01 Jan 2022	31 Dec 2031

Interest	Interval	Loan Type
2%	Semi-Annually	Annuity

Paid	Collateral
83.122,97 €	1,50 ETH
	27.16%

Show Payments

Tech Expansion Loan

9.000,00 €

9.360,00 € incl. interest

To: Computer Service Halle GmbH

Emission Date	Due Date
01 Jun 2024	01 Dec 2024

Interest	Interval	Loan Type
4%	Weekly	Term

Paid	Collateral
5.040,00 €	1,50 ETH
	93.85%

Show Payments

Test Kredit 1

15.000,00 €

16.500,00 € incl. interest

To: Computer Service Halle GmbH

Emission Date	Due Date
08 Jan 2025	08 Jun 2025

Interest	Interval	Loan Type
10%	Monthly	Term

Paid	Collateral
0,00 €	4,00 ETH
	0.00%

Show Payments

Abbildung 9: Aktive Kredite aus Kreditorensicht

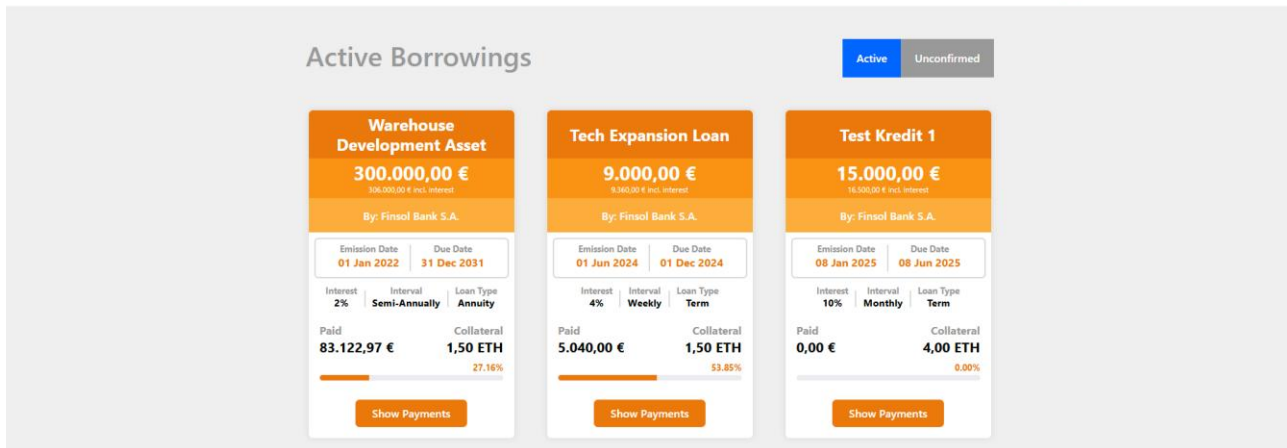


Abbildung 10: Aktive Kredite aus Debitorensicht

Die Ausführung von Funktionen wird pro Rolle geprüft, sodass bspw. die Funktion „Provide Collateral“ lediglich durch die Rolle des Debtors oder die Funktion „Confirm Collateral“ ausschließlich durch die Rolle des Kreditors ausgelöst werden kann.

Ferner wurden Protokollierungs- und Logging-Mechanismen umgesetzt, wobei der aktuelle Implementierungsstand die folgenden Mechanismen umfasst.

Blockchain-Logging:

Im aktuellen Testszenario wird Ganache als private Entwicklungs-Blockchain genutzt. Die Protokollierung ist so eingerichtet, dass alle JSON-RPC-Aufrufe in der Konsole festgehalten werden. Besonders relevant ist hierbei der Aufruf von `ethSendTransaction`. Dabei werden wesentliche Daten wie der Transaktions-Hash, der Gasverbrauch, die Blocknummer und die Blockzeit dokumentiert.

Datenbank-Logging:

Der Datenbank-Server ermöglicht die Protokollierung von SQL-Statements. Diese Funktion ist standardmäßig deaktiviert, kann jedoch bei Bedarf aktiviert werden.

API-Request-Logging im Backend:

Im Backend werden alle eingehenden API-Requests sowohl in der Konsole als auch in einer Datei protokolliert. Die erfassten Informationen umfassen den Zeitpunkt des Requests, das Loglevel (z. B. Info, Warning, Error), die angefragte URL und den HTTP-Response-Code. Über die Request-URL lässt sich nachvollziehen, welche Aktionen vom Frontend aus angestoßen wurden. Derzeit stehen detaillierte Informationen, wie beispielsweise Session-Daten oder spezifische Programmabläufe, noch nicht zur Verfügung.

Die aktuelle Implementierung der Protokollierungs- und Logging-Mechanismen ist prototypisch und soll zukünftig erweitert werden.

Ein weiterer Bestandteil der Anwendung im Hinblick auf Datensicherheit und Datenschutz ist die Einbindung von MetaMask als Krypto-Wallet. Diese dient als Zugang des Benutzers zu den Tokens auf der Blockchain, wobei jeder Benutzer über eine Account-ID auf der Blockchain (Wallet-Adresse) verfügt. Weiterhin verknüpft sie Blockchain-Accounts mit Token und signiert Transaktionen in Smart Contracts. Die Krypto-Wallet stellt sicher, dass keine Speicherung von privaten Schlüsseln durch die UNICO-App erfolgt. Private Schlüssel der MetaMask-Wallet werden ausschließlich lokal in MetaMask gespeichert, wodurch die UNICO-App zu keinem Zeitpunkt Zugriff auf private Schlüssel, Seed-Phrasen oder Wallet-Daten hat. Zum anderen ermöglicht die Einbindung von MetaMask eine transparente und sichere Transaktionsabwicklung, denn jede Transaktion wird von MetaMask zur Signatur vorgelegt. Da die Wallet vor jeder Signatur um ausdrückliche Bestätigung bittet, hat der Nutzer vollständige Kontrolle über jegliche Aktionen. Weiterhin gewährleistet die

Krypto-Wallet, dass keinerlei personenbezogene Daten in der UNICO-App gespeichert werden, da die Interaktion ausschließlich über die öffentliche Wallet-Adresse erfolgt, welche technisch notwendig ist, um Transaktionen mit der Blockchain durchzuführen.

Bezüglich des Datenschutzes und der Datensicherheit wurden gesetzliche und insb. regulatorische Anforderungen erhoben. Diese umfassen im Wesentlichen die Vorgaben der Datenschutzgrundverordnung (DSGVO), die Mindestanforderungen an das Risikomanagement (MaRisk) AT 4.2 sowie die Bankenaufsichtsrechtlichen Anforderungen an die IT (BAIT). Eine valide Umsetzung dieser Anforderungen fand im Rahmen des Projekts nicht statt. Gründe hierfür sind die damit verbundenen Kosten und zeitlichen Aufwendungen, welche Entwicklungsressourcen binden. Außerdem ist es das erklärte Ziel des Projekts, einen funktionsfähigen Demonstrator zu entwickeln. Die Entwicklung dieses Demonstrators trug allerdings der späteren Implementierung von Compliance-relevanten Themenstellungen Rechnung. So besitzt die Berücksichtigung dieser keinen Einfluss auf die Funktionsfähigkeit der Entwicklung.

Da der Prototyp lediglich im Rahmen einer Demonstratorumgebung entwickelt wurde, war die umfassende und finale Umsetzung der Vorgaben zu Datenschutz und Datensicherheit bisher nicht vonnöten. Eine entsprechende Vorbereitung des Demonstrators erfolgte jedoch bereits. Im künftigen Prototyp werden die Richtlinien zum Datenschutz und der Datensicherheit umfänglich berücksichtigt.

5. Prototyp dezentrales Credit Event-Register in Demonstratorumgebung

Im Ergebnis dieses Arbeitspakets steht folglich der Prototyp eines dezentralen Credit Event-Registers. Dieser Prototyp wird in einer Demonstratorumgebung bereitgestellt, d.h. in einer containerisierten Umgebung. Die Demonstratorumgebung wird in einem lokalen Netzwerk implementiert und ist nicht als öffentlich zugängliche Webplattform konzipiert.

Fazit

Die für AP 1 definierten Ziele konnten somit umfänglich erreicht werden. Lediglich im Hinblick auf die ursprünglich anvisiert Anwendungslandschaft, welche 3rd parties integriert, mussten Einschränkungen gemacht werden. Nichtsdestotrotz konnte anhand der tatsächlich realisierten Anwendungslandschaft die Machbarkeit der Pilotierung eines dezentralen Credit Event-Registers gezeigt werden.

2.1.2 AP 2: Repräsentation geeigneter Kreditsicherheiten als Security Token

Ziele

AP 2 war der Repräsentation geeigneter Kreditsicherheiten als Security Token gewidmet. Im Ergebnis dieses AP, welches unter Leitung der splainX GmbH durchgeführt wurde, sollten Aussagen zur Sicherheit bezüglich der Rechtsposition stehen. Weiterhin zielte das AP auf das Treffen einer Make- oder Buy-Entscheidung hinsichtlich der Tokenisierungslösung ab. Es diente ferner der Integration der Tokenisierungslösung im Demonstrator.

Ergebnisse

1. Sicherheit bezüglich der Rechtsposition

Der erste Schritt im Rahmen von Arbeitspaket 2 bestand in der Überprüfung der Rechtsposition von Collaterals. Es wurde untersucht, wie Collaterals als Token auf der Blockchain abgebildet werden können. Dabei zeigte sich, dass Collaterals, die bereits primär auf der Blockchain existieren, rechtlich sicher und bindend gestaltet werden können. So können Collaterals, wie bspw. die Namensschuldverschreibungen im sX-System oder Kryptowährungen, wie Ethereum- oder Bitcoin-Beträge, problemlos, rechtssicher und bindend als digitale Werte auf der Blockchain abgebildet werden. Herkömmliche Besicherungen, wie bspw. Fahrzeugscheine oder Grundbucheinträge, können hingegen lediglich als digitale Kopien existieren und sind durch die fehlende behördliche Anerkennung als digitale Repräsentation bzw. die mangelnde Rechtsanpassung für Blockchain-basierte Systeme rechtlich unsicher bzw. unverbindlich. Auf Basis der erlangten Erkenntnisse fiel deshalb die Entscheidung, im Demonstrator ausschließlich Kryptowährungen, insb. Ethereum, als Collateral zu verwenden. Die Verwendung von (tokenisierten) Garantieerklärungen ist jedoch ebenfalls möglich. Allerdings ist hierbei der Verwertungsprozess abweichend von jenem der Kryptowährungen.

2. Make- or Buy-Entscheidung hinsichtlich Tokenisierungslösung

Ferner sollte eine fundierte Make-or-Buy-Entscheidung bezüglich der Tokenisierungslösung getroffen werden. Dafür wurden verschiedene bereits etablierte Marktangebote eingehend analysiert, darunter bspw. die Plattform *Tokenize.it* von Christoph Jentsch, welche GmbH-Token zur Erfolgsbeteiligung erstellt. Weiterhin wurde auch die splainX-eigene Lösung zu Garantieerklärungen geprüft. Bitcoin-Kredit-Lösungen, wie bspw. angeboten von Nebeus, Binance, Nexo uvm., kamen während der Umsetzung des Projekts an den Markt. Deren Angebot stellt ein in sich geschlossenes Geschäftsmodell dar, dessen Anwendungen an Kryptowährungsinvestoren gerichtet ist. Die Analyse ergab, dass prinzipiell jede der geprüften Lösungen in das System eingebunden werden kann. Die Integration wäre jedoch mit zu hohen Kosten sowie einem übermäßig hohen technischen Aufwand verbunden gewesen, die für die Entwicklung eines Demonstrators nicht verhältnismäßig erschienen. Die Einbindung von Lösungen zu Bitcoin-Krediten war aufgrund des Adressatenkreises sowie der Abgeschlossenheit der Lösungen wenig sinnvoll. Auf Basis der gewonnenen Erkenntnisse und insb. aus Wirtschaftlichkeitsgründen wurde sich daher gegen die Nutzung einer bestehenden Lösung und für die Eigenentwicklung durch die Verwendung von Ethereum als Collateral entschieden. Perspektivisch muss die Entscheidung über die Nutzung einer bestehenden Lösung oder einer Eigenentwicklung jedoch stets im Einzelfall getroffen werden. Bspw. wäre im Falle einer geplanten Tokenisierung von GmbH-Anteilen eine Buy-Entscheidung sinnvoll, um bestehende Expertise und Lösungen zu nutzen.

3. Integration der Tokenisierungslösung im Demonstrator

Aufgrund der im vorangegangenen Schritt getroffenen Entscheidung zur Nutzung von Ethereum als Collateral entfiel die Integration einer Tokenisierungslösung im Demonstrator. Die Einbindung einer etablierten Lösung wäre wirtschaftlich nicht sinnvoll gewesen.

Fazit

Folglich lässt sich feststellen, dass die drei für AP 2 festgelegten Ziele vollständig erreicht wurden. Da eine Make-Entscheidung hinsichtlich der Tokenisierungslösung getroffen wurde, war die Integration einer existierenden Marktlösung nicht vonnöten.

2.1.3 AP 3: Pilotierung eines dezentralen Besicherungsregisters

Ziele

AP 3 wurde unter Leitung der splainX GmbH bearbeitet und umfasste die Pilotierung eines dezentralen Besicherungsregisters. Es zielte auf den semantischen Aufbau und die Erstellung von Transformationsschemata der Daten des dezentralen Besicherungsregisters ab. Ebenfalls beinhaltete das AP das Design der System- und Prozesslandschaft auf Basis passender Modellierungssprachen. Den Schwerpunkt bildete die Entwicklung des Prototyps des dezentralen Besicherungsregisters in der Demonstrator-Umgebung. Eine weitere zentrale Aufgabe dieses AP bestand in der Erstellung einer Übersicht über zu beachtende Richtlinien bezüglich des Datenschutzes und der Datensicherheit sowie der Sicherstellung der Einhaltung der diesbezüglich ermittelten Anforderungen.

Ergebnisse

1. Semantischer Aufbau und Transformationsschemata der Daten

Der semantische Aufbau des Datenmodells wurde bereits im Rahmen des Entity-Relationship-Modells (ERM) in Arbeitspaket 1 (Pilotierung des dezentralen Credit Event-Registers) umfassend betrachtet und wird daher an dieser Stelle nicht weiter vertieft. Die Transformationsschemata in diesem AP beziehen sich auf die Verarbeitung und Darstellung der Collateral-Daten auf der Blockchain und dienen der Überführung der Datentypen in eine geeignete Form für die dezentrale Speicherung und Nutzung. Für die Besicherungstransaktionen wurde eine spezielle Smart-Contract-Funktion entwickelt, die den Besicherungsbetrag in Ether direkt aus der Wallet des Nutzers abbucht und auf dem zu besichernden Token sperrt. Um eine benutzerfreundliche Darstellung zu gewährleisten, wird der hinterlegte Ether-Betrag aus dem Token ausgelesen und in der Benutzeroberfläche angezeigt. Dieses Transformationsschema stellt sicher, dass Besicherungswerte konsistent verarbeitet und im dezentralen Register korrekt abgebildet werden.

2. System- und Prozesslandschaft auf Basis passender Modellierungssprachen

Im Rahmen dieses AP wurde dieselbe System- und Prozesslandschaft wie für das Credit Event-Register genutzt. Aus diesem Grund wird an dieser Stelle nicht näher darauf eingegangen.

3. Übersicht über zu beachtende Richtlinien bzgl. Datenschutz und Datensicherheit, Einhaltung des Datenschutzes und der Datensicherheit

Auf Datenschutz und Datensicherheit wurde bereits in AP 1 detailliert eingegangen.

4. Prototyp dezentrales Besicherungsregister in Demonstrator-Umgebung

Der Prototyp des dezentralen Besicherungsregisters wurde in einer Demonstrator-Umgebung umgesetzt und basiert auf einer technischen Architektur, in der Besicherungen durch auf der Blockchain gespeicherte Token repräsentiert werden. Jedes besicherte Finanzinstrument ist mit einem Token verknüpft, der den hinterlegten Collateral-Betrag widerspiegelt. So stellt beispielsweise der Umstand, dass ein Kredit mit 4 Ether besichert ist, ein entsprechendes Token dar. Die Gesamtheit dieser Tokens bildet das Besicherungsregister.

Die Benutzeroberfläche wurde so gestaltet, dass das Collateral stets im Kontext des besicherten Finanzinstruments angezeigt wird. Da als Collateral eine Kryptowährung – in diesem Fall Ether – genutzt wird, besteht die Darstellung der Besicherung darin, dass für jeden Kredit das zugehörige Collateral sichtbar ist. Dies ermöglicht eine transparente und intuitive Übersicht über die hinterlegten Sicherheiten.

Die nachstehende Abbildung illustriert die Transformation eines Kredits vom Anfang bis zum bereitgestellten Collateral.

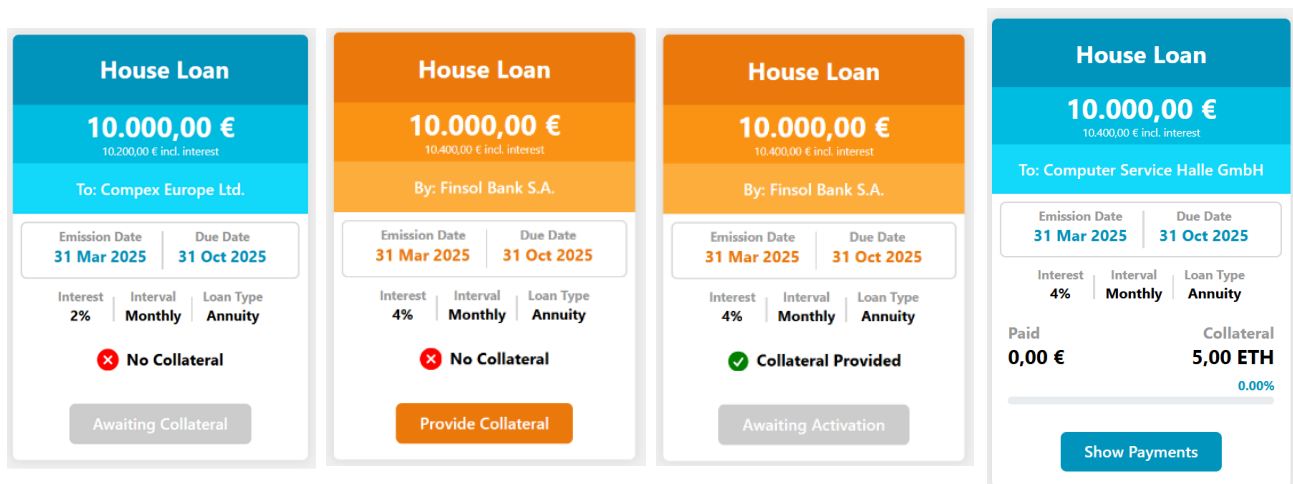


Abbildung 11: Prozess zur Bereitstellung eines Collaterals

Fazit

Für AP 3 konnten somit ebenfalls alle definierten Ziele erreicht werden.

2.1.4 AP 4: Pilotierung eines dezentralen Protokolls zur Initiierung der Inanspruchnahme bzw. Verwertung von Kreditsicherheiten

Ziele

Die Leitung von AP 4, welches der Pilotierung eines dezentralen Protokolls zur Initiierung der Inanspruchnahme bzw. Verwertung von Kreditsicherheiten gewidmet war, oblag der splainX GmbH. In diesem AP sollte der semantische Aufbau und die Erstellung von Transformationsschemata der Daten vorgenommen werden. Das Design der System- und Prozesslandschaft auf Basis passender Modellierungssprachen stand ebenfalls im Fokus. Des Weiteren sollte eine Übersicht über zu beachtende Richtlinien bzgl. Datenschutz und Datensicherheit erstellt sowie die Einhaltung der diesbezüglich ermittelten Anforderungen sichergestellt werden. In

diesem AP bildete die Entwicklung eines Prototyps für das dezentrale Protokoll zur Initiierung der Inanspruchnahme bzw. Verwertung von Kreditsicherheiten in der Demonstrator-Umgebung einen Schwerpunkt.

Ergebnisse

1. Semantischer Aufbau und Transformationsschemata der Daten

Während der semantische Aufbau bereits anhand des ERM-Modells in Arbeitspaket 1 behandelt und definiert wurde, lag der Schwerpunkt dieses Arbeitsschritts auf der praktischen Umsetzung der Datenverarbeitung und der Absicherung von Transaktionen, insbesondere im Umgang mit Zahlungsausfällen und der Anbindung externer Schnittstellen.

Wenn festgestellt wird, dass eine Zahlung fehlgeschlagen ist, greift eine speziell entwickelte Smart-Contract-Funktion, die das hinterlegte Collateral automatisch in die Wallet des Gläubigers überträgt. Für die Anbindung dieses Prozesses an externe Zahlungssysteme war eine detaillierte Analyse relevanter Bankschnittstellen erforderlich. Dabei wurden zwei Ansätze eingehend geprüft: zum einen die Deutsche Bank API, die in einer Sandbox-Umgebung validiert wurde, und zum anderen eine EBICS-Protokoll-Implementierung für Node.js. Der technische Aufwand der Implementierung wurde jedoch als zu hoch für den vorgesehenen Anwendungsfall bewertet. Da echte Zahlungstransaktionen im Rahmen der Entwicklung aus praktischen Gründen nicht direkt genutzt werden konnten, lag der Fokus darauf, gemeinsame Strukturen der verschiedenen Schnittstellen zu identifizieren und die spezifischen Anforderungen für die Anwendung abzuleiten. Auf dieser Basis wurde ein Abstraktionslayer konzipiert, das es ermöglicht, unterschiedliche Bankschnittstellen flexibel anzubinden, ohne das Backend der Anwendung anpassen zu müssen. Nachfolgende Abbildung visualisiert den Payment Service mit den abstrahierten Bankenschnittstellen.

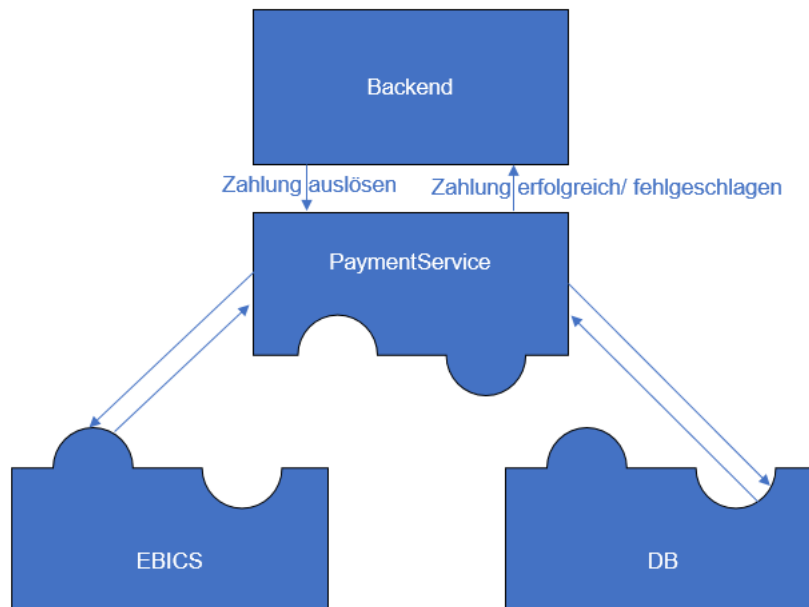


Abbildung 12: Payment Service mit abstrahierten Bankenschnittstellen

2. System- und Prozesslandschaft auf Basis passender Modellierungssprachen

Innerhalb von AP 4 wurde auf dieselbe System- und Prozesslandschaft wie für das Credit Event-Register zurückgegriffen.

3. Übersicht über zu beachtende Richtlinien bzgl. Datenschutz und Datensicherheit, Einhaltung des Datenschutzes und der Datensicherheit

Auf die Richtlinien und die Einhaltung des Datenschutzes und der Datensicherheit wurde bereits eingehend innerhalb der Erläuterungen zu AP 1 eingegangen, weshalb an dieser Stelle auf detaillierte Ausführung verzichtet wird.

4. Prototyp dezentrales Protokoll zur Initiierung der Inanspruchnahme bzw. Verwertung von Kreditsicherheiten in Demonstrator-Umgebung

Um den Prozess der Inanspruchnahme und Verwertung von Kreditsicherheiten in einer dezentralen Architektur prototypisch abzubilden, wurden die relevanten Bankschnittstellen abstrahiert und eine eigene Banksimulation entwickelt und implementiert. Zentrales Element dieser Architektur ist der entwickelte PaymentService, der als integraler Bestandteil des Backends fungiert. Der PaymentService übernimmt die Aufgabe, Zahlungsvorgänge auszuführen und den beteiligten Systemkomponenten den Status der Transaktion – Erfolg oder Fehlschlag – zurückzumelden.

Für die Demonstrator-Umgebung wurde ein Simulator erstellt, der auf Anweisung des PaymentService Zahlungen simuliert. Der PaymentService übermittelt dabei den Befehl an den Simulator, eine Zahlung auszulösen. Der Simulator entscheidet daraufhin, ob der Zahlungsvorgang als erfolgreich oder fehlgeschlagen gilt und meldet das Ergebnis entsprechend zurück.

Ergänzend dazu wurden mit EBICS und der DB API zwei reale Schnittstellen als Softwarebestandteile integriert. Über eine flexible Steuerung kann im System ausgewählt werden, ob der PaymentService mit EBICS, der DB API oder dem Simulator arbeiten soll. Diese Komponenten sind modular aufgebaut und können wie Puzzleteile ausgetauscht werden. Dadurch lässt sich der Prototyp sowohl mit simulierten als auch mit realen Zahlungsprozessen betreiben, ohne Änderungen an der zentralen Backend-Logik vornehmen zu müssen. Die nachstehende Abbildung illustriert den modularen Aufbau.

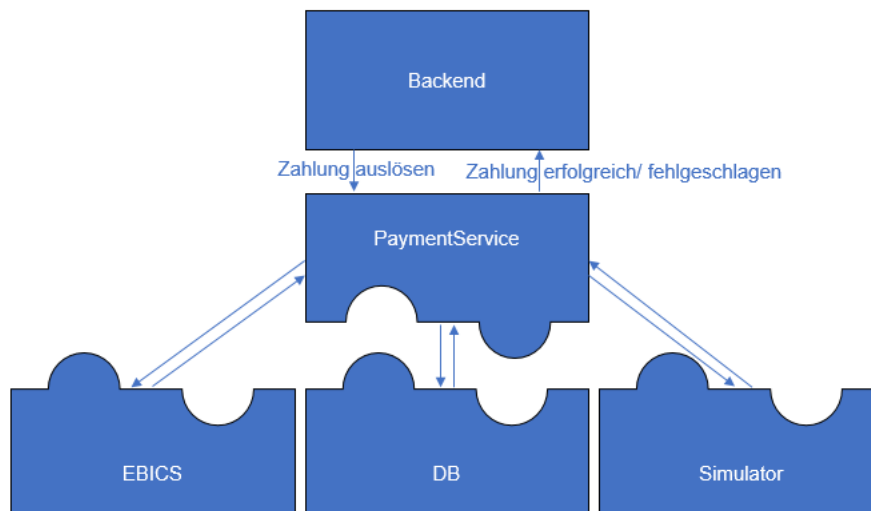


Abbildung 13: Payment Service mit abstrahierten Bankschnittstellen und Simulator

Fazit

Die für AP 4 festgelegten Ziele konnten somit ebenfalls umfänglich erreicht werden.

2.1.5 AP 5: Aufbau eines Demonstrators

Ziele

AP 5 war dem Aufbau des Demonstrators unter Leitung der splainX GmbH gewidmet. Als Ergebnis von AP 5 sollte die vorbereitete Instanz des integrierten Systems für den Produktivbetrieb entstehen. Darüber hinaus diente das AP der Administration, dem Management und dem Monitoring der Pilotierung sowie der Entstehung gepflegter Nutzerprofile und Zugangsberechtigungen. Im Fokus standen zudem der Testbetrieb inkl. Einführungskonzept des Blockchain-Netzwerks, die Definition quantitativer und qualitativer Merkmale zur Bewertung des Systems sowie die Durchführung einer anforderungsbezogenen Fehleranalyse und -korrektur der Smart Contracts. Ein weiteres Ziel dieses AP bestand in der Gewährleistung der Funktionsfähigkeit, des Datenschutzes und der Datensicherheit.

Ergebnisse

Gemäß dem zugrunde liegenden Software-Architekturkonzept wurde die Anwendung in mehrere eigenständige Node.js-Packages untergliedert:

- **model:** beinhaltet die Definition der zentralen Datenobjekte und Interfaces.
- **domain:** implementiert die Geschäftslogik durch entsprechende Service-Klassen.
- **persistence:** stellt Repository-Klassen bereit, um Daten sowohl in der Datenbank als auch auf der Blockchain zu speichern.
- **backend:** organisiert Router, Controller und Validatoren für den Betrieb des Backend-Servers und die Bereitstellung der API.
- **frontend:** umfasst React-Komponenten für die Darstellung der Inhalte sowie Services zur Kommunikation mit der Backend-API.

Im Backend werden domänenspezifische API-Dienste zur Verfügung gestellt, die im Demonstrator über definierte Endpunkte nutzbar sind. Für die Benutzeroberfläche kamen verschiedene React-Komponenten zum Einsatz, welche die Anwendungslogik visualisieren. Die gesamte Entwicklung orientierte sich an der Test-Driven-Development-(TDD)-Methode: Vor der Implementierung der jeweiligen Komponenten wurden zunächst passende Tests erstellt. Dadurch ist sichergestellt, dass sämtliche Bestandteile der Anwendung bei jedem Build-Prozess automatisiert durch Unit-Tests überprüft werden.

1. Vorbereitete Instanz des integrierten Systems für den Produktivbetrieb

Da der Demonstrator bereits eine hohe Nähe zur späteren Produktivumgebung aufwies, wurden im Rahmen der Vorbereitung auf den Produktivbetrieb für die zentralen Systembausteine – Frontend, Backend und Blockchain – jeweils eigenständige Container-Images erstellt. Diese Images sind so konzipiert, dass sie in einer orchestrierten Umgebung (z. B. mittels Kubernetes) bei einem Cloud-Anbieter betrieben werden können. Hinsichtlich der Blockchain-Topologie wurde dabei im Projekt UNICO eine Ethereum-kompatible Blockchain gewählt. Für die Entwicklung und Demonstration kamen Ganache als Blockchain-Simulation sowie die Truffle-Suite zum Kompilieren, Deployen und Testen der Smart Contracts zum Einsatz. Sowohl das simulierte Blockchain-Netzwerk als auch die Truffle-Suite wurden ebenfalls als Docker-Container bereitgestellt und in die Gesamtarchitektur integriert. Damit wurde sichergestellt, dass das Gesamtsystem modular deploybar ist und die einzelnen Komponenten im Zusammenspiel innerhalb eines Netzwerks orchestriert werden können. Die erstellten Container-Images kamen auch im Demonstrator zum Einsatz, um eine möglichst realistische Betriebsumgebung zu simulieren und die Übergangsfähigkeit in den Produktivbetrieb zu gewährleisten.

2. Administration und Management inkl. Monitoring der Pilotierung

Im Rahmen des Projekts wurde ein Register (i.S. einer Übersicht) entworfen, das den unterschiedlichen Anforderungen der jeweiligen Anwenderrollen gerecht wird. Zu Beginn stand dabei die Definition der jeweiligen Berechtigungen im Fokus, um festzulegen, welche Rollen Zugriff auf welche Ansichten und Daten erhalten. Eine besondere Herausforderung stellte die Gewährleistung der Datenkonsistenz dar, da Informationen sowohl in einer dezentralen Blockchain als auch parallel in einer internen Datenbank gespeichert werden. Dies machte eine genaue Abstimmung zur Synchronisation und Konsistenzsicherung erforderlich. Zusätzlich wurde eine Klassifizierung der Daten vorgenommen, da nicht alle Transaktionsdetails für jede Nutzerrolle einsehbar sein dürfen. Die Entwicklung und Umsetzung des Registers wurden durch kontinuierliche Anwendungstests begleitet, aus denen wertvolle Rückmeldungen für die Weiterentwicklung gewonnen wurden.

Zur Absicherung der Stabilität und Funktionalität des Pilotsystems kam ein testgetriebener Entwicklungsansatz zum Einsatz. In diesem Rahmen wurden sämtliche Software-Bestandteile regelmäßig geprüft und durch automatisierte Tests abgesichert. Ergänzend dazu wurden umfassende Integrationstests entwickelt, um die fehlerfreie Zusammenarbeit aller Systemkomponenten sicherzustellen. Die Überwachung und Auswertung der Testergebnisse erfolgen zentral über eine CI/CD-Pipeline auf Basis von GitLab (siehe nachfolgendes Beispielbild).

















 Passed 00:04:04	Add mock functionality #1581977529 UN-60-implement-mock-functionality-for-failed-and-successful-payments 136aa1d4		 
 Passed 00:04:05	Merge branch 'task/UN-57-store-collateral-am... #1581975931 UN-60-implement-mock-functionality-for-failed-and-successful-payments 8c6230e6		 
 Passed 00:03:51	Remove PaymentEntriesScheduleService.test #1456817745 story/UN-58-create-payment-plan-and-provide-api-endpoint-for-loan-repayment 7de23c96		 
 Failed 00:03:50	Fix calculations and tests #1456756214 story/UN-58-create-payment-plan-and-provide-api-endpoint-for-loan-repayment ffb731df		 

Abbildung 14: Auszug aus der CI/CD-Pipeline

Zudem wurden ein strukturiertes Release-Management sowie eine Versionskontrolle eingeführt. Hierfür kommt ein Git-Repository zum Einsatz, das unterschiedliche Branches für verschiedene Entwicklungszwecke bereitstellt. So bildet der Branch Main die jeweils aktuell veröffentlichte Version ab, während der Branch Develop den momentanen Entwicklungsstand repräsentiert. Für die Umsetzung neuer Funktionen werden separate Feature-Branche verwendet. Darüber hinaus unterstützen Release-Branche die Vorbereitung von neuen Versionen, und Hotfix-Branche dienen der schnellen Behebung von Fehlern in bereits ausgelieferten Releases. Als Branching-Strategie wird das Git-Flow-Modell eingesetzt, welches durch die klare Trennung der Entwicklungszweige den Ablauf in der agilen Softwareentwicklung effizient strukturiert und die Nachvollziehbarkeit des Projektfortschritts direkt im Code verbessert. Sobald eine neue Version freigegeben wird, erfolgt eine Kennzeichnung des entsprechenden Quelltextstands im Repository durch die Vergabe einer Versionsnummer. Diese Markierung löst automatisch einen Build-Prozess innerhalb der CI/CD-Pipeline aus, wodurch eine neue Version generiert und anschließend auf der vorgesehenen Cloud-Infrastruktur bereitgestellt wird.

3. Gepflegte Nutzerprofile und Zugangsberechtigungen

Ein weiteres Ziel dieses Arbeitspakets war die Implementierung strukturierter Benutzerprofile mit den entsprechenden Zugriffsrechten. Dafür wurden in der Anwendung drei verschiedene Profile erstellt: jeweils eines für die beiden Hauptrollen sowie ein weiteres für die Administratorfunktion. Jeder Benutzer ist einer spezifischen Rolle zugeordnet, wodurch der Zugriff auf Daten und Funktionen entsprechend gefiltert wird. Der Debitor kann Kreditverträge einsehen und erhält im Status „Awaiting Collateral“ die Möglichkeit, über die Funktion „Provide Collateral“ Sicherheiten bereitzustellen. Der Kreditor hingegen sieht in diesem Status den Hinweis „No Collateral“ unter seinen Unconfirmed Lendings und kann nach Bereitstellung durch den Debitor die Funktion „Confirm Collateral“ ausführen. Die Anwendung stellt sicher, dass Funktionen nur von den jeweils berechtigten Rollen ausgelöst werden können, sodass eine klare Trennung der Verantwortlichkeiten gewährleistet ist.

Der Zugriff auf die Anwendung erfolgt über die Anmeldung mit Benutzername und Passwort. Nach einem erfolgreichen Login generiert das Backend ein JSON Web Token (JWT) als Session-Token und sendet es an das Frontend. Bei jeder Anfrage vom Frontend an das Backend wird dieses Session-Token mitgesendet. Das Backend überprüft dabei, ob das Token dem angemeldeten Benutzer zugeordnet ist und ob die Sitzung noch aktiv ist. Falls ein ungültiges Token übermittelt wird, wird der Benutzer automatisch auf die Login-Seite weitergeleitet.

4. Testbetrieb inkl. Einführungskonzept des Blockchain-Netzwerks

Für den Testbetrieb des Blockchain-Netzwerks wurde eine Container-Umgebung so konzipiert, dass die systemtechnische Architektur mit minimalem Installationsaufwand lokal ausgeführt werden kann. Zur Veranschaulichung verschiedener Testszenarien wurden Demo-Nutzer erstellt, die den verschiedenen Rollen entsprechen (Kreditor: Charles, Debitor: Daniel). Darüber hinaus wurden die Anwendung und ihre Funktionen in unterschiedlichen Kontexten demonstriert, mit potenziellen Nutzern diskutiert und validiert.

Während der Projektlaufzeit wurde noch kein detailliertes Einführungskonzept ausgearbeitet, da der Fokus auf der Entwicklung eines Demonstrators lag und nicht auf dem späteren produktiven Einsatz. Die Erstellung eines umfassenden Einführungskonzepts erfolgt im Anschluss an das Projekt in Zusammenarbeit mit potenziellen Nutzern und Partnern. Dennoch wurden bereits während der Projektphase konkrete Überlegungen angestellt, die in das zukünftige Konzept einfließen werden, darunter die Ablösung von Ganache sowie der Aufbau einer privaten Blockchain mit GETH.

5. Quantitative und qualitative Merkmale zur Bewertung des Systems

Zur Bewertung der Systemleistung wurden für sämtliche Funktionsbereiche der Anwendung automatisierte Tests entwickelt. Eine neue Version wird nur bereitgestellt, wenn alle Tests erfolgreich durchlaufen wurden. Das Testmanagement wurde bereits im Rahmen von Ergebnis 2 dieses Arbeitspakets beschrieben. Zusätzlich werden die Antwortzeiten von Serveranfragen erfasst und hinsichtlich ihrer Qualität bewertet. Wird eine vordefinierte Antwortzeit überschritten, erfolgt eine Warnmeldung.

Obwohl innerhalb des Projekts UNICO bislang keine spezifischen Lasttests zur Messung von Anfragedauern durchgeführt wurden, ist deren Umsetzung in der Einführungsphase geplant. Regelmäßige qualitative Bewertungen des Systems fanden bereits mit potenziellen Nutzern statt, deren Feedback kontinuierlich in die Weiterentwicklung der Anwendung einfluss und zur Validierung der Systemqualität beitrug.

6. Anforderungsbezogene Fehleranalyse und -korrektur der Smart Contracts

Zur Identifikation und Behebung von Fehlern in den entwickelten Smart Contracts wurden umfassende Integrationstests konzipiert und umgesetzt. Diese Tests erfolgen automatisiert und prüfen das Zusammenspiel verschiedener Komponenten der Anwendung. Jede Änderung am Quellcode führt zu einer direkten Rückmeldung über die Einhaltung der definierten Anforderungen durch den Smart Contract und alle relevanten Systemkomponenten. Falls Tests fehlschlagen, kann unmittelbar mit der Fehleranalyse begonnen und notwendige Korrekturen vorgenommen werden.

7. Funktionsfähigkeit, Datenschutz und Datensicherheit ist gewährleistet

Aspekte des Datenschutzes und der Datensicherheit wurden im Rahmen des Anforderungsmanagements berücksichtigt. Durch das umfassende Testmanagement – siehe dazu 2. Administration und Management inkl. Monitoring der Pilotierung – wird sichergestellt, dass sowohl die funktionalen Anforderungen der Anwendung erfüllt als auch datenschutzrechtliche Vorgaben eingehalten werden. Auf diese Weise wird gewährleistet, dass der Schutz sensibler Daten und die sichere Verarbeitung innerhalb der Anwendung durchgängig sichergestellt sind.

Fazit

Die sieben für AP 5 definierten Ziele konnten somit fast vollständig erreicht werden. Wie beschrieben sind für eine künftige produktive Nutzung der Anwendung weiterführende Entwicklungen und Überlegungen sowie die Implementierung von Lasttests vonnöten.

2.1.6 AP 6: Sicherstellung der Nachhaltigkeit

Ziele

AP 6 wurde unter Leitung der splainX GmbH durchgeführt und diente der Sicherstellung der Nachhaltigkeit. Es zielte auf die vollständige Dokumentation der programmierten Softwarekomponenten und der Systemlandschaft sowie auf die Dokumentation und Abstraktion der Forschungsergebnisse ab. Des Weiteren sollten Ergebnispublikationen als wissenschaftliche Veröffentlichungen und Ergebnispräsentationen auf öffentlichen Veranstaltungen vorgenommen werden. Die Einbindung der Ergebnisse in Forschung und Lehre sollte gleichermaßen erfolgen.

Ergebnisse

1. Vollständige Dokumentation der programmierten Softwarekomponenten

2. Vollständige Dokumentation der Systemlandschaft

Die Dokumentation der entwickelten Softwarekomponenten wurde maßgeblich über User Stories in Jira realisiert. Diese enthalten nicht nur detaillierte Prozessbeschreibungen, sondern auch die hinterlegten Implementierungsdetails, die den Entwicklungsprozess nachvollziehbar machen. In den User Stories wird beschrieben, welche Komponente entwickelt werden musste und wie diese mit externen Systemen, beispielsweise über API-Requests, interagiert.

Eine separate Quelltextdokumentation wurde bewusst vermieden. Stattdessen wurde die Codequalität im Rahmen der Qualitätssicherung über regelmäßige Reviews sichergestellt. Ziel war es, den Quellcode so zu schreiben, dass die Abläufe verständlich und klar sind. Dieses Vorgehen orientierte sich an den Prinzipien aus Robert C. Martins "Clean Code – A Handbook of Agile Software Craftsmanship" (2008), in dem es heißt:

„The code is clean if it can be understood easily by everyone on the team. Clean code can be read and enhanced by a developer other than its original author.“

Der Fokus lag somit darauf, einen gut strukturierten Code zu schaffen, anstatt umfangreiche Kommentare oder separate Dokumentationen anzufertigen. Dies erwies sich als effizientere Herangehensweise im Entwicklungsprozess.

Zusätzlich wurde die Systemlandschaft umfassend dokumentiert, insbesondere im Hinblick auf die Container-Umgebung. Diese Dokumentation umfasst Installationsanweisungen sowie Konfigurationsdateien, welche die Struktur der Docker-Services als zentralen Bestandteil der Systemarchitektur beschreiben. Dadurch wurde die Nachvollziehbarkeit der technischen Infrastruktur sichergestellt, was es dem Team erleichterte, Anpassungen vorzunehmen und eventuelle Probleme effizient zu beheben.

3. Dokumentation und Abstraktion der Forschungsergebnisse

Die im Projekt erzielten Ergebnisse wurden für eine Weiterverwendung in folgenden, auf Blockchain basierenden Projekten abstrahiert und aufbereitet. In diesem Zusammenhang sind allgemeine Vorgehensweisen dokumentiert worden, die sich insb. auf den Payment Trigger (siehe AP 1: Pilotierung dezentrales Credit Event-Register) beziehen. Es war möglich, die Schnittstellen so weit zu verallgemeinern, dass sie für andere Blockchain-Projekte, welche Zahlungsströme zwischen Fiat-Geld und Kryptogeld involvieren, wiederverwendet werden können.

4. Ergebnispublikationen als wiss. Veröffentlichungen, öffentliche Veranstaltungen

Die Ergebnispublikation des Projekts UNICO erfolgt durch die Veröffentlichung der Teile I und II des Sachberichts bei der Technischen Informationsbibliothek (TIB) Hannover. Weiterhin wurden die Ergebnisse zum De-

monstratortag der Blockchain-Schaufensterregion Mittweida am 13.09.2024 präsentiert (Mitschnitt der Präsentation abrufbar unter: [UNICO: Tokenisierung und Abwicklung von Kreditsicherheiten | Demonstratortag 2024](#)).

5. Einbindung der Ergebnisse in Forschung und Lehre

Im Rahmen des Projektes UNICO übernahm die Hochschule Mittweida überwiegend die Einbindung der Ergebnisse in Forschung und Lehre. Ebenso fließen sie in das WEB3-Modul und die Finanzmanagementausbildung an der Westsächsischen Hochschule Zwickau sowie in bereits laufende und künftige Forschungsprojekte ein.

Fazit

Die für AP 6 festgelegten Ziele konnten ebenfalls folglich umfänglich erreicht werden.

2.1.7 AP 7: Projektkoordination und Öffentlichkeitsarbeit

Ziele

Die Projektkoordination und Öffentlichkeitsarbeit wurde kontinuierlich und fortlaufend unter Leitung der splainX GmbH vorgenommen. Im Ergebnis von AP 7 sollten Informationsmaterialien für die beteiligten Akteure bspw. i.F.v. Protokollen und Präsentationen, der Aufbau eines internen und externen Berichtswesens bezüglich der Projekt- und Arbeitsergebnisse sowie die fachwissenschaftliche Aufarbeitung und Veröffentlichung der Projektergebnisse stehen.

Ergebnisse

1. Informationsmaterial für die beteiligten Akteure

Während der Projektlaufzeit wurden regelmäßig Fortschrittsüberprüfungen angestellt. Hierzu wurden auf Arbeitsebene mit weiteren Partnern aus der WIR!-Verbundregion, im Besonderen der Volksbank Mittweida eG., Arbeitsgespräche zur Anwendbarkeit und technischen Umsetzung geführt. In Vorbereitung auf diese Treffen, wie auch bspw. den Workshop zur Geschäftsmodellentwicklung Blockchain Competence Center Mittweida (BCCM), wurden Informationsmaterialien zu Projektfortschritt und -ergebnissen in Form von Präsentationen aufbereitet. Der jeweils aktuelle Entwicklungsstand wurde anhand des Demonstrators präsentiert. Des Weiteren wurden Zwischenergebnisse an dem sogenannten *Demonstratortag* und dem internen WIR!-Projekttreffen im September 2024 der interessierten Öffentlichkeit vorgestellt. Weiterhin wurden Erkenntnisse aus dem Projekt in Präsentationsunterlagen für Investoren eingearbeitet. Im wissenschaftlichen Bereich bot das Projekt eine Diskussionsgrundlage, um mit Studenten der WHZ alternative Wege einer Working Capital-Finanzierung zu diskutieren.

2. Internes und externes Berichtswesen bzgl. der Projekt- und Arbeitsergebnisse

Dem Projektträger gegenüber wurden die Arbeitsergebnisse anhand der Zwischenberichte und des vorliegenden Abschlussberichtes kommuniziert. Zu den aktuellen Projektergebnissen wurde zudem regelmäßig im Rahmen der Gesellschafterversammlungen der splainX berichtet.

Die Produktentwicklung ist der internen Perspektive zugeordnet und erfolgte auf Basis agiler Arbeitsweisen unter Anwendung der Scrum-Methodik. Es fanden in der Regel alle zwei Wochen sogenannte Sprints statt. In diese Sprints wurden schrittweise extern erhobene Anforderungen integriert. Gleichzeitig wurden dabei häufig technische Abhängigkeiten identifiziert.

Die Umsetzung der erfassten Anforderungen erfolgte üblicherweise bis zum darauffolgenden Sprint, wo sie vom Product Owner geprüft und abgenommen wurden. Parallel dazu wurden beim nächsten Sprint neue Anforderungen für den übernächsten Sprint aufgenommen. Die erzielten Ergebnisse wurden, wie zuvor beschrieben, im Netzwerk der Kooperationspartner präsentiert und analysiert. Anpassungen flossen anschließend als neue Anforderungen erneut in den Sprint-Prozess ein.

Die Sprints bildeten somit die Basis für die interne, überwiegend technische Berichterstattung. Die zeitliche Steuerung orientierte sich an den im Projektantrag festgelegten Meilensteinen, die im Rahmen eines Meilenstein-Controllings überprüft wurden.

3. Fachwissenschaftliche Aufarbeitung und Veröffentlichung der Projektergebnisse

Sowohl die fachwissenschaftlichen Projektergebnisse als auch das aufgebaute personelle Know-How fließen in die Entwicklung aktueller und zukünftiger Projekte ein. Eine Veröffentlichung der Projektergebnisse erfolgt über die Publizierung des Teil I/II des Sachberichts bei der Technischen Informationsbibliothek Hannover. Darüber hinaus wurden und werden die Ergebnisse in Form von Vorträgen und Praktika in den Schwerpunkt Finanzmanagement des Bachelor-Studiengangs Management sowie das WEB3-Modul an der Westsächsischen Hochschule Zwickau integriert. Wissenstransfer fand im Besonderen innerhalb der Blockchain-Schaufensterregion Mittweida statt. splainX lieferte wichtige Beiträge zur Implementierung Blockchain-getriggelter Zahlungen auf Bankkonten. Auch bot das Projekt UNICO Inspiration für weitere Start-ups, Geschäftsmodelle aus der Hinterlegung von Krypto Assets zu entwickeln.

Fazit

Die für AP 7 formulierten Ziele wurden somit umfassend erreicht.

2.2 Wichtigste Positionen des zahlenmäßigen Nachweises

Die Ausgaben von EUR 272.747,47 für Personalkosten in der Position *0812 Beschäftigte E12-E15* dienen der Anstellung von Software-Entwicklern zur Durchführung des Vorhabens. Die Summe unterschreitet die gemäß Zuwendungsbescheid bewilligten Gesamtausgaben dieser Position, welche sich auf EUR 338.119,20 belaufen. Wie beabsichtigt wurden während der gesamten Projektlaufzeit vier Beschäftigte der Position 0812 eingestellt, wovon Herr Kramer als Senior Developer und gleichzeitig als Geschäftsführer agierte. Aufgrund geringerer Qualifikation konnte die geplante Gehaltshöhe für die drei weiteren Mitarbeiter jedoch nicht gerechtfertigt werden und fiel entsprechend niedriger als vorgesehen aus. Insbesondere konnte die Einstellung eines zweiten Senior Developers mit mehrjähriger, einschlägiger Berufserfahrung in leitender Funktion im Bereich der Software-Entwicklung zur Übernahme von Aufgaben mit besonderer Schwierigkeit aufgrund des in der Branche und Region vorherrschenden Fachkräftemangels nicht erfolgen.

Zur Erreichung der Projektziele unterstützten jedoch zwei geringfügig Beschäftigte das Projekt in der Position *0820 Lohnempfänger(innen) MArb*. Die Ausgaben dieser Position belaufen sich auf EUR 20.499,12 und unterschritten somit die bewilligten Gesamtausgaben von EUR 30.888,00. Dies liegt darin begründet, dass trotz des frühzeitigen Beginns der Suche nach geeignetem Personal statt den drei geplanten Stellen bedauerlicherweise nur zwei besetzt werden konnten.

Die Position 0835 diente der Vergabe von Aufträgen für rechtliche Einordnungen und Stellungnahmen. Obwohl für diese Position EUR 36.000 bewilligt worden waren, wurden keine entsprechenden Aufträge ausgelöst, sodass keine Ausgaben anfielen. Da im Rahmen des Projektes eine Beschränkung auf ausgewählte Kreditsicherheiten vorgenommen wurde, war die Beauftragung rechtlicher Stellungnahmen nicht mehr nötig, sodass aus Wirtschaftlichkeitsgründen darauf verzichtet wurde. Ferner konnte auf einige Erkenntnisse bezüglich rechtlicher Grundlagen aus dem Projekt KryWert zurückgegriffen werden.

Die Position 0846 Inlandsreisen, für welche EUR 4.226,00 bewilligt worden waren, blieb unberührt, da aufgrund der zeitlichen Verfügbarkeit der Teilnehmer Online-Zusammenkünfte präferiert wurden. Des Weiteren wurden aufgrund der Komplexität Abstimmungen und Diskussionsrunden zu rechtlichen Themenstellungen obsolet. So konnten die Aufwendungen für Reisetätigkeiten eingespart werden.

Im Rahmen der Position 0850 "Gegenstände und andere Investitionen" wurden lediglich EUR 2.221,26 für den Erwerb eines einzigen Notebooks ausgegeben, obwohl ursprünglich EUR 10.000 bewilligt waren. Diese Entscheidung fußte darauf, dass sich die vorhandenen, wenn auch veralteten, Notebooks der splainX GmbH für die Anforderungen des gesamten Projekts als ausreichend erwiesen. Im Sinne einer verantwortungsvollen Liquiditätsplanung wurde daher auf den Kauf weiterer Geräte verzichtet.

Folglich unterschritten die tatsächlich angefallenen Gesamtausgaben i.H.v. EUR 295.467,85 die bewilligten Ausgaben von EUR 419.233,20.

2.3 Notwendigkeit und Angemessenheit der geleisteten Projektarbeiten

Alle durchgeführten Arbeiten im Rahmen des Projektes UNICO einschließlich der dafür aufgewandten Ressourcen waren sowohl notwendig als auch angemessen, um die im Arbeitsplan formulierten Aufgaben erfolgreich bearbeiten und die Projektziele erreichen zu können. Insbesondere zur Durchführung der anspruchsvollen Software-Entwicklungsaufgaben bedurfte es qualifizierten und erfahrenen Fachpersonals, für welches der Großteil der Ausgaben aufgebracht wurde. Ohne die fachliche Expertise der Software-Entwickler wäre der erfolgreiche Aufbau eines prototypischen Demonstrators einer Systemlandschaft, in dem Emittenten oder Inhaber von Debt Token diese unabhängig von einem Tokenisierungsnetzwerk (Emissionsplattform) nach Bedarf mit Kreditsicherheiten ausstatten können, fehlgeschlagen. Die wesentlichen Herausforderungen in den Bereichen Tokenisierung von Kreditsicherheiten, die Entwicklung eines dezentralen Credit Event-Registers, wenn auch nur für Zahlungsausfälle, sowie die Konzeption und Implementierung von Smart Contracts konnten somit zielgerichtet und praxisnah bearbeitet werden. Die geleisteten Projektarbeiten sind damit vor dem Hintergrund des Ergebnisses als angemessen und notwendig zu bewerten.

2.4 Voraussichtlicher Nutzen und Verwertbarkeit des Ergebnisses

Dem Grundprinzip des UNICO-Projekts gingen Angebote zur Kreditausreichung gegen die Hinterlegung von Kryptowährungen wie Bitcoin als Sicherheit (Collateralized Crypto Loans) voraus. Hierbei boten Anbieter wie Salt Lending und BlockFi Kredite in Fiat-Währungen gegen Krypto-Collateral an. Die Services richteten sich an Krypto-Investoren, die ihre illiquiden Krypto Assets ohne Auflösung ihrer Positionen finanziell nutzbar machen wollten. Die Kreditvergabe wurde meist durch Smart Contracts oder zentralisierte Verwahrung abgesichert. Dieses Prinzip sollte im UNICO-Projekt genutzt werden, um die Fungibilität weiterer illiquider Vermögenswerte oder von Kreditleihgeschäften (wie Garantien) zu erhöhen. Mit diesem Gedanken befand sich splainX in der Beantragungsphase des Projekts UNICO in einem noch recht unerschlossenen Markt. In der Umsetzungsphase kamen weitere Angebote auf: die Nutzung von (virtuellen) GmbH-Anteilen oder von tokenisierten Vermögenswerten, wie Aktien oder Edelmetallansprüchen, für die Vergabe von Krediten. Diese Modelle zielen auf eine Koppelung der Kreditvergabe gegen Sicherheiten ab. Der Ansatz von splainX im Projekt UNICO besitzt einen höheren Abstraktionsgrad, da sich die Lösung nicht explizit auf Kryptowährungen bezieht und unabhängig von der Art der Kreditsicherheit gestaltet. Im Vordergrund steht im UNICO-Projekt eher die Effiziente Abwicklung des Finanzierungsprozesses: von der Ausreichung des Fremdfinanzierungsinstruments, der Verknüpfung dessen mit (tokenisierten) Kreditsicherheiten bis zur Rückzahlung des Fremdfinanzierungsinstruments und wenn diese scheitert bis zur Verwertung der eingeräumten Kreditsicherheiten. Mit diesem Ansatz wird mittels des UNICO-Projekts versucht, den gesamten Kreditvergabe- und -abwicklungsprozess zwischen Schuldner und Gläubiger mithilfe der DLT zu automatisieren.

Die gewonnenen Ergebnisse lassen sich auf eine Vielzahl von Finanzierungssituationen übertragen. Insb. wenn es sich um Asset-basierte Finanzierungen, wie bspw. bei kapitalmarktorientierten Finanzierungen mit Unterlegung eines Pools an Zahlungsansprüchen, handelt, verspricht diese Vorgehensweise durch den hohen Automatisierungsgrad erhebliche Potenziale zur Steigerung der Effizienz. Die Anwendung in den Bereichen Finanzierung über Asset Backed Securities, Mortgage Backed Securities, Collateralized Debt Obligations oder Asset Backed Commercial Papers erscheint daher vielversprechend. Eine Umsetzung ist derzeit in Prüfung. splainX bietet hierfür ein Tokenization-as-a-Service-Modell an. Die Gesellschaft erhält dabei eine Provision i.H.v. ca. 0,5 bis 0,75% der Forderungswerte. splainX fungiert weiters als Payment Initiation Service Provider (PISP) und Account Information Service Provider (AISP).

Herausforderungen werden in der Anbindung der Systeme von Forderungsverkäufern (Originators) gesehen. splainX muss für die Automatisierung der Prozesse geeignete Schnittstellen entwickeln und zur Verfügung stellen. Hinsichtlich Authentifikation/KYC und Wallet-Integration sind weitere Entwicklungen der Anwendung vonnöten. Die Aufnahme des Regelbetriebs kann voraussichtlich ab Q3/2026, regulatorisch voraussichtlich jedoch erst ab ca. Q1/2027 erfolgen. Zudem hat das Unternehmen bereits konkrete Anwender in Aussicht.

Demnach lässt sich feststellen, dass aus dem Projekt UNICO ein konkretes Geschäftsmodell für die Weiterführung und Nutzung der Anwendung für die splainX GmbH hervorging.

2.5 Fortschritt bei anderen Stellen

Während der Projektlaufzeit sind keine Fortschritte auf dem Gebiet des Vorhabens bei anderen Stellen bekannt geworden.

2.6 Veröffentlichungen

Neben der Veröffentlichung der Teile I und II des Sachberichts bei der Technischen Informationsbibliothek (TIB) Hannover ist die Publikation von Ergebnissen bislang weder erfolgt noch künftig vorgesehen. Die Projektergebnisse fließen jedoch bereits in das WEB3-Modul sowie die Finanzmanagement-Ausbildung an der Westsächsischen Hochschule Zwickau ein. Weiterhin wurden die Ergebnisse zum Demonstratortag der Blockchain-Schaufensterregion Mittweida am 13.09.2024 (Mitschnitt der Präsentation abrufbar unter: [UNICO: Tokenisierung und Abwicklung von Kreditsicherheiten | Demonstratortag 2024](#)) präsentiert.