



ECOPLAN GmbH
Justus-Liebig Str. 6
36093 Künzell

HESSEN



Hessische Staatskanzlei

Hessische Ministerin für
Digitale Strategie und Entwicklung

Machbarkeitsstudie zur Digitalisierung

Projekttitle: Speicherung sensibler Informationen mit Hilfe
der Blockchain-Technologie am Beispiel von CRM-Systemen

Stand: Juni 2021

Autor: Dr. Jochen Eckard

Impressum

Machbarkeitsstudie der ECOPLAN GmbH im Rahmen des Distr@I-Förderprogramms des Landes Hessen

Projektbeteiligte:

ECOPLAN GmbH
Justus-Liebig Str. 6
36093 Künzell

www.ecoplan-crm.de

Die technische Umsetzung innerhalb der ECOPLAN GmbH erfolgte unter Leitung des technischen Geschäftsführer Marcel Gerk in einem Team mit vier weiteren Informatikern.

Die wirtschaftliche, organisatorische und unternehmerische Begleitung des Projektes wurde durch den Geschäftsführer/Gesellschafter Roland Vollmer übernommen und sichergestellt.

Die ECOPLAN GmbH ist ein etablierter Softwareanbieter für Verbände, Kammern, Fundraiser und Non-Profit-Organisationen mit mehr als 25 Mitarbeitern. Seit der Gründung 1993 sorgt das Team aus Betriebswirten und Informatikern für Effizienz und Erfolg bei Digitalisierung von Geschäftsabläufen in Organisationen. Die Kunden von ECOPLAN nutzen CRM-Software zur Bearbeitung einer Vielzahl von Geschäftsvorfällen. Hierzu zählen u.a. die Verwaltung von Adressen und Mitgliederdaten, die Organisation von Veranstaltungen und Gremien, die Durchführung der Beitragsabrechnungen, die analoge und digitale Kommunikation, die Verwaltung von Projekten, Verträgen, Spenden und vieles mehr. Im Rahmen von Softwareeinführungs- und Softwarebetreuungsprojekten werden Module, Funktionen und kundenindividuelle Anpassungen entwickelt, damit die Mitarbeiter einer Mitgliederorganisation ihr Tagesgeschäft möglichst transparent und effizient in einer integrierten Software abwickeln können. Die Software wird in der Regel in bestehende Systemlandschaften integriert, z.B. Schnittstellen zu Portalen, Buchhaltungssoftware, etc.

Leiter/Autor der Studie: Dr. Jochen Eckard

Inhaltsverzeichnis

Abbildungsverzeichnis	4
1 Management Summary	6
2 Einführung / Ausgangslage	7
3 Zielsetzung und Methodik	7
4 Andere Studien	9
5 Durchführung der Machbarkeitsstudie	9
5.1 Experteninterviews	9
5.2 Prüfung rechtlicher Anforderungen	13
5.3 Tiefere Analyse Zielgruppe / Märkte	17
5.4 Vorbereitung der Teststellungen	18
5.4.1 Auswahl von zwei CRM-Systemen, Installation, Testdaten	18
5.4.2 Identifikation potentieller Blockchain-Modelle / Technologiefilter	19
5.4.3 Grundtypen der Blockchain-Modelle	21
5.4.4 Große bekannte Blockchain-Modelle / Vorselektion	23
5.4.5 Vergleich der Eigenschaftsausprägungen vorselektierter Blockchain-Modelle	26
5.4.6 Auswahl Blockchain-Modell für das Projektvorhaben	31
5.4.7 Hyperledger Fabric – Technische Details	32
5.5 Installation, Test Blockchain-Modell	38
5.6 Konzeption Prototypen-Schnittstellen-Logik	47
5.7 Entwicklung Prototypen	49
5.7.1 Entwicklung, Test Prototypen Schnittstellen-Logik (3 Sprints)	49
5.7.2 Testläufe Datenspeicherung	55
5.7.3 Evaluation Prototypen Leistung und Sicherheit	59
5.7.4 Anpassung Prototypen erneuter Testlauf	81
5.8 Roll-out im Testunternehmen - Bundesverband deutscher Banken (BdB), Berlin	82
5.8.1 Identifikation Prototypen und Onboarding im Testunternehmen	82
5.8.2 Roll-out und Integration der technischen Ansätze im Testunternehmen	82
5.8.3 Test Prototyp im Testunternehmen	83
5.9 Gruppendiskussion Testteilnehmer	86
5.9.1 Zusammenfassung der Ergebnisse / Reflexion	86
5.10 Ableitung Ressourcenbedarf, Aufwand, Kosten, Nutzen	87
6 Fazit und Ausblick	90
7 Literaturverzeichnis	93
8 Anlage	96

Abbildungsverzeichnis

Abbildung 1: Grundbausteine der Blockchain-Technologie	20
Abbildung 2: Mögliche Ausprägungsvarianten einer Blockchain.....	23
Abbildung 3: Vergleich ausgewählter Blockchain-Modelle.....	27
Abbildung 4: Komponenten von Hyperledger Fabric	32
Abbildung 5: Hyperleder Fabric Infrastruktur.....	33
Abbildung 6: Speicherung privater Daten in Hyperledger Fabric	37
Abbildung 7: Konsensablauf von Hyperledger Fabric.....	38
Abbildung 8: Beispiel-Code für die Erstellung eines Docker-Compose File.....	40
Abbildung 9: Übersicht der laufenden Docker-Container.....	41
Abbildung 10: Beispiel für eine Chaincode-Implementierung	42
Abbildung 11: Auszug aus einer Private Data Collection.....	43
Abbildung 12: Hyperledger Fabric auf einem Verbund eigener Rechner/Server.....	44
Abbildung 13: Konfiguration der IBM Public Cloud Hyperledger Fabric.....	45
Abbildung 14: Auszug aus der IBM Public Cloud Block-History für Hyperledger Fabric	45
Abbildung 15: Schnittstellen-Logik über die native Einbindung im CRM-Client.....	47
Abbildung 16: Schnittstellen-Logik über Tomcat-Webservice.....	48
Abbildung 17: Beispiel für die konfigurierten Felder in Dynamics 365	50
Abbildung 18: Verlegung des Tomcat-Dienstes in den Docker-Container	51
Abbildung 19: Verschlüsselung durch Schlüssel aus CA Fabric.....	52
Abbildung 20: Verschlüsselung durch externe Public Key Infrastruktur	53
Abbildung 21: Beispiel einer CRM-Eingabemaske	55
Abbildung 22: Teststellung für erste Testläufe	56
Abbildung 23: Ergebnisse Latenzen des Caliper-Messtools – Feldwert (1 KB)	57
Abbildung 24: Verlagerung der Endorser-Peers und Tomcat-Dienstes auf den Server	58
Abbildung 25: Teststellung für die Evaluierung	60
Abbildung 26: Liniengraphik - Feldwert schreiben ins Ledger (1KB)	62
Abbildung 27: Übersicht Latenzen - Feldwert schreiben ins Ledger (1KB).....	62
Abbildung 28: Boxplot – Feldwert schreiben ins Ledger (1KB).....	62
Abbildung 29: Liniengraphik - Feldwert lesen aus Ledger (1KB).....	63
Abbildung 30: Übersicht Latenzen - Feldwert lesen aus Ledger (1KB).....	63
Abbildung 31: Boxplot – Feldwert lesen aus Ledger (1KB).....	63
Abbildung 32: Liniengraphik - Feldwert schreiben in Private Data Bereich (1KB).....	64
Abbildung 33: Übersicht Latenzen - Feldwert schreiben in Private Data (1KB)	64
Abbildung 34: Boxplot – Feldwert schreiben in Private Data Bereich (1KB)	64
Abbildung 35: Liniengraphik - Feldwert lesen aus Private Data Bereich (1KB).....	65
Abbildung 36: Übersicht Latenzen - Feldwert lesen aus Private Data (1KB)	65
Abbildung 37: Boxplot – Feldwert lesen aus Private Data Bereich (1KB)	65
Abbildung 38: Liniengraphik - Dokument schreiben ins Ledger (400KB).....	66
Abbildung 39: Übersicht Latenzen – Dokument schreiben in Private Data (400KB)	66
Abbildung 40: Boxplot – Dokument schreiben ins Ledger (400KB)	66
Abbildung 41: Liniengraphik - Dokument lesen aus Ledger (400KB).....	67
Abbildung 42: Übersicht Latenzen – Dokument lesen aus Ledger (400KB)	67
Abbildung 43: Boxplot – Dokument lesen aus Ledger (400KB)	67
Abbildung 44: Liniengraphik - Dokument schreiben in Private Data (400KB)	68
Abbildung 45: Übersicht Latenzen – Dokument schreiben in Private Data (400KB)	68
Abbildung 46: Boxplot – Dokument schreiben in Private Data (400KB).....	68
Abbildung 47: Liniengraphik - Dokument lesen aus Private Data (400KB)	69
Abbildung 48: Übersicht Latenzen – Dokument lesen aus Private Data (400KB)	69
Abbildung 49: Boxplot – Dokument lesen aus Private Data (400KB).....	69
Abbildung 50: Liniengraphik - Dokument schreiben ins Ledger (1.77MB)	70
Abbildung 51: Übersicht Latenzen – Dokument schreiben ins Ledger (1.77MB)	70
Abbildung 52: Boxplot – Dokument schreiben ins Ledger (1.77MB).....	70
Abbildung 53: Liniengraphik - Dokument lesen aus Ledger (1.77MB).....	71
Abbildung 54: Übersicht Latenzen – Dokument lesen aus Ledger (1.77MB).....	71

Abbildung 55: Boxplot - Dokument lesen aus Ledger (1.77MB)	71
Abbildung 56: Liniengraphik - Dokument schreiben in Private Data (1.77MB).....	72
Abbildung 57: Übersicht Latenzen – Dokument schreiben in Private Data (1.77MB).....	72
Abbildung 58: Boxplot - Dokument schreiben in Private Data (1.77MB)	72
Abbildung 59: Liniengraphik - Dokument lesen aus Private Data (1.77MB).....	73
Abbildung 60: Übersicht Latenzen – Dokument lesen aus Private Data (1.77MB).....	73
Abbildung 61: Boxplot - Dokument lesen aus Private Data (1.77MB).....	73
Abbildung 62: Liniengraphik - Zusammenfassung Latenzen Feldwert (1KB).....	74
Abbildung 63: Liniengraphik - Zusammenfassung Latenzen Dokument (400KB)	74
Abbildung 64: Liniengraphik - Zusammenfassung Latenzen Dokument (1.77MB).....	74
Abbildung 65: Ergebnisse der Latenz-Messung im Überblick.....	75
Abbildung 66: Beispiel - Technische Infrastruktur des entwickelten Modellansatzes	77
Abbildung 67: Teststellung Bundesverband deutscher Banken (BdB).....	83
Abbildung 68: CRM-Eingabemaske Firmierung von Kontaktdaten	84
Abbildung 69: CRM-Eingabemaske „Gremienbeteiligung“	84
Abbildung 70. Dokumentenspeicherung aus CRM-System	85
Abbildung 71: Durchschnittliche Latenzen des BdB-Testlaufs.....	85

1 Management Summary

Aufgrund der stark wachsenden Datenbestände der letzten Jahre ist das Verwalten und Speichern von Daten zu einer entscheidenden Fragestellung für Unternehmen und anderen Organisationen geworden. Sie müssen sich neben ihrem Kerngeschäft verstärkt mit der richtigen IT-Infrastruktur auseinandersetzen, um reibungslose Prozesse zu gewährleisten und die Daten sicher zu speichern. Aus wirtschaftlichen und rechtlichen Gründen ist ein Teil dieser Daten vor unberechtigten Einblicken oder Änderungen besonders zu schützen. Mit dem geplanten Vorhaben soll eine (herstellerunabhängige) Software-Schnittstellen-Logik in Verbindung mit der Blockchain-Technologie entwickelt werden, um den Anteil sensibler Informationen aus CRM-Systemen von kleinen und mittleren Unternehmen (KMU) sowie anderen Organisationen „sicher“ und „nachvollziehbar“ (Änderungshistorie) bei vertretbarem Kostenrahmen speichern zu können. Der Lösungsansatz soll eine Alternative zu den existierenden Modellen, wie z.B. der Speicherung in zentralen Datenbanken oder der Speicherung in einem kostenintensiven „Storage-System“, darstellen. Verschiedene technische Ansätze sollen prototypisch auf einem rudimentären Niveau konzipiert, entwickelt und evaluiert werden. Erfolg und Wirksamkeit der Ansätze sind abzuschätzen sowie nach Umsetzbarkeit zu klassifizieren. Neben der Klärung der Blockchain-Variante (Private/Public Blockchain, Permissioned oder Permissionless) stellen die Anforderungen der Datenschutzgrundverordnung (DSGVO) beim Speichern personenbezogener Daten auf ein Blockchain-Netzwerk eine besondere Herausforderung dar. Über die Wahl eines geeigneten Konsensmechanismus sollen die Transaktions- bzw. Energiekosten niedrig gehalten werden. Darüber hinaus soll in Bezug auf die Anforderungen geklärt werden, wo die Blockchain aufzusetzen ist: Auf einem Verbund eigener Rechner und Server (Inhouse) oder extern bezogen als Dienstleistung (Cloud)? Eine umfassende Sicherheitsbetrachtung des Gesamtansatzes soll Aufschluss über die möglichen Schwachstellen bzw. Angriffsvektoren aufzeigen. Bei der Konzeption und Entwicklung der Schnittstellen-Logik liegt der Schwerpunkt, neben den Kriterien wie einfache Installation und Wartbarkeit, vor allem auf der weitgehenden Herstellerunabhängigkeit bei der Schnittstellen-Anbindung. Somit wird zu klären sein, welche konkrete Ausgestaltung die zu entwickelnde Schnittstellen-Logik und das dazugehörige Blockchain-Modell haben müsste, damit sie erfolgreich in die Praxis überführt werden können und langfristig einsatzfähig ist. Die Machbarkeitsstudie soll dazu dienen, eine fundierte Grundlage für die Ausarbeitung des Projektkonzeptes zu liefern, indem Voraussetzungen, Chancen und Risiken geklärt werden. Hierdurch soll die Wirksamkeit des Projekts erhöht und Fehlinvestitionen frühzeitig vermieden werden. Am Ende der Studie soll der positive oder negative Machbarkeitsnachweis als Ergebnis vorliegen. Ein vergleichbares Vorgehen/Produkt ist am Markt für KMU-Unternehmen und andere Organisationen nicht zu finden. Zur Blockchain-Technik gibt es gegenwärtig eher vielzählige Visionen, Theorien und Konzepte als real existierende, funktionierende Anwendungen.

2 Einführung / Ausgangslage

Das Verwalten und Speichern der stetig zunehmenden Datenmengen stellt für Unternehmen und andere Organisationen eine große Herausforderung dar. Um die Daten sicher zu speichern, müssen sich die IT-Verantwortlichen mit der richtigen IT-Infrastruktur auseinandersetzen. Aus wirtschaftlichen und rechtlichen Gründen sind sensible Daten besonders zu schützen. Daten mit Personenbezug unterliegen beim Verarbeiten und Speichern zusätzlich den Anforderungen der Datenschutzgrundverordnung (DSGVO). Gewöhnlich erfolgt die Speicherung der Daten bei Unternehmen und anderen Organisationen auf eigenen Rechnern und Servern (Inhouse) oder extern auf gemieteten Cloud-Servern eines spezialisierten IT-Dienstleisters. Eine Mischform beider Varianten ist in der Praxis ebenfalls häufig zu finden. Die Speicherung der Daten in einer sogenannten Cloud ist noch eine recht junge Entwicklung und wird häufig als Paradigmenwechsel bezeichnet. Cloud-Dienstleister ermöglichen Unternehmen und anderen Organisationen ihre IT-Infrastruktur komplett auszulagern. Mittlerweile gibt es viele Anbieter, die sich auf unterschiedliche Cloud-Lösungen spezialisiert haben und versichern, dass der Datenschutz wie auch die Datensicherheit umfassend gewährleistet sei. Die sichere Speicherung von Daten über einen eigenen Netzwerkserver (Inhouse) stellt sich ebenfalls problematisch dar. Mit dem aktuellen Stand der Technik gestaltet sich der Verschlüsselungsprozess für Daten auf dem eigenen Netzwerkserver eher komplex und aufwändig. In der Regel findet zwar eine Verschlüsselung statt, aber nur auf dem Weg (Kommunikation) zur Festplatte. Dort werden die Daten jedoch im Standard unverschlüsselt gespeichert. Zur Verschlüsselung der Daten auf der Festplatte wird ein aufwändiges Schlüsselmanagement benötigt. Diese Implementierung und Wartung ist keine leichte Aufgabe, da die benötigten Verfahren hierzu auf sehr unterschiedlichen Ebenen der Informationstechnik (Software-/Hardware-Verschlüsselung) ansetzen und in der Regel, ohne entsprechende Hardwareimplementierung, mit spürbaren Performanceeinbußen zu rechnen ist.¹ Die Cloud-Dienstleister bieten mittlerweile verbesserte Features im Bereich des Sicherheitsmanagements an. Je nach Sicherheitsstufe steigen aber die Kosten dafür schnell in die Höhe. Gerade kleine und mittlere Unternehmen (KMU) und andere Organisationen haben oft nur ein begrenztes IT-Budget, müssen aber alle Anforderungen an die Datensicherheit erfüllen. Mit der zunehmenden Datenmenge und dem Voranschreiten der Digitalisierung der letzten Jahre nimmt auch die Diskussion über ökologische Aspekte und die Frage nach Nachhaltigkeit zu. Welche Auswirkungen bringt die Datenvermehrung und die dazugehörige Verarbeitung, z.B. in Bezug auf den Energie- oder Rohstoffverbrauch mit sich? Es stellt sich die Frage, wie die Datenverarbeitung auszugestalten ist, um ökologische und nachhaltige Anforderungen zu erfüllen.²

An der zuvor dargestellten Situation lässt sich erkennen, dass es im Bereich der sicheren Datenspeicherung erkennbare Schwächen für Unternehmen und Organisationen gibt.

3 Zielsetzung und Methodik

Die Machbarkeitsstudie soll dazu dienen, eine fundierte Grundlage für die Ausarbeitung des Projektkonzeptes zu liefern, indem Voraussetzungen, Chancen und Risiken geklärt werden. Hierdurch soll die Wirksamkeit des Projekts erhöht und Fehlinvestitionen frühzeitig vermieden werden. Die anvisierten zwei technischen Hauptkomponenten umfassen die „Schnittstellen-Logik“ (Programmcode) und die „Blockchain“. Die Schnittstellen-Logik soll dabei die Kommunikation zwischen dem CRM-System und der Blockchain übernehmen.

¹ Goldhammer, K. et al. (2018). Kompass IT-Verschlüsselung. Orientierungs- u. Entscheidungshilfen für KMU zum Einsatz von Verschlüsselungslösungen. Studie im Auftrag des BMWi. In: https://www.bmwi.de/Redaktion/DE/Publikationen/Studien/kompass-it-verschluesselung.pdf?__blob=publicationFile (Zugriff: 16.08.2020)

² Bundesministerium für Bildung und Forschung (BMBF). Digitalisierung und Nachhaltigkeit. In: <https://www.bmbf.de/de/digitalisierung-und-nachhaltigkeit-10466.html> (Zugriff: 18.11.2020)

Neben der tieferen Analyse der Zielgruppen und Märkte soll die Akzeptanz der Zielgruppe im Hinblick auf das Projektergebnis untersucht werden. Es sollen verschiedene technische Ansätze prototypisch auf einem rudimentären Niveau konzipiert, entwickelt und evaluiert werden. Erfolg und Wirksamkeit der Ansätze sind abzuschätzen sowie nach Umsetzbarkeit zu klassifizieren. Zentral dabei ist die Frage, welche konkrete Ausgestaltung die zu entwickelnde Software-Schnittstellen-Logik und das dazugehörige Blockchain-Modell haben müsste, damit sie erfolgreich in die Praxis überführt werden können.

Das Bundesamt für Sicherheit in der Informationstechnik (BSI) verweist hierbei besonders auf das zu berücksichtigende Spannungsverhältnis zwischen IT-Sicherheit, Effizienz und dem Designziel der Blockchain.³

Folgende technische Leitfragen standen verstärkt im Vordergrund:

- Welche Blockchain-Variante soll zum Einsatz kommen? Eine Private/Public Blockchain, Permissioned oder Permissionless?
- Wo soll die Blockchain aufgesetzt werden: Auf einem Verbund eigener Rechner/Server oder extern bezogen als Dienstleistung (Cloud)?
- Was sind die Vor- und Nachteile der Blockchain-Varianten bezogen auf das Vorhaben?
- Wie ist der technische Ansatz abzubilden, damit die Datenkommunikation mit der Blockchain für den Anwender performant verläuft (Latenzen/Transaktionszeiten)?
- Welches Blockchain-Modell und welcher kryptographische Ansatz ist zu wählen, um eine Langzeitsicherheit zu gewährleisten?
- Wo ist die Schnittstellen-Logik technisch zu implementieren: Auf Datenbankebene, auf Clientseite oder als Webservice?
- Wie ist die Schnittstellen-Logik technisch auszugestalten, damit sie sich einfach und wartungsarm in die bestehende IT-Systemumgebung der Zielgruppe praktikabel integrieren lässt?
- Wie ist die Schnittstellen-Logik zu entwickeln damit sie die definierten Funktionen erfüllt und nicht auf ein System bezogen, sondern relativ herstellerunabhängig einsetzbar ist?
- Wie gestaltet sich der Integrations- und Wartungsaufwand für die Zielgruppe?

In Bezug auf ökonomische, ökologische und rechtliche Belange waren folgende Fragen zu klären:

- Wie ist der gesamte Ansatz auszugestalten, damit der Bezugspreis für die Zielgruppe marktgerecht bleibt?
- Wie ist das Konsensmodell der Blockchain auszugestalten, damit die Transaktions- bzw. Energiekosten niedrig ausfallen?
- Welche Ressourcen und welches Know-how werden benötigt und welche Kosten sind damit verbunden?
- Wie ist die zeitliche Abfolge der erforderlichen Arbeitsschritte für das gesamte Vorhaben?
- Wie sehen die rechtlichen Aspekte, z.B. in Bezug auf datenschutzrechtliche Bestimmungen (DSGVO) und Einhaltung von Schutz- und Markenrechten, aus?

Als Ergebnis der Machbarkeitsstudie soll abschließend der positive oder negative Machbarkeitsnachweis für das vorliegende Projektvorhaben aufgezeigt werden.

Das Design der Untersuchung basierte auf einem triangulären Methodenmix aus Tests und Simulationen der entwickelten technischen Ansätze sowie qualitativen Verfahren in Form von leitfadengestützten Experteninterviews und einer Gruppendiskussion. Über ein exploratives Prototyping wurden verschiedene technische Lösungsansätze entwickelt und evaluiert. Die technischen Ansätze wurden dabei nur rudimentär entwickelt, d.h. auf einem einfachen

³ Berghoff, C. et al. (2019). Blockchain sicher gestalten, Bundesamt für Sicherheit in der Informationstechnik (BSI) S. 15 ff.. In: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Krypto/Blockchain_Analyse.pdf?__blob=publicationFile&v=5 (Zugriff: 16.08.2020)

technischen Niveau mit dem Ziel der Überprüfung der Funktionstüchtigkeit. Mit Hilfe von ausgewählten Metriken wurden die Ansätze in verschiedenen Kategorien bewertet.

4 Andere Studien

Für das vorliegende Projektvorhaben, der Speicherung sensibler Daten aus einem CRM-System auf ein Blockchain-Netzwerk, wurden keine direkt vergleichbaren externen Studien gefunden. An vielen Stellen in Literatur und dem World Wide Web wird empfohlen, aus Gründen der Sicherheit, Performance und dem Datenschutz, sensible Daten nicht auf einer Blockchain zu speichern⁴, sondern lediglich deren Hashwerte. Projektansätze, die dieses Vorgehen der „Hashwert-Speicherung“, abbilden sind in Mehrzahl zu finden. Unter dieser Prämisse weist der vorliegende Projektansatz/Studie einen gewissen Abgrenzungscharakter zu den bestehenden Studien/Projektansätzen auf. Zu erwähnen ist, dass etwa zeitgleich mit Beginn dieses Projektvorhabens/Studie vom US-Verteidigungsministerium⁵ ein ähnliches Projekt in Auftrag gegeben wurde, bei dem sensible Forschungsdaten auf eine Blockchain gesichert werden sollen.

5 Durchführung der Machbarkeitsstudie

Um einen Überblick über die Kenntnisse verantwortlicher Stellen in Unternehmen zur vorliegenden Technologie zu erhalten, wurden zunächst mehrere Experteninterviews durchgeführt und ausgewertet.

5.1 Experteninterviews

Im Mittelpunkt der Experteninterviews stand die Einbindung des spezifischen Expertenwissens von Akteuren aus dem IT-Admin Umfeld, da auf diese Personengruppe bei der möglichen Einführung der technischen Ansätze in Unternehmen oder sonstigen Organisationen auch die Hauptumsetzung in Form von Implementierungs- und Wartungsarbeiten zukommt. Um einen Zugang zu diesem Erfahrungswissen zu erlangen, fiel die Wahl der Erhebungsmethode auf die Form des Experteninterviews. Eingesetzt wurde die Form eines offenen, leitfadengestützten Interviews in dessen Zentrum das Expertenwissen stand. Meuser und Nagel konstatieren in diesem Zusammenhang, dass es sich beim Experteninterview methodologisch betrachtet eher um ein „randständiges Verfahren“ (Meuser & Nagel 2010)⁶ handelt, obgleich seine Verwendung in der Forschungspraxis weit verbreitet ist. Der Einsatz des Experteninterviews erfolgte in der vorliegenden Studie triangulär im Rahmen der Methodenkombination, um zusätzliche Informationen zum Forschungsthema zu gewinnen. Über den explorativ-erschließenden Charakter sollten zusätzliche Informationen über die Erhebung von Deutungswissen des Experten in Form von Regeln, Sichtweisen und Interpretationen zum vorliegenden Studienthema gewonnen werden.

Der Aufbau und Inhalt des Interviewleitfadens und die Auswahl der Experten wurde kollektiv in der ECOPLAN-Projektgruppe getroffen. Die Auswahl der Experten stützte sich auf folgende Leitfragen:

⁴ Naceur, M.R.B. (2018). Blockchain – Ein Dilemma für den Datenschutz. In: <https://www.computerwoche.de/a/blockchain-ein-dilemma-fuer-den-datenschutz,3545513> (Zugriff: 05.10.2020)

⁵ US-Verteidigungsministerium verwendet Blockchain, um sensible Daten zu speichern. In: <https://www.ledgerinsights.com/us-department-of-defense-blockchain-secure-sensitive-data/> (Zugriff: 08.10.2020)

⁶ Meuser & Nagel (2010). S. 457

1. Trägt das spezifische Expertenwissen des Interviewpartners dazu bei, Aussagen zur technischen Ausgestaltung der Ansätze zu erhalten?
2. Hat die Arbeit des Akteurs einen hohen Stellenwert für die Beurteilung der geplanten Produktentwicklung und die Einführung in Unternehmen?
3. Vertritt der Experte ggf. eine wichtige Funktion bei der Überzeugung der Unternehmensentscheider für den vorliegenden Projektansatz?

Im Rahmen der vorliegenden Untersuchung fielen folgende Personengruppen unter den Expertenbegriff:

- Leiter IT-Management
- IT-Administrator
- Technischer IT-Consultant

Das Interview wurde vorab im Rahmen der Erprobungsphase mit dem IT-Administrator der ECOPLAN GmbH einem Pretest unterzogen. Die Interviewfragen können dem Anhang entnommen werden.

Für das geplante Interview wurden insgesamt 4 Experten ausgewählt und befragt.
Interviewleiter: Dr. Jochen Eckard

Name: Sven Linxweiler, B. Sc.
Position: Teamleiter Informationstechnologie
Institution: Deutsche Vereinigung für Wasserwirtschaft, Abwasser und Abfall e.V. (DWA)
www.dwa.de
Expertengespräch am 23.09.2020

Name: David Wollmann
Position: Executive Consultant
Institution: NTT Security (Germany) GmbH
www.global.ntt
Expertengespräch am 01.10.2020

Name: Felix Hahn
Position: IT-Consultant
Institution: OREXIS GmbH
www.orexis.de
Expertengespräch am 14.10.2020

Name: Sascha Kraatz
Position: Leiter IT-Management
Institution: Bundesverband deutscher Banken e.V. (BdB)
www.bdb.de
Expertengespräch am 14.10.2020

Die Interviews wurden mit MS-Teams in einem Zeitfenster von ca. 30 Minuten durchgeführt, aufgezeichnet und im Anschluss transkribiert.

Zur Vorbereitung wurden die Interviewfragen den Experten im Vorfeld zugesandt. Zusätzlich wurde den Interviewpartnern vorab eine „Executive Summary“ und ein Schaubild des geplanten Vorhabens zur Verfügung gestellt. Über die Interview-Fragen sollten die Experten zunächst ihre allgemeinen Kenntnisse zur Blockchain-Technologie beschreiben und dann Bezug auf das geplante Vorhaben nehmen. Abschließend sollten sie eine Einschätzung über die Entwicklung der Blockchain-Technologie in der Zukunft geben. Zur Datenaufbereitung wurde die Sequenzialität der Einzelinterviews aufgehoben und das empirische Material mittels Codierung

verschiedenen Kategorien zugeordnet, welche anhand des Studieninteresses erstellt wurden. Die Einzelinterviews wurden zu einer neuen Datengrundlage kombiniert, welche im Anschluss zugleich als mehrdimensionale Arbeitsgrundlage diente. Es erfolgte ein systematischer Vergleich der Interviewaussagen und die Herausarbeitung von Gemeinsamkeiten und Unterschieden, welche typologisierend zusammengefasst wurden. So entstand letztlich als Ergebnis eine hochverdichtete Zusammenfassung des Expertenwissens.

Ergebnisse der Interviews:

Was wissen Sie über die Blockchain-Technik zur Datenspeicherung?

Die Mehrheit der Experten waren der Meinung, ihr Wissen zur Blockchain-Technik ginge über die bekannten Schlagworte hinaus, aber es fehle noch an praktischem Know-how und an tieferem Verständnis technischer Funktionsweisen und Zusammenhängen der Blockchain-Technik. Besonders der Aspekt der Sicherheit in Bezug auf die Blockchain wurde von allen Experten hervorgehoben. Über die genannten Schlagworte, wie z.B. „Dezentralität“, „Konsensverfahren“ und „Hash-Schlüsselung“, wurde der Blockchain-Technologie von allen Befragten eine hohe Sicherheitsstufe zugesprochen. Einer der Experten hatte bereits Erfahrungen in einem Pilotprojekt zur Blockchain sammeln können. Auch die Unveränderbarkeit der Daten in der Blockchain wurde genannt. Ein Interviewter verwies hierbei auf das Spannungsfeld zwischen der Unveränderbarkeit der Daten und den Anforderungen der Datenschutzgrundverordnung (DSGVO).

Lässt sich Ihrer Meinung nach, das geplante technische Vorhaben in Ihrer Systemumgebung betreiben?

Bei der Frage, ob sich das geplante technische Vorhaben in der Systemumgebung der Experten umsetzen ließe, gab es zunächst bei allen Befragten etwas Zurückhaltung bei der Beantwortung. Neben der technischen Umsetzung kamen zunächst inhaltliche Fragen zum Projektvorhaben in Bezug auf die Speicherung der sensiblen Informationsobjekte auf. Von den Befragten wurde explorativ angeregt, nicht nur Feldwerte mit geringer Zeichenanzahl als sensible Information zu speichern, sondern auch sensible Dokument-Objekte (Word, Excel, PDF, etc.) sicher auf der Blockchain abzulegen. Dabei wurden auch konkrete Anwendungsfälle (Dokumente) aus ihrem Unternehmensumfeld genannt. Eine technische Umsetzung des Projektansatzes in ihren Unternehmen hielten alle Experten für möglich. Ein Experte hatte eher Bedenken, dass sein Unternehmen eventuell zu klein sei, um eine Blockchain-Lösung aufzusetzen. Alle Experten sahen Einsatzmöglichkeiten für den technischen Ansatz. Ein Experte hielt zwar die technische Umsetzung in seinem Unternehmen für durchführbar, sah aber eher ein Problem des Nichtvorhandenseins hochsensibler Daten, die eine Umsetzung rechtfertigen könnte.

Welche Vor- und Nachteile sehen Sie bei dem geplanten Vorhaben?

Als Gründe für den Einsatz der Blockchain-Technologie wurde von der Mehrheit der Experten vor allem die dezentrale Speicherung der Daten genannt. Darüber hinaus wurde die Blockchain als fälschungssicherer und weniger stark angreifbar eingestuft als herkömmliche Systeme zur Datenspeicherung, gefolgt von Argumenten wie Nachvollziehbarkeit und Fälschungssicherheit. Kritisch betrachtet wurde die Problematik der Datenlöschung oder Datenänderung bei der Blockchain-Technologie. Ein Befragter verwies auf das wachsende Datenvolumen auf den Clients, das zu händeln sei, wenn jede Eingabe unveränderbar und nicht löscher festgeschrieben wird, z.B. in dem Fall, dass sich ein Benutzer bei der Eingabe aus Versehen „vertippt“. Auch die Datenmenge- bzw. Datengröße wurde genannte. Es wurde befürchtet, dass die Datenmenge bzw. Feldgröße nicht ausreichen könnte, um die anvisierten sensiblen Daten aus dem Unternehmensumfeld speichern zu können. Es sollte daher zusätzlich die Speicherung von Dokumenten möglich sein, um in der Unternehmenspraxis einsetzbar zu sein. Ein Experte merkte an, er könne nicht einschätzen, wieviel Infrastruktur, Zeit, Aufwand bzw. Kosten mit dem

Aufsetzen der Blockchain verbunden sei. Der Aufwand müsse im Vergleich zum Nutzen stehen. In diesem Zusammenhang wurde auch der Einsatz eines „Storage-Systems“ anstatt einer Blockchain-Lösung genannt. Ein anderer Experte stellte sich den Auswahlprozess bezüglich der unkritischen und sensiblen Informationen schwierig vor. Auch die eventuell deutlich verlängerten Ladezeiten der sensiblen Informationen aus der Blockchain zur Laufzeit im CRM-System wurde aufgeführt.

Wie beurteilen Sie die mögliche Akzeptanz für den beschriebenen technischen Ansatz in Ihrem Unternehmen/Organisation?

In Bezug auf die Frage der möglichen Akzeptanz des technischen Vorhabens im Unternehmen waren die Antworten sehr unterschiedlich und kurz. Ein Experte äußerte hierbei, „*Wir stehen neuen Dingen eher skeptisch gegenüber*“. Ein Einsatz käme nur in Betracht bzw. würde akzeptiert werden, wenn er einen deutlichen Vorteil erzielen könnte. Andere Experten äußerten Offenheit und Interesse gegenüber neuen Technologien für ihr Unternehmen und ordneten das Thema dem großen Bereich der Digitalisierung zu, welcher von ihrem Unternehmen forciert werde. Ein Experte machte die Akzeptanz davon abhängig, dass neben der geplanten Feldwert-Speicherung auch Dokumente sicher und performant gespeichert werden können. Das würde die Chance eines möglichen Einsatzes deutlich erhöhen.

Wie speichert Ihr Unternehmen/Organisation sensible Daten und sind Sie mit der Lösung zufrieden?

Bei der Frage nach der bestehenden Datenspeicherungsform schilderten die Experten sehr unterschiedliche Lösungen. Der Schwerpunkt der Datenspeicherung liegt nach Schilderungen der Experten bei der zentralen Speicherung von Informationen auf eigenen Servern „Inhouse“ mit angeschlossenem Backup-System. Ein Experte erklärte dazu, dass der Sicherungsmechanismus bei seinem Unternehmen nicht sehr hoch sei. Die Informationen seien im Klartext gespeichert und nur zugriffsgeschützt. Der eigentliche Sicherungsmechanismus würde nur aus Methoden bestehen, um die Daten vor Verlust zu schützen, z.B. durch diverse Backup-Systeme. Die Bandbreite der genannten internen Lösungen reichte vom einfachen Server mit Backup-System bis zum eigenen Rechenzentrum „Inhouse“ mit Storage-Systemen. Nur ein Befragter gab an, dass die Daten aus seinem Unternehmensumfeld aufgrund der geringen Menge in einer Cloud-Lösung gespeichert werden. Er merkte dazu aber an, dass er mit dieser Lösung nicht zufrieden sei, da er die Verantwortung für die Sicherheit der Daten gerne selbst in die Hand nehmen würde. Die anderen Experten waren mit der bestehenden Datenspeicherungsmethode in ihrem Unternehmen zufrieden, aber offen für Verbesserungen.

Wo sehen Sie weitere Möglichkeiten für den Einsatz der Blockchain-Technologie?

Als Einsatzmöglichkeit der Blockchain-Technologie wurde von allen Experten die über die Medien bekannten Anwendungsbereiche der Technologie genannt, z.B. für Finanztransaktionen, Gesundheitswesen, Herkunftsnachweise, Supply Chain Management oder öffentliche Verwaltung. Die Experten äußerten hier übereinstimmend das die Einsatzmöglichkeiten hauptsächlich von der Verarbeitung sensibler fälschungssicherer Informationen geprägt seien. Ein Experte assoziierte die Einsatzmöglichkeiten mit Großprojekten und nannte hier speziell Walmart. Über die medial bekannten Einsatzgebiete hinaus wurden von den Befragten keine weiteren Möglichkeiten genannt. Ein Experte brachte die Vermutung hervor, dass die Blockchain-Technik aufgrund mangelnden Know-hows wahrscheinlich oft mit Bitcoin verwechselt bzw. gleichgesetzt werde. Dies könnte ein Grund sein, warum es über die bekannten Anwendungsfälle hinaus kaum weitere Innovationen gäbe.

Wie schätzen Sie die Entwicklung der Blockchain-Technik in der Zukunft ein?

Alle Experten bescheinigten der Blockchain-Technologie für die Zukunft ein hohes Potential und merkten übereinstimmend an, dass der Hype der Blockchain erst mit dem Aufkommen der Kryptowährungen begann. Ein Befragter schilderte verschiedene Hemmfaktoren welche die Blockchain-Entwicklung aktuell beeinflussen würden. Er sah die Blockchain gesellschaftlich aktuell in einer Nische mit teilweise eher negativem Image wie „Bitcoin, Darknet ... gefährlich, kompliziert“ behaftet. Die Blockchain-Technologie müsse aus dieser Nische herausgeholt und dem IT-Personal informativ nähergebracht werden. Viele IT-Beschäftigte hätten Berührungsängste mit der Technologie, weil sie vielen noch unbekannt sei. Es gäbe nur wenig strukturierte Praxisinformation, die einen guten Überblick über die Technik liefere. Es müssten praktikable und verständliche Anleitungen für die Praxis erstellt werden, damit sie zum Einsatz kommen könnten. Ein vertretbarer Aufwand wäre dabei auch zu berücksichtigen. Alle Experten waren der Meinung, dass die Blockchain-Technologie keine Universallösung ist und sich der sinnvolle Einsatz nur auf bestimmte Bereiche beschränkt. Als Hauptanwendungsbereich der Blockchain-Technologie für die Zukunft wurde der Währungsbereich mehrfach genannt, aber auch Blockchain-Lösungen ohne Kryptowährung wurden eine Berechtigung für die Zukunft zugeschrieben.

Zusammenfassung der Ergebnisse

Aus den geführten Interviews wurde deutlich, dass zwar gewisse Grundkenntnisse zur Blockchain-Technologie bei den Experten vorliegen, aber ein tiefergehendes Wissen über vernetzte Zusammenhänge der Technologie und der möglichen Implementierung betreffend fehlt. Bei fast allen Fragen war der Inhalt und das Volumen der Antworten überschaubar. Daraus wurde vermutet, dass zum relativ neuen Thema der Blockchain und deren Anwendungsmöglichkeiten noch keine umfassenden, tiefergehenden Kenntnisse bzw. Erfahrungswerte vorlagen. Alle Experten antizipierten mit der Blockchain-Technologie einen hohen Sicherheit-Level. Dem geplanten Vorhaben, der Speicherung sensibler Daten aus CRM-Systemen auf eine Blockchain, sahen sie interessiert, aber teilweise auch zurückhaltend, entgegen. Bei der Speicherung sensibler Daten kam mehrfach die Frage auf, ob neben spezifizierten Feldwerten einer Eingabemaske auch sensible Dokumente als Objekt (z.B. PDF, Excel- oder Word-Datei) gespeichert werden können. Dies war ein wichtiger Hinweis für die technische Ausgestaltung der Schnittstellen-Logik und des Blockchain-Frameworks für das vorliegende Projekt.

5.2 Prüfung rechtlicher Anforderungen

Der Anwendungsbereich der Blockchain-Technologie gestaltet sich vielseitig. Neben den Digitalwährungen wie z.B. Bitcoin oder Ether, können auch Verträge in digitaler, selbstausführender Form über die Blockchain als sogenannte „Smart Contracts“ abgewickelt werden. Dadurch wird das Recht vor besondere Herausforderungen gestellt und unterschiedliche Rechtsgebiete werden tangiert. Die Bandbreite umfasst die zivilrechtliche Behandlung von Zahlungsvorgängen oder Smart Contracts einschließlich der Haftungsproblematik über Datenschutz und IT-Sicherheit, aufsichts- und verbraucherrechtliche Fragen bis hin zu strafrechtlichen Aspekten. Bei dem vorliegenden Projektvorhaben ist der Anwendungsbereich der Blockchain-Technologie auf die Speicherung von sensiblen Daten aus einem CRM-System beschränkt. Es werden keine Währungstransaktionen durchgeführt oder sonstige selbstausführende Verträge (Smart Contracts) für „geschäftliche“ Vorgänge eingesetzt. Daher wurde die rechtliche Beurteilung im Weiteren vorrangig auf die Datenspeicherung und deren rechtliche Implikationen untersucht. Eine vollumfängliche Untersuchung der rechtlichen Situation der Blockchain-Technologie würde über den Umfang der Studie hinaus gehen und müsste in einer getrennten Studie untersucht werden. Der Schwerpunkt der Untersuchung lag im Bereich der Datenschutzgrundverordnung (DSGVO). Ob beim Einsatz der Blockchain-Technologie datenschutzrechtliche Anforderungen zu berücksichtigen sind, richtet sich danach, ob die

verwendeten Transaktionsdaten einen Personenbezug aufweisen. Die Mehrzahl der Artikel der DSGVO beschreiben den Anwendungsbereich und das Inkrafttreten; oder sie geben Anweisungen, wie staatliche Einrichtungen organisiert sein sollen. Alle weiteren Artikel betreffen allgemein datenverarbeitende Unternehmen. Bei der Verarbeitung personenbezogener Daten in einer Blockchain sind insbesondere die Artikel 15-18 und 20 der DSGVO zu beachten.⁷

- Auskunftsrecht (Artikel 15)

Das Auskunftsrecht umfasst die Forderung nach Nachvollziehbarkeit und Transparenz der Datenerhebung und Datenverarbeitung. In einer öffentlichen sowie in einer privaten Blockchain kann der Nutzer aufgrund des Designziels der Transparenz einer Blockchain jederzeit die eigenen Transaktionsdaten einsehen. Informationen jedoch in Bezug auf das Auskunftsrecht in Form von Angaben über die verarbeitende Stelle, Empfänger, Verarbeitungszwecke, etc. sind je nach Blockchain-Modell nur schwer zugänglich. Gerade bei öffentlichen, genehmigungsfreien Blockchains sind diese Daten aufgrund der fehlenden Regulierung schwer in Erfahrung zu bringen.

Eine der größten datenschutzrechtlichen Herausforderungen ist der Zustand, dass sich die Unveränderbarkeit der Blockchain und die Durchsetzung von datenschutzrechtlichen Betroffenenrechten antagonistisch gegenüberstehen. Da die Daten auf der Blockchain faktisch nicht mehr verändert werden können, gestaltet sich die Umsetzung des Rechts auf Löschung von Daten, das Recht auf Berichtigung und das Recht auf Vergessen schwierig.

- Recht auf Berichtigung (Artikel 16) und Recht auf Löschung (Artikel 17)

Jegliche Änderungen bzw. Löschung von Transaktionsdaten widerspricht dem grundlegenden Ziel der Manipulationssicherheit einer Blockchain. Eine Datenänderung kann bei der Blockchain-Technologie nur über die Generierung einer neuen Transaktion realisiert werden. Die alten Werte bleiben im „Hauptbuch“ (Ledger) bestehen. Änderungen direkt an den „alten“ Werten hätten zur Folge, dass die Konsistenz der Hashwerte in der Blockkette nicht mehr gegeben ist. Die Blockchain müsste komplett neu durchgerechnet (Hashing), genehmigt und verteilt werden. Aufgrund dieser Problematik haben sich einige Ansätze zur Datenänderung bzw. Datenlöschung entwickelt.

In genehmigungsbasierten Blockchains könnte über ein „Rollback-Verfahren“ eine bevorzugte Gruppe von Peers bestimmt werden, die die Historie der Blockchain umzuschreiben und Transaktionen nachträglich zu ändern.

Ein weiteres Verfahren ist die sogenannte „Off-Chain-Datenspeicherung“. Dabei werden die personenbezogenen Daten nicht in der Blockchain, sondern extern in einer „Off-Chain-Datenbank“ gespeichert und über einen Hashwert in der Blockchain referenziert. Werden Änderungen vorgenommen, ist der Hashwert nicht mehr valide, der Transaktionsblock ist aber weiterhin gültig und somit die Blockkette unverändert konsistent.

Auch der Einsatz von sogenannten „Chameleon-Hashes“ gehört zu den Lösungsansätzen.

Dabei wird über eine Hashfunktion eine spezielle Hintertür „trapdoor“ benutzt, um die Konstruktion von Kollisionen bei unterschiedlichen Daten mit demselben Hashwert zu arrangieren. Über diesen Mechanismus lassen sich Daten auf der Blockchain ändern bzw. austauschen, ohne den Integritätsschutz der Blockchain zu verletzen. Die Funktion der Hintertür ist von einer vertrauenswürdigen Stelle zu verwalten.

Ein weiterer Ansatz besteht darin, sogenannte „Forks“ der Blockchain zu erzeugen. Ein Fork bedeutet die Herbeiführung einer Abspaltung der Blockchain bei dringendem Revisionsbedarf aufgrund bedeutender Sicherheitsvorfälle. Voraussetzung dabei ist, dass sich eine gewisse kritische Menge der Nutzer darauf verständigen, dem Fork zuzustimmen und die Konsensbildung in einem neuen Zweig befürworten, damit das benötigte Vertrauen in die neue Blockchain

⁷ Berghoff, C. et al. (2019). Blockchain sicher gestalten, Bundesamt für Sicherheit in der Informationstechnik (BSI) S. 61 ff.. In: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Krypto/Blockchain_Analyse.pdf?__blob=publicationFile&v=5 (Zugriff: 16.08.2020)

sichergestellt werden kann. Bei genehmigungsfreien Blockchains kann sich dies allerdings als schwierig erweisen.

Eine ganz andere Vorgehensweise stellt das Verfahren „Mutable Blockchains“ dar. Über spezielle Smart Contracts wird die Sichtbarkeit von Transaktionsdaten gesteuert. Auch hier muss geregelt sein, wer diese Sperrung der Sichtbarkeit verwaltet.

Bei allen Ansätzen zur Änderung von Daten auf der Blockchain ist genau darauf zu achten, dass keine neuen Angriffsmöglichkeiten auf die Blockchain entstehen. Für die zuvor beschriebenen Verfahren der Datenänderung ist anzumerken, dass die zur Umsetzung befähigten Akteure zusätzliche Rechte bzw. eine besondere Stellung innerhalb der Blockchain benötigen. Dieser Umstand hat Auswirkungen auf das zugrunde liegende Vertrauensmodell (Konsensmechanismus). Darüber hinaus ist zu berücksichtigen, dass durch die verteilte Datenhaltung der Blockchain-Technologie viele lokale Kopien mit verschiedenem Aktualisierungsstatus auf den Peers (Nodes) liegen, die evtl. von den Änderungen nicht erfasst werden und damit die alten Daten weiterhin vorhanden sind.

Das Bundesamt für Sicherheit in der Informationstechnik (BSI) weist in seinem Dokument „Blockchain sicher gestalten“ darauf hin, dass die Vereinbarkeit der Grundeigenschaft einer Blockchain, unveränderlich zu sein, mit Datenschutzaspekten, wie z.B. dem Recht auf Löschung, technisch im Design der Implementierung des Anwendungsfalles Rechnung getragen werden muss.

- Recht auf Einschränkung der Verarbeitung (Artikel 18)

Bei der Blockchain-Technologie werden die Transaktionen in Blöcke gespeichert und miteinander verknüpft. Eine Einschränkung der Verarbeitung oder eine Trennung nach Zweck ist nicht Ziel dieser Technologie. Um dieser Forderung nachzukommen, könnten anonyme Transaktionen oder Smart Contracts eingesetzt werden, mit deren Hilfe die Anwender selbst den Zugriff auf ihre Daten steuern können und die Sicht für Außenstehende erschweren können. In privaten Blockchains ist die Beschränkung der Datenverfügbarkeit in Form von variabler Zugriffssteuerung und einem umfassenden Rollenkonzept möglich.

- Recht auf Datenübertragbarkeit (Artikel 20)

Das Recht auf Datenübertragbarkeit fordert, dass sich die zu einer Person gehörenden Daten in einem strukturiertem gängigen Format zusammenstellen lassen und der betreffenden Person zur Verfügung gestellt werden können. Je nach technischer Realisierung der Blockchain kann dies relativ einfach umgesetzt werden, zumindest dann, wenn sämtliche Daten direkt in der Blockchain gespeichert werden.

- Anonymisierung / Pseudonymisierung

Die Verwendung von Anonymisierungstechniken kann einen Ausweg darstellen.⁸ Im Hinblick auf die rechtliche Einordnung haben beide Formen jedoch unterschiedliche Auswirkungen auf die Bestimmungen des Datenschutzrechts.

- Pseudonymisierung

Nach Art. 4 Nr. 5 der Datenschutzgrundverordnung liegt eine Pseudonymisierung vor, wenn die Verarbeitung personenbezogener Daten in einer Weise erfolgt, in der die personenbezogenen Daten ohne Hinzuziehung zusätzlicher Informationen nicht mehr einer spezifischen betroffenen Person zugeordnet werden können, sofern diese zusätzlichen Informationen gesondert aufbewahrt werden und technischen und organisatorischen Maßnahmen unterliegen, die gewährleisten, dass die personenbezogenen Daten nicht einer identifizierten oder identifizierbaren natürlichen Person zugewiesen werden. Dabei werden pseudonymisierte Daten immer noch als personenbezogen betrachtet, wenn sie es vor der Pseudonymisierung auch waren.

⁸ Martini, M. et al. (2017). Die Blockchain-Technologie und das Recht auf Vergessenwerden. In: <https://www.uni-speyer.de/fileadmin/Lehrstuehle/Martini/BlockchainundRechtAufVergessenwerdenTyposkriptversion20-03-19NZ.pdf> (Zugriff: 12.10.2020)

- Anonymisierung

Eine Anonymisierung liegt vor, wenn die personenbezogenen Daten nicht mehr identifiziert werden können oder nur mit unverhältnismäßig großem Aufwand an Zeit, Kosten und Arbeitskraft. Ist dies der Fall, dann gelten die Grundsätze des Datenschutzes nicht mehr.⁹

Die Verschlüsselung von Daten ändert hierbei jedoch grundsätzlich nichts an ihrem Personenbezug solange ein entsprechender Schlüssel verfügbar ist, um die Daten zu entschlüsseln.

Bei genauerer Betrachtung der Blockchains wird klar, dass je nach eingesetzter Technologie personenbezogene Daten vorliegen, wenn auch in pseudonymisierter Form und damit das Datenschutzrecht Anwendung findet. Selbst „gehashte“ Daten (Einwegfunktion) können in den Anwendungsbereich fallen, wenn die Rohdaten bekannt sind und Hashes erzeugt werden, die mit den bestehenden Hashes verglichen werden und so ein Rückschluss auf die Ursprungsdaten möglich ist.

- Verantwortliche Stelle

Im Bereich der DSGVO liegt die Annahme zu Grunde, dass bei der Verarbeitung personenbezogener Daten jeweils ein zentraler Intermediär existiert, an den sich die Anforderungen der DSGVO richten. Der Verantwortliche entscheidet allein oder gemeinsam mit anderen über die Zwecke und Mittel der Verarbeitung von personenbezogenen Daten. Ihm obliegt die Mitteilungspflicht im Zusammenhang mit der Berichtigung oder Löschung personenbezogener Daten oder der Einschränkung der Verarbeitung. Bei Blockchain-Netzwerken hingegen entscheiden aufgrund des dezentralen Aufbaus oft eine Vielzahl von Akteuren darüber. Dies ist gerade bei öffentlichen und genehmigungsfreien Blockchains, wie etwa Bitcoin oder Ethereum, die als transnationale Netzwerke konzipiert sind, der Fall. Hinzu kommt, dass die DSGVO Datentransfers zwischen der EU und Drittstaaten nur erlaubt, wenn in Letzteren eine Datenschutzverordnung gilt, die analoge Schutzstandards vorweisen kann. Ist dies nicht der Fall, müssen zusätzliche Schutzmaßnahmen ergriffen werden. Dies lässt sich allerdings nur schwer in dezentralen Netzwerken umsetzen.

- DSGVO in öffentlichen / privaten Blockchains

Aus datenschutzrechtlicher Sicht unterscheiden sich die verschiedenen Varianten der Blockchain wie öffentliche, private und konsortiale Systeme, in wesentlichen Punkten. Bei öffentlichen Blockchain-Systemen, wie z.B. Bitcoin oder Ethereum bestehen keine Hürden für den Zugang zu den Daten und zur Nutzung des Netzwerks. Diese Systeme erlauben eine grundsätzliche Sichtbarkeit von Daten auf der Blockchain. Die Betreiber werden ersetzt durch ökonomische Anreizmechanismen (Konsensmechanismen) verschiedener Gruppen von Akteuren. Bei diesen Modellen existieren keine zu verpflichtende Betreiber. Ein weiteres Merkmal von öffentlichen Blockchain-Systemen ist die transparente Darstellung aller gespeicherten Transaktionen. In Verbindung mit der eindeutigen digitalen Referenz (öffentlicher Schlüssel), die einen Teilnehmer spezifiziert, sind viele der Systeme für die Anwender nicht anonym. Für die Darstellung der Historie der Daten und zum Analysieren und Aufbereiten von Daten existieren frei zugängliche Tools (z.B. Block-Explorer, Transaktionsgraphenanalyse, etc.). Mit diesen Tools lässt sich der dahinterstehende Akteur der Daten/Transaktionen ermitteln. In privaten und konsortialen Blockchains ist die datenschutzrechtliche Ausprägung eine andere. Die Entwicklung von privaten und konsortialen Blockchain-Systemen begründet sich auf spezielle Anforderungen von Unternehmen. Der Einsatz von privaten Systemen stellt höhere Anforderungen an den Schutz der Daten. Bei diesen Blockchain-Modellen existiert eine zentrale, identifizierbare und verantwortliche Stelle/Person im System und der Zugang zum Blockchain-Netz ist auf eine identifizierbare Zahl an Teilnehmern beschränkt. Der Zugang zu Transaktionen im System kann zusätzlich mit einem Rechtekonzept eingeschränkt werden.

⁹ Kunde, E. et al. (2017). Blockchain und Datenschutz, Faktenpapier h im Auftrag vom KPMG. In: <https://www.bitkom.org/sites/default/files/file/import/180502-Faktenpapier-Blockchain-und-Datenschutz.pdf> (Zugriff: 08.10.2020)

5.3 Tiefere Analyse Zielgruppe / Märkte

Wer als relevante Zielgruppe für das vorliegende Projektvorhaben in Frage kommt, leitet sich vorrangig aus den projektspezifischen Anforderungskriterien ab. So wurden im ersten Schritt die verschiedenen Eigenschaften und Merkmale herausgearbeitet, die für die Zielgruppenauswahl entscheidend sind. Zur Anbindung der Schnittstellen-Logik wird ein CRM-System mit offener System- und vorhandener Schnittstellenarchitektur vorausgesetzt. Die meisten am Markt verfügbaren CRM-Systeme bieten diese technische Möglichkeit. Der Zielgruppenfokus soll auf kleinen und mittleren Unternehmen (KMU) oder anderen Organisationen liegen, die über ein schmales IT-Budget verfügen. Per Definition werden kleine und mittelgroße Unternehmen dabei nach KMU-Klassifizierung der Europäischen Union, d.h. mit weniger als 250 Beschäftigten und einem Umsatz von höchstens € 50 Mio. (bzw. einer Bilanzsumme von max. € 43 Mio.), erklärt.

Die Veröffentlichungen von Projekten der Blockchain-Technologie in den Medien zeigt, dass vor allem große Unternehmen (Konzerne) über Eigenlösungen bzw. Projektansätze in diesem Bereich verfügen. Eine Eingrenzung auf bestimmte Branchen entfällt. Der Standort und das geschäftliche Stadium spielen dabei nur eine untergeordnete Rolle. Vorstellbar wäre die zu entwickelnde Schnittstellen-Logik zur Speicherung sensibler Informationen auf eine Blockchain parallel auch auf ERP-Systeme oder sonstige Verwaltungssysteme einer öffentlichen Behörde zu übertragen, vorausgesetzt die Systeme verfügen über eine offene Systemarchitektur. Es kann angenommen werden, dass in den Datenbeständen von Unternehmen aller Branchen und sonstigen Organisationen ein bestimmter Anteil sensibler Informationen vorliegt.

Gerade für KMU und andere Organisationen, die eine gewisse Affinität und Innovationsfreude für technische Neuerungen mit sich bringen, soll der technische Ansatz eine Option zu bestehenden Lösungen darstellen. Für Unternehmen und Organisationen, die eher verhalten und abwartend auf den Einsatz neuer Technologien reagieren, soll die vorliegende Studie dazu beitragen, eine gewisse Aufklärung zu leisten und die Technik, den Verantwortlichen näher zu bringen und damit eventuelle vorliegende Berührungängste weiter abzubauen. Es ist davon auszugehen, dass bei der Durchführung des vorliegenden Projektvorhabens und der damit verbundenen vielzähligen Kommunikationsprozesse weitere Einsatzszenarien offengelegt werden können.

Neben dem Einsatz bei kleinen und mittleren Unternehmen und sonstigen Organisationen könnte der Ansatz auf weitere Zielgruppe übertragen werden z.B.:

Rechtsanwälte, Notare

Wichtige Geschäfts- und Urkundendaten

Steuerberater

Wichtige Geschäfts- und Abschlussdaten

Banken

Sensibles internes Zahlenwerk

Gesundheitswesen

Patentendaten, Forschungsdaten

5.4 Vorbereitung der Teststellungen

Im Nachfolgenden wird die Vorbereitung der Teststellungen beschrieben.

5.4.1 Auswahl von zwei CRM-Systemen, Installation, Testdaten

Auswahl von zwei CRM-Systemen

Eine umfangreiche Recherche des Anbietermarktes für CRM-Systeme vermittelte zunächst einen Gesamtüberblick über die aktuelle Marktsituation und die Leistungsfähigkeit der einzelnen Systeme. Im Projektteam wurden im Anschluss definiert, welche Kriterien ein CRM-System als Testsystem für das vorliegende Projektvorhaben mindestens aufweisen sollte. Das CRM-Testsystem sollte über einen gewissen Bekanntheitsgrad und Marktanteil (Installationen) im KMU-Bereich verfügen. Zur Anbindung über Schnittstellen sollten die Testsysteme über eine offene Systemarchitektur verfügen.

Auszug möglicher CRM-Testsysteme:

- Salesforce CRM
- MS Dynamics 365 CRM
- SAP Hybris CRM
- Cursor CRM
- Nimble CRM
- SugarCRM
- 1CRM
- ...

Als Testsystem für die Überprüfung der Schnittstellen-Anbindung sollten im vorliegenden Projekt zwei CRM-Systeme zum Einsatz kommen. Die Auswahl fand gemeinsam im Projektteam statt. Die Entscheidung fiel auf das CRM-System des Gießener Softwarehauses Cursor und auf die CRM-Suite von MS Dynamics 365 CRM. Die Entscheidung für das Cursor-CRM begründete sich aus dem Umstand, dass der Firma ECOPLAN bereits Erfahrungen mit diesem System in anderen Projektsituationen vorlagen und sich dadurch der zu erwartende Einarbeitungsaufwand verkürzte. Die Auswahl des CRM-Systems von Microsoft beruhte auf der zu beobachtenden Wachstumsrate bzw. dem steigenden Marktanteil der Office365-Suite in den Unternehmen in den letzten Jahren.

1. CRM-System CURSOR „CRM 4.0“ der Cursor Software AG, Gießen



Die CURSOR Software AG zählt seit über 30 Jahren zu den führenden Anbietern von Software und Beratung für das Kunden- und Geschäftsprozessmanagement (CRM – Customer-Relationship-Management und BPM – Business Process Management).

Der Kundenstamm des mittelständischen Softwarehauses Cursor umfasst Unternehmen im KMU-Bereich vor allem in den Bereichen Dienstleistung, Energie, Entsorgung, Finanzen, Facility Management, Gesundheitswesen, Industrie, Non-Profit-Organisationen, Verbände, Kammern und Fundraiser. Das CRM-Softwaresystem von Cursor bietet über die offene Systemarchitektur viele Schnittstellen im Standard (OAP, REST, XML, MAPI, SMTP(S), IMAP(S), XML, EXCEL,

CSV, EWS, MS Graph API, CTI, SAP-Connector). Im Bereich des Customizings bietet das Cursor-CRM verschiedene Ansätze. Normale Anwendungslogik/Maskenlogik wird mit „Groovy“ umgesetzt. Zusätzlich können Prozesse mit der integrierten BPMN-Engine abgebildet werden. Reports und Auswertungen können mit dem integrierten Jasper-Studio erstellt werden.
<https://www.cursor.de/>

2. Microsoft Dynamics 365 CRM



Microsoft Dynamics CRM ist eine Customer-Relationship-Management Software von Microsoft. Sie ist ein Teil der Microsoft Dynamics Unternehmenssoftwareanwendungen. MS Dynamics CRM ist eine anpassbare und flexible CRM-Lösung für den Mittelstand. Das CRM-System von Microsoft Dynamics bietet folgende Schnittstellen: SOAP, XML, CSDL, JSON, REST, MAPI, SMTP(S), IMAP(S), EWS, MS Graph API und Azure. Microsoft offeriert für das CRM-System keine bestimmten Zielgruppen, da MS Dynamics CRM in allen Branchen zum Einsatz kommt. Im Bereich des Customizings bietet dieses System eine breite Funktionsauswahl. Über Visual-Studio können Services in .net, java, php, go, ruby, node.js und python umgesetzt werden. Hinzu können Unternehmensprozesse über Workflows und dem „Windows Workflow Foundation-Tool“ erstellt werden.

<https://dynamics.microsoft.com/de-de/>

5.4.2 Identifikation potentieller Blockchain-Modelle / Technologiefilter

Im Folgenden werden verschiedene Blockchain-Modelle dargestellt. Dabei werden gewisse Grundkenntnisse der Blockchain-Technologie vorausgesetzt. Nicht alle technischen Begrifflichkeiten sowie Ansätze aus dem Bereich der Blockchain-Technologie werden hier ausführlich beschrieben.

Die Blockchain stellt eine noch relativ junge Technologie im Kontext der Digitalisierung dar. Sie erlangte mit dem Aufkommen der Kryptowährung Bitcoin im Jahr 2009 erstmals weltweite mediale Aufmerksamkeit. Die Blockchain ist inzwischen mehr als nur die Technologie hinter Bitcoin. Aktuell erschließen sich zahlreiche neue Anwendungsfelder und Umsetzungsmöglichkeiten, die weit über eine virtuelle Währung hinausgehen. Der Blockchain liegt das Konzept eines sogenannten „Distributed Ledger“ (verteiltes Hauptbuch) zugrunde um digitale Datensätze oder Transaktionen in einem verteilten Rechnernetz zu speichern. Das Blockchain-Netz kann dabei unterschiedlich ausgestaltet sein. Als offene Netzwerke, an denen jeder teilnehmen kann oder als private Netzwerke, bei denen die Teilnehmer zuerst zugelassen werden müssen. Die dezentrale Speicherung nicht veränderbarer Daten ist gemeinsamer Bestandteil aller Blockchain-Modelle. Die Unterschiede ergeben sich z.B. beim Benutzerkreis, der Validierung, dem verwendeten Konsensmechanismus und der Transaktionsgeschwindigkeit. Die Transaktionsdaten werden zu Blöcken zusammengefasst und nach Prüfung im Distributed Ledger gespeichert. Jeder Block erhält einen Verweis zum vorherigen Block und der wiederum zu dem ihm vorangegangenen Block (Blockkette). Alle Informationen aus einem Block mit dem Hashwert des vorherigen Blocks werden über einen weiteren Hashwert verschlüsselt. Nur bestimmte Bestandteile der ursprünglichen Information werden offen zugänglich gemacht. Um einen Datenblock der Blockkette hinzuzufügen, muss er von den Teilnehmern des Netzes validiert werden. Wenn die Mehrheit der Teilnehmer die Korrektheit des Blocks bestätigt, wird dieser an die Blockkette angehängt. Die Art und Weise dieses Vorgangs bestimmt der jeweilige

Konsensmechanismus. Da in der Blockchain jeder Block über einen Hashwert auf den vorherigen Block verweist, ist eine nachträgliche Änderung von Werten innerhalb eines Blocks nicht möglich. Dadurch wird sichergestellt, dass die in der Blockchain enthaltenen Daten richtig sind und nicht verändert bzw. manipuliert wurden. Weiterhin wird darüber die Integrität der Daten gewährleistet. Jede gespeicherte Transaktion bleibt in seiner Form im Block erhalten. Durch die Verteilung der Blockchain als identische Kopie auf allen Rechnern (Peers) im Netz hat der Ausfall eines Netzwerkknoten keine kritischen Folgen auf das Gesamtsystem. Der Abgleich von unterschiedlichen Datenbanken entfällt. Die Blockchain-Modelle können daneben (komplexe) Aktionen durch sogenannte „selbstaussführende intelligente Verträge“ (Smart Contracts) ausführen. Dies dient z.B. der automatisierten Abwicklung von Verträgen, indem sie Bedingungen in Echtzeit überwachen und Vertragsbestandteile automatisiert selbständig umsetzen. Dabei agieren sie sicher und verifizierbar. Damit lässt sich Geschäftslogik in vielfältiger Weise selbständig ausführen. Aufgrund dieser Fähigkeiten sind unterschiedliche Anwendungsfelder vorstellbar. Die Technologie der Blockchain basiert auf verschiedenen Bekannten, aber auch neuen Technologien-Komponenten aus den Bereichen der verteilten Systeme wie P2P-Netzwerke, Sicherheit und Kryptografie.

Somit umfasst die Blockchain-Technologie ein komplexes und umfangreiches Gebiet, das aus zahlreichen Konzepten und verschiedenen Blockchain-Varianten besteht.

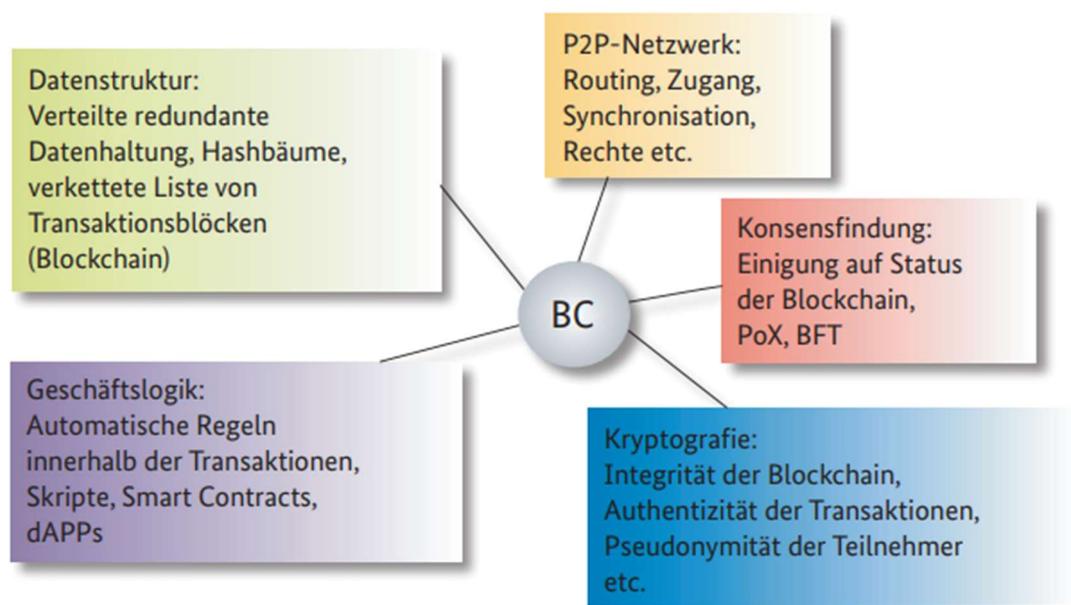


Abbildung 1: Grundbausteine der Blockchain-Technologie¹⁰

¹⁰ Berghoff et al. (2019). Blockchain sicher gestalten, Bundesamt für Sicherheit in der Informationstechnik (BSI) S. 10. In: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Krypto/Blockchain_Analyse.pdf?__blob=publicationFile&v=5 (Zugriff: 16.08..2020)

5.4.3 Grundtypen der Blockchain-Modelle

Die zum Zeitpunkt der Studie verfügbaren Blockchain-Modelle lassen sich grob in drei Grundtypen unterteilen: öffentliche (public) Blockchain, private (private) Blockchain, konsortiale (federated) Blockchain.¹¹

Öffentliche (public) Blockchain

Eine „öffentliche Blockchain“ basiert auf einem offenen Netzwerk.¹² Ein Beitritt ist ohne Erlaubnis möglich, so dass jeder ihr beitreten kann, indem er die Software herunterlädt und einen eigenen Knoten, auch als „Peer“ oder „Node“ bezeichnet, betreibt. Die Teilnehmer können dabei anonym sein. Bekannte öffentliche Blockchains, wie Bitcoin und Ethereum setzen sich aus tausenden von Knoten zusammen. Dadurch entsteht ein globales und dezentrales Netzwerk unabhängiger Knoten. Jeder Knoten kommuniziert mit anderen Knoten und verifiziert Transaktionen, anstatt dass dies über eine zentrale Stelle oder eine kleine Gruppe von Instanzen geschieht. Die Teilnehmer können Daten schreiben und lesen. Jeder Akteur kann so das Hauptbuch einsehen (z.B. über den Blockchain-Explorer) und am Konsensprozess teilnehmen. Grundsätzlich gibt es bei dieser Variante keine Beschränkung der Teilnahme. Das Netzwerk verfügt in der Regel über einen speziellen Anreizmechanismus, um mehr Teilnehmer zu gewinnen. Jede Transaktion wird von den Akteuren geprüft, um zu entscheiden, ob sie gültig ist. Das geschieht über verschiedene Konsensmodelle. Zwei weit verbreitete Konsensalgorithmen sind Proof of Work (PoW) und Proof of Stake (PoS). Mit ihnen wird bestimmt, welche Blockchain-Teilnehmer Blocks zu einer Blockchain hinzufügen und validieren dürfen. Bei Proof of Work (PoW), muss ein rechenintensives Puzzle gelöst werden (der Prozess wird „Mining“ genannt), bevor ein Block validiert werden darf. Als Motivation für die Arbeit erhalten die Validierer z.B. eine Transaktionsgebühr gutgeschrieben oder dürfen neue „Token“ generieren. Die Schwierigkeit, die Rechenaufgabe zu lösen, ist proportional zur Gesamtmenge an Rechenleistung, die für die Lösung der Aufgabe eingesetzt wird. Der gesamte Prozess ist mittlerweile sehr energieintensiv. Öffentliche Blockchains mit PoS-basiertem Konsensalgorithmus verbrauchen etwas weniger Energie, da der Akteur, der einen Block hinzufügen oder validieren darf, vorher festgelegt wird. Bei dieser Variante haben die Teilnehmer z.B. die Möglichkeit, Blöcke proportional zu ihrem Einsatz an Token („Stake“) zu validieren. Die Offenheit dieser Blockchain-Modelle, bietet wenig Privatsphäre für Transaktionen. Um ein verteiltes Hauptbuch in großem Maßstab zu führen, ist in Folge eine beträchtliche Menge an Rechenleistung und damit Energie erforderlich. Anwendung finden die Public Blockchains vor allem bei den meisten großen Kryptowährungen. So zählen Bitcoin und Ethereum heute zu den bekanntesten und größten öffentlichen Blockchain-Netzwerken.

Private (private) Blockchain

Anders verhält es sich bei einer „private Blockchain“. Sie wird nur von autorisierten Teilnehmern über eine Zugriffskontrolle betrieben. Damit ist sie zentralisierter als eine öffentliche Blockchain, die tausende von Knoten umfassen kann. Die Teilnehmer benötigen eine Erlaubnis, die Blockchain zu lesen, zu schreiben oder zu prüfen. Es gibt eine oder mehrere Entitäten, die das Netzwerk steuern. Eine Regulierungsbehörde könnte Lizenzen für die Teilnahme ausstellen oder ein definiertes Konsortium könnte stattdessen die Entscheidungen treffen. Die Blockchain kann über verschiedene Datenzugriffsebenen verfügen, um bestimmte Daten vertraulich zu behandeln. Dadurch gewährleisten sie ein höheres Maß an Sicherheit, Datenschutz und Leistung. Die Transaktionen und Daten sind nicht öffentlich sichtbar und können nur von den teilnehmenden Parteien abgerufen werden. Private Blockchains ermöglichen Benutzern unterschiedliche Berechtigungsstufen, sodass der Zugriff eingeschränkt werden kann. Es können bei Bedarf Knoten hinzugefügt oder entfernt werden. Da die Anzahl der autorisierten Teilnehmer

¹¹ Schiller K. (2019). Die Blockchain Typen im Überblick. In: <https://blockchainwelt.de/blockchain-typen-ueberblick/> (Zugriff: 23.10.2020)

¹² Schmitz P. (2019). Was ist eine Public Blockchain?. In: <https://www.blockchain-insider.de/was-ist-eine-public-blockchain-a-871294/> (Zugriff: 23.10.2020)

(Knoten) in einer privaten Blockchain geringer ist als in einer öffentlichen Blockchain, können wesentlich mehr Transaktionen pro Sekunde verarbeitet werden. Im Gegensatz zu öffentlichen Blockchains müssen bei privaten Blockchains keine Anreize für „Miner“, die die Transaktionen validieren ausgezahlt werden. Transaktionen werden bei privaten Blockchain-Netzen billiger, denn es sind oft nur wenige Knoten zur Validierung nötig. Die Berechnung der neuen Blocks ist weniger aufwändig was sich in einem geringeren Energiebedarf niederschlägt.

Konsortiale Blockchain (Federated Blockchain)

Sogenannte „Konsortiale“ oder „Federated“ Blockchains gehören zu den privaten, zustimmungspflichtigen Blockchains. Eingesetzt wird diese Form der Blockchain vor allem bei Unternehmen, die in einer geschäftlichen Beziehung zueinanderstehen und eine gemeinsame Blockchain für ihre Business-Anwendungen nutzen wollen. Bei dieser Art der Blockchain-Variante lassen sich vor allem Geschäftsanwendungen unter strengen Governance-Richtlinien umsetzen. Alle Beteiligten erhalten dadurch eine ähnliche Behandlung und keine einzelne Stelle (Entität) hat die Autorität über das Netzwerk. Für die teilhabenden Unternehmen bietet das Blockchain-Konsortium eine regulatorische Umgebung, bei der jedes Unternehmen bei Bedarf seine Informationen austauschen kann.

Die Ausgestaltung und Formen einer Blockchain sind vielfältig und betreffen verschiedene Funktionsbereiche. So kann eine Blockchain nicht nur allgemein und frei für jeden zugänglich sein, sondern auch alternativ gewisse Beschränkungen hinsichtlich ihrer Teilnehmer und Nutzungsart haben. Die Begriffe öffentlich (public) und privat (private) beschreiben dabei die Möglichkeiten zur Teilnahme am Blockchain-Netzwerk selbst. Mit geschlossen (permissioned) bzw. offen (permissionless) ist hingegen die Möglichkeit zur Teilnahme am Konsens-Mechanismus definiert.

Gespeicherte Daten in einer offenen Blockchain können von jedem Blockchain-Teilnehmer gelesen werden, während in einer geschlossenen Blockchain nur bestimmte Teilnehmer Daten lesen dürfen.

Diese zwei Gegensätzen öffentlich/privat und offen/geschlossen ergeben insgesamt vier grundlegende Ausprägungsvarianten von Blockchains. Jedes dieser Merkmale eignet sich für unterschiedliche Anwendungsfälle.

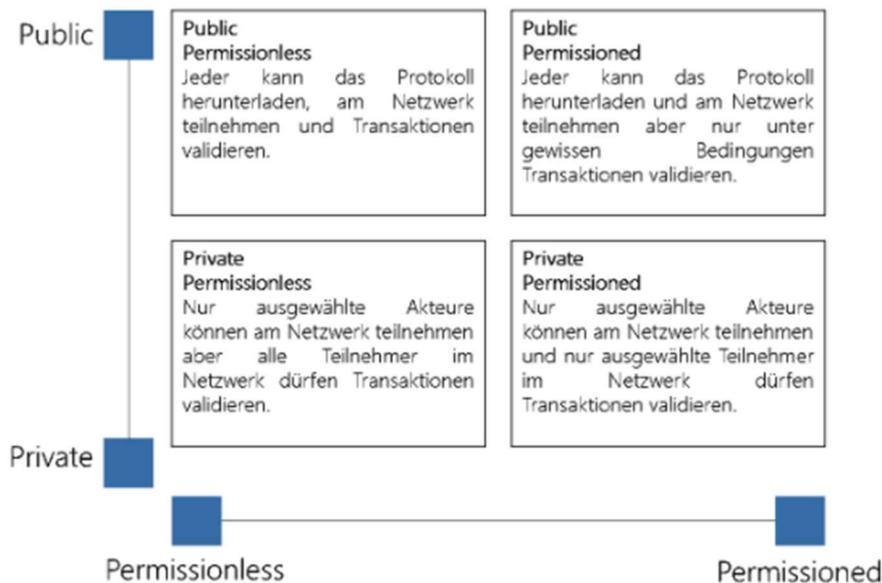


Abbildung 2: Mögliche Ausprägungsvarianten einer Blockchain¹³

Die Mischformen, wie public-permissioned oder private-permissionless Blockchains, sollen aufgrund ihrer Seltenheit und Komplexität an dieser Stelle nicht weiter ausgeführt werden.

Zusammenfassend kann festgehalten werden: Öffentliche und private Blockchains unterscheiden sich in der Funktionsweise, wie sie es Benutzern ermöglichen am Netzwerk teilzunehmen. Beide haben ihre Vor- und Nachteile und sind für unterschiedliche Anwendungsfälle ausgelegt.

5.4.4 Große bekannte Blockchain-Modelle / Vorselektion

In den vergangenen 5 Jahren hat sich das Blockchain-Umfeld stark weiterentwickelt, neue technische Ansätze haben sich durchgesetzt und wurden etabliert. Mittlerweile sind am Markt viele Blockchain-Varianten zu finden. Im Nachfolgenden werden einige der bekanntesten und repräsentativsten Blockchain-Plattformen im Hinblick auf ihre Anwendungspotentiale sowie Vor- und Nachteile beschrieben. Die vorliegende Studie erhebt keinen Anspruch auf eine vollumfängliche Benchmark-Studie für Blockchain-Systeme. In Anbetracht der projektspezifischen Anforderungen sollte das geeignetste Modell ermittelt werden. Es sollten nur Modelle in Betracht kommen die zum Zeitpunkt der Studie weit am Markt verbreitet waren. Es wurde angenommen, dass sich diese Modelle auch in Zukunft weiter am Markt durchsetzen und langfristiger zur Verfügung stehen als kleine „Nischen-Modelle“. Die vier nachfolgend analysierten Blockchain-Modelle fanden auch in anderen Studien Betrachtung, z.B. in der Studie „Analyse und Auswahl von Blockchain-Modellen für die US-Bundesregierung“, die von den IEEE Computer And Reliability Societies gemeinsam veröffentlicht und von James P. Howard II, Johns Hopkins Applied Physics Laboratory und Maria E. Vachino von Easy Dynamics Corp. 2019 verfasst wurde.¹⁴ Ebenfalls in der Arbeit von Jérémy Robert Yves Le Traon (2020), „Zulässige Blockchain-Frameworks in der Branche: Ein Vergleich“ werden die vier Blockchain-Modelle thematisiert.¹⁵

¹³ Zeiselmaier, A. et al. Technologie, Aufbau und Begrifflichkeiten im Kontext der Blockchain. In: <https://www.ffe.de/themen-und-methoden/digitalisierung/929-technologie-aufbau-und-begrifflichkeiten-im-kontext-der-blockchain> (Zugriff: 27.10.2020)

¹⁴ James P. Howard II et al. (2019). Ethereum, Fabric, Corda, And Multichain. Only One Is Government Ready. In: <https://www.forbes.com/sites/benjessel/2020/04/21/ethereum-fabric-corda-and-multichain-only-one-is-government-ready-new-report/?sh=1eff3285263b> (Zugriff: 10.11.2020)

¹⁵ Polge, J.; Robert, J. et al. (2020). Permissioned blockchain frameworks in the industry: A comparison. In: <https://www.sciencedirect.com/science/article/pii/S2405959520301909> (Zugriff: 18.11.2020)

Das Bitcoin-Netzwerk ist zwar das bekannteste Blockchain-Netzwerk, die Bitcoin-Blockchain ist jedoch keine universelle Umgebung für Anwendungen, die über die Kryptowährung hinausgehen. Das Bitcoin-Netzwerk konzentriert sich eng auf die Ausführung von Programmen, die Transaktionen mit der Bitcoin-Kryptowährung beinhalten. Für das vorliegende Projektvorhaben kommt das Bitcoin Blockchain-Modell daher nicht in Betracht.

Ethereum-Blockchain

Die Ethereum-Blockchain zählt vorrangig zu den öffentlichen Blockchains, Allerdings ist es möglich, auf Basis der Ethereum-Blockchain-Technologie semi-öffentliche oder geschlossene Blockchains aufzubauen und zu betreiben. Da die Ethereum-Software grundsätzlich auf einer öffentlichen Blockchain basiert, sind weniger umfangreiche Datenschutzfunktionen vorhanden. Die Programmlogik (Smart Contracts) wird in einer höheren Programmiersprache geschrieben und auf einer verteilten Hauptbuchtechnologie ausgeführt. Die überwiegende Mehrheit der intelligenten Verträge von Ethereum ist in einer vertragsorientierten Hochsprache namens „Solidity“ geschrieben. Eine Hochsprache, die speziell zu diesem Zweck entwickelt wurde. Es besteht zudem auch eine Go-Implementierung zum Schreiben von Smart Contracts, jedoch sind in Go geschriebene Programme nicht direkt implementierbar. Es müssen sogenannte Compiler geschrieben werden, um Smart Contracts von Go in Bytecode umwandeln zu können. Für die Anwendung der Smart Contracts steht eine virtuelle Maschine (VM) auf der Blockchain zur Verfügung, für deren Nutzung für die Ausführung des Programmiercodes eine Gebühr bezahlt werden muss. Die Gebühr ist in der Kryptowährung Ether zu bezahlen. Die Unterstützung von Tokens ist eine integrale Kernfunktion dieser Technologie. Mit Blick auf die Schaffung neuer Innovationen wurde Ethereum von Grund auf in Richtung Einsatz von Tokens konzipiert und ist somit eine Infrastruktur, auf der Unternehmen digitale Assets und digitale Geschäftsmodelle entwickeln können. Ethereum ist eine Blockchain, die völlig dezentral betrieben werden kann, ohne dass eine Kontrollinstanz erforderlich ist. Daher ist es möglich, mit Ethereum auch halböffentliche Blockchains vollständig dezentral zu gestalten. Ethereum zeigt seine volle Leistungsfähigkeit, wenn es um den Austausch digitaler Werte in einem vertrauenswürdigen Ökosystem unabhängiger Teilnehmer geht.¹⁶

Bei Quorum¹⁷ handelt es sich um eine private und zugangsbeschränkte Version der Ethereum-Blockchain („private permissioned blockchain“), die von der Investmentbank J.P. Morgan als Mitglied der Ethereum Enterprise Alliance (EEA) für einen verbesserten Zahlungsverkehr entwickelt wurde. Im Gegensatz zu Ethereum gewährleistet Quorum die Vertraulichkeit von transaktionsbezogenen Daten und macht diese nur den zugriffsberechtigten Transaktionsparteien, nicht aber dem gesamten Netzwerk zugänglich. Durch die Verifikation von Transaktionen auf Transaktions- anstatt auf Netzwerkebene ist die Blockchain darüber hinaus besser skalierbar und verfügt über eine höhere Transaktionsgeschwindigkeit.

Die Quorum-Blockchain ist eine konsortiale, permissioned Blockchain. Transaktionen, die getätigt werden, sind in Quorum kostenlos.

Hyperledger Fabric

Hyperledger Fabric¹⁸ ist ein Blockchain-Modell der Linux Foundation. Insgesamt sind im Projekt „Hyperledger“ verschiedene Distributed Ledger-Technologien und Projekte der Linux Foundation zusammengefasst. Der bekannteste Teil von Hyperledger ist das von IBM initiierte, entwickelte und geförderte Framework Hyperledger Fabric. Dieses Framework eignet sich besonders als Blockchain für B2B-Anwendungen und steht für eine leistungsfähige Distributed Ledger-Technologie. Durch die Begrenzung von Transaktionen auf beteiligte Mitglieder kann Hyperledger Fabric nur in geschlossenen Umgebungen eingesetzt werden, in denen alle Teilnehmer bekannt

¹⁶ Rajarshi, M.. Was ist Ethereum? In: <https://blockgeeks.com/guides/de/was-ist-ethereum/> (Zugriff: 19.11.2020)

¹⁷ Kerkmann, J. (2020). Was steckt hinter Business Blockchain Quorum? In: <https://blockchainwelt.de/was-steckt-hinter-der-business-blockchain-quorum/> (Zugriff: 17.10.2020)

¹⁸ <https://www.hyperledger.org/> (Zugriff: 10.11.2020)

sind. Die Zugangsbeschränkungen, die die Privatsphäre erhöhen, werden durch eine zentrale Stelle innerhalb des Netzwerkes festgelegt. Hyperledger Fabric verlangt von allen Transaktionen ein kryptografisches Zertifikat, das die vertraulichen Daten eines Benutzers beinhaltet. Aus jeder Identität kann das Protokoll spezielle Sicherheitsschlüssel generieren, mit denen die Teilnehmer im Netzwerk Transaktionen durchführen können. Die Identitäten der Akteure bleiben verborgen, um die Privatsphäre innerhalb des Netzwerkes zu sichern. Über spezielle „Channels“ können nur Teilnehmer miteinander kommunizieren, die über den berechtigten Zugriff verfügen. Hyperledger Fabric ist hoch modular aufgebaut, sodass je nach Bedarf unterschiedliche Komponenten zum Einsatz kommen. Die Daten werden von allen Mitgliedern der geschlossenen Hyperledger Fabric Blockchain verteilt und gespeichert. Die Dezentralisierung beschränkt sich auf die teilnehmenden Akteure. Private Transaktionen sind bei dieser Variante möglich. Spezielle Token werden in Hyperledger nicht unterstützt. Token oder ähnliche Konstrukte können zwar erstellt werden, bleiben aber auf den jeweiligen Anwendungsfall beschränkt. In Hyperledger Fabric werden intelligente Verträge (Smart Contracts) als „Chaincode“ bezeichnet. Fabric unterstützt das Schreiben von Chaincodes in Java, Node.js oder Go. Es existieren vielseitige Einstellungsmöglichkeiten innerhalb der Konfiguration. Als Blockchain-Variante eignet sich Hyperledger Fabric besonders für Anwendungsfälle, in denen Daten zwischen einer geschlossenen Gruppe performant ausgetauscht werden sollen. Für dezentrale Anwendungen, die auf digitalisierten Assets basieren, kommt Hyperledger Fabric aufgrund der fehlenden Tokenisierung nicht in Betracht. Leistungs-, Skalierbarkeits- und Datenschutzerfordernisse werden in Hyperledger Fabric über einen speziellen Betriebsmodus und eine detaillierte Zugriffssteuerung unterstützt. Durch die modulare Architektur kann Fabric an eine Vielzahl von Anwendungen angepasst werden. Alle Komponenten einer Hyperledger-Blockchain können innerhalb von sogenannten „Docker Containern“ betrieben werden.

Corda R3

Corda R3 ist ein flexibles Blockchain-Modell des R3-Konsortiums.¹⁹ Es ist auf Open-Source-Basis entwickelt und ermöglicht Unternehmen aus der Finanzbranche, ihre Geschäftsprozesse abzuwickeln. Dabei konzentriert sich Corda hauptsächlich auf die Verarbeitung komplizierter Transaktionen. Die Smart Contracts in Corda können jedoch nicht nur Programmierlogik, sondern auch „juristische Prosa“ enthalten, was für die hochregulierte Finanzindustrie sehr sinnvoll ist. Hinter der Plattform Corda R3 steht ein Konsortium aus über 300 Banken und Industriepartnern, das die Plattform im November 2016 auf Basis der Blockchain-Technologie veröffentlicht hat. Das Konsortium arbeitet auch mit Microsoft und Intel zusammen. Corda ist heute, neben Hyperledger Fabric, eines der führenden Software Frameworks für die Etablierung einer permissioned Blockchain.²⁰ Neben der Community-Version stellt R3 auch Corda Enterprise zur Verfügung. Corda Enterprise ist die kommerzielle Version von Corda. Die Enterprise-Version ist interoperabel und kompatibel mit Corda Open Source. Entwickelt wurde Corda für die Erfassung, Verwaltung und Synchronisation von Finanzverträgen zwischen regulierten Instanzen. Corda konzentriert sich dabei auf die Verarbeitung komplizierter Finanz-Transaktionen. Derzeit werden Lösungen auf Basis von Corda von vielen Banken, Versicherungen und Aufsichtsbehörden eingesetzt. Der Zugriff auf die Transaktionsdaten ist dabei eingeschränkt. Informationen werden bei Corda nur selektiv im Netzwerk verteilt (Need-To-Know-Prinzip). So hat jeder Teilnehmer nur Zugang zu Informationen, die ihn selbst betreffen. Corda führte diesen Mechanismus ein, um das Problem des Datenschutzes zu lösen. Die Idee dahinter ist die Transaktionen nur mit dem eigenen Netzwerk zu teilen. Corda hat keine eigene Kryptowährung. Der Ansatz von Corda unterstützt verschiedene Konsensverfahren. Auf Basis von Corda können auch Anwendungen erstellt werden, die über den Finanzsektor hinaus gehen.

¹⁹ Schiller, K. (2019). Was ist R3 Corda - Die Business Blockchain? In: <https://blockchainwelt.de/r3-corda-die-business-blockchain/> (Zugriff: 27.11.2020)

²⁰ <https://www.r3.com/corda-platform/> (Zugriff: 25.11.2020)

MultiChain

MultiChain²¹ ist eine Open-Source-Plattform, die eine Abzweigung (Fork) der Bitcoin-Blockchain darstellt. Entwickelt wurde sie von Coin Science. Mit MultiChain lassen sich öffentliche und private (permissioned) Blockchain-Netze erstellen. Die privat erstellten Blockchain-Modelle lassen sich besonders von Organisationen für Finanztransaktionen verwenden. Das konsensbasierte Berechtigungs-Management ist gerade für private Blockchains interessant. MultiChain unterstützt die Übertragung von benutzerdefinierten Assets / Tokens“. Es kann verwendet werden, um branchenübergreifende Anwendungen mit einem verteilten Konsensalgorithmus zu erstellen. Die Blockchain-Plattform unterstützt die Programmiersprachen C, C ++, Python, Ruby und JavaScript. MultiChain verfügt über sogenannte intelligente Filter.²² Eine Funktionalität, die eine benutzerdefinierte Regelcodierung zum Überprüfen von Transaktionen ermöglicht. Ein konfigurierbarer Konsensmechanismus (Round-Robin) ersetzt bei der privaten Variante das Proof-of-Work Verfahren und erhöht damit die Transaktionsgeschwindigkeit. Wenn es um Transaktionsregeln geht, unterscheidet sich MultiChain speziell von Fabric, Ethereum und Corda. MultiChain verfügt über mehrere, integrierte Abstraktionen, die grundlegende Bausteine für Anwendungen bereitstellen, ohne dass Entwickler ihren eigenen Code schreiben müssen. Bei der Entwicklung einer Anwendung auf MultiChain können die Transaktionsregeln nur mit speziellen „intelligenten Filtern“ abgebildet werden. Für eine kommerzielle Nutzung und Supportanfragen fallen bei MultiChain Gebühren an.

5.4.5 Vergleich der Eigenschaftsausprägungen vorselektierter Blockchain-Modelle

Die zuvor beschriebenen Blockchain-Architekturen verfügen über unterschiedliche Schwerpunkte in ihren Eigenschaftsausprägungen. Um eine gewisse Sicherheit in Bezug auf die Weiterentwicklung und den Fortbestand der Technologie zu gewährleisten, wurden für die Auswahl ausschließlich große, am Markt bekannte Blockchain-Modelle in Betracht gezogen. Ein wichtiges Beurteilungskriterium stellten vor allem die rechtlichen Anforderungen an die Blockchain-Modelle dar. Da bei der Speicherung von sensiblen Daten aus CRM-Systemen in der Regel davon auszugehen ist, dass ein gewisser Anteil der Daten einen Personenbezug aufweist, wurde insbesondere die Erfüllung der Anforderungen aus der Datenschutzgrundverordnung (DSGVO) geprüft. In einer umfangreichen Recherche wurden vielzählige Informationen aus Fachliteratur und dem Word-Wide-Web zu den unterschiedlichen am Markt verfügbaren Blockchain-Modellen extrahiert und im Hinblick auf das technische Vorhaben geprüft, z.B. allgemeine Beschreibungen, Zugang, Umgang mit Datenschutzerfordernungen, Konsensverfahren, ausführbarer Code (Smart-Contract), Transaktionsgebühren, treibende Kräfte, Entwicklungsstand sowie Bekanntheitsgrad.

²¹ <https://www.multichain.com/> (Zugriff: 26.11.2020)

²² Greenspan, G. (2018). Multichain, Private Blockchains, Smart Contracts. In: <https://www.multichain.com/blog/2018/12/smart-contract-showdown/> (Zugriff: 23.11.2020)

Hersteller	Ethereum	Hyperledger Fabric	R3 Corda	MultiChain
Internet	www.ethereum.org	www.hyperledger.org	www.r3.com	www.multichain.com
Kurzbeschreibung Blockchain	Entwicklungsplattform für ein breites Spektrum von Anwendungen	Modulare Blockchain-Plattform und Software-Stack für B2B-Anwendungen in Containern	Spezialisierte Entwicklungsplattform für Versorgungsketten/Finanzbranche	Blockchain mit integriertem Berechtigungsmanagement
Zugang	öffentlich/privat	privat	privat	öffentlich/privat
Konsensbildung	Implementierung Proof of Work (PoW, PoS) in öffentlichen Blockchains u. Proof of Authority in privaten Blockchains (z.B. Quorum)	Kein Mining, unterschiedliche Nodes übernehmen verschiedene Aufgaben (Clients, Peers/Endorsers, Orderers)	Kein Mining – mehrere Alternativen; unterschiedliche Knoten übernehmen verschiedene Aufgaben	Berechtigung rotiert im Round-Robin-Verfahren
Kryptowährung	Ether (ETH) in öffentlichen Blockchains	keine	keine	Native Unterstützung für mehrere Währungen
Transaktionsgebühren	nur bei öffentlicher Variante	keine	keine	keine
Ausführbarer Code	smart contract	Chaincode	smart contract	Intelligente Filter
Programmiersprache	Solidity	Java, Node.js, Go	Kotlin, Java (u.a.)	Python, C++, JavaScript, Ruby
Datenschutz	Bei öffentlicher Variante Datenschutz mangelhaft, bei privater Variante über Zulassung und Sichteinschränkung	Über Zulassung, Kanäle, private Daten (änderbar, löscherbar)	Über Zulassung und Sichteinschränkung	Über Zulassung und Sichteinschränkung
Treibende Kräfte	Enterprise Ethereum Alliance (unter anderem Accenture, Deloitte, Commerzbank AG, Ernst & Young, Hyperledger, Microsoft)	IBM, SAP, Huawei, Nokia, Intel, Samsung, Deutsche Börse, American Express, J. P. Morgan, BNP Paribas, Wells Fargo, Blockstream, Lykke	Corda Partner Network	SAP, Accenture, BCG, Cognizant, PwC
Bekanntheitsgrad	hoch	hoch	hoch	mittel
Reifegrad DTL	fortgeschritten	fortgeschritten	fortgeschritten	fortgeschritten

Abbildung 3: Vergleich ausgewählter Blockchain-Modelle²³

Zugang öffentlich / privat

Öffentliche und private Blockchains sind für unterschiedliche Anwendungsfälle konzipiert und haben einen unterschiedlichen Einsatzzweck. Bei öffentlichen Blockchains ist die enorme Rechenleistung, die im Rahmen des Konsensverfahrens (Mining) benötigt wird, als nachteilig anzusehen. Auch der Mangel an Privatsphäre ist hier kritisch zu beurteilen. Da jeder die Daten überprüfen bzw. einsehen kann, sind öffentliche Blockchains nicht dafür geeignet, vertrauliche Informationen zu speichern. Die öffentliche Verfügbarkeit des Inhalts von Transaktionen für alle Teilnehmer kann für zahlreiche Geschäftsanwendungsfälle problematisch sein. Deshalb finden in der Praxis für Unternehmensanwendungen vorwiegend private Blockchain-Modelle in zulassungsbeschränkter (permissioned) Form Anwendung.²⁴ Sie sind in der Regel durch den Verzicht des „Mining-Prozesses“ schneller, kostengünstiger und der Betreiber kann bestimmen, wer die Informationen einsehen darf. Public Blockchains eignen sich hingegen hauptsächlich für Anwendungen, die den Vorteil des offenen Systems nutzen wollen. Durch die Verwendung einer privaten und genehmigten Blockchain-Plattform können Unternehmen Daten sowohl transparent speichern als auch die Privatsphäre und Sicherheit sensibler Daten besser gewährleisten. Dabei werden den ausgewählten Teilnehmern nur bestimmte (eingeschränkte) Rechte eingeräumt. Die Transaktionen werden rein unternehmensintern validiert und sind nicht öffentlich zugänglich. Praxisprojekte, die insbesondere durch regulatorische Rahmenbedingungen, wie Compliance, mit der DSGVO getrieben sind, haben zudem einen zunehmenden Wechsel von öffentlichen Blockchains hin zu privaten Blockchain-Modellen bewirkt.

Bei Hyperledger Fabric und Corda handelt es sich um rein private Blockchain-Netzwerke - der Zugriff auf diese ist auf ausgewählte Parteien/Teilnehmer beschränkt. Ethereum bietet dagegen eine Plattform für jede Art von Anwendung. Der Schwerpunkt bei Ethereum liegt bei der

²³ Eigene Darstellung in Anlehnung an: <https://www.com-magazin.de/dl/1/0/6/6/1/5/9/Blockchains-im-Ueberblick-Auswahl.pdf> (Zugriff: 18.11.2020)

²⁴ Schmitz, P. (2019). Definition Permissioned / Private Blockchain.
 In: <https://www.blockchain-insider.de/was-ist-eine-permissioned-blockchain-a-871313/#:~:text=Eine%20Permissioned%20Blockchain%20ist%20eine,Personen%20angesehen%20und%20C3%BCberpr%C3%BCft%20werden.> (Zugriff: 17.10.2020)

öffentlichen Variante. Es besteht jedoch die Möglichkeit, eine private Blockchain nachzubilden. Ein Beispiel hierfür ist das Projekt „Quorum“. Anders als Ethereum gewährleistet Quorum die Vertraulichkeit von transaktionsbezogenen Daten und macht diese nur den zugriffsberechtigten Transaktionsparteien, nicht aber dem gesamten Netzwerk zugänglich. Mit dem Blockchain-Framework MultiChain kann sowohl eine öffentliche als auch eine private Blockchain aufgesetzt werden.

Konsensverfahren

Die Ethereum-Blockchain bietet als Konsensmechanismus das Verfahren Proof-of-Word (PoW) oder Proof-of-Stake (PoS) an. Der PoW-Mechanismus umfasst den von Bitcoin bekannten Mining-Prozess. Da das Verfahren sehr CPU-lastig ist, hat die Rechengeschwindigkeit der Knoten maßgeblichen Einfluss darauf, wer das Rätsel löst und einen passenden Nonce-Wert findet. In Ethereum müssen alle Peers einen Konsens über die Reihenfolge aller Transaktionen erzielen, unabhängig davon, ob ein Benutzer an einer bestimmten Transaktion teilgenommen hat. Dadurch ist das Verfahren wesentlich zeit- und kostenintensiver als die Konsensmechanismen der privaten Blockchains Corda oder Hyperledger Fabric. Alternativ dazu bietet Ethereum in der privaten Variante den Konsensmechanismus Proof-of-Authority (PoA) an. Dieser Mechanismus arbeitet ähnlich wie der in Hyperledger Fabric oder R3 Corda.²⁵ In Hyperledger Fabric und Corda gibt es kein Mining. Bei Fabric sind mehrere alternative Ansätze möglich. Unterschiedliche Knoten (Peers) übernehmen verschiedene Aufgaben (Clients, Peers/Endorsers, Orderers). Ein Client handelt im Namen eines Endbenutzers und erstellt Transaktionen und ruft diese auf. Die Peers pflegen das Hauptbuch und erhalten bestellte Aktualisierungsnachrichten von Bestellern, um neue Transaktionen durchzuführen. Bei Corda sind mehrere alternative Implementierungen modular möglich. Dabei übernehmen unterschiedliche Knoten verschiedene Aufgaben (zum Beispiel Knoten mit Notar-Funktion). Bei Corda wird die Gültigkeit von Transaktionen sichergestellt, indem ein intelligenter Vertragscode für eine bestimmte Transaktion ausgeführt wird. Alle erforderlichen Signaturen werden überprüft und es wird sichergestellt, dass alle Transaktionen, auf die verwiesen wird, ebenfalls gültig sind. Damit wird ein Konsens über die Einzigartigkeit unter Peers erreicht. Somit wird bei Corda der Konsensus ebenfalls nicht auf Netzwerk, sondern auf der Transaktionsebene erreicht.

In der privaten Version von MultiChain erfolgt der Konsens über eine PBFT-Variante, bei der es nur einen Validator pro Block gibt und mehrere Validatoren im Round-Robin-Verfahren arbeiten.

Kryptowährung

Ethereum als öffentliche Blockchain ist die einzige unter den Blockchain-Modellen, die mit einer nativen Kryptowährung „Ether“ ausgestattet ist. Fabric, Corda und MultiChain benötigen dagegen keine Kryptowährung, da der Konsens nicht über Mining erzielt wird.

Smart Contract / Programmiersprache

Der Anwendungscode hat bei den einzelnen Blockchain-Modellen unterschiedliche Bezeichnungen, z.B. Intelligente Verträge (Smart Contract), intelligente Filter oder Chaincode. Sein Hauptzweck ist es, mit der zugrundeliegenden Infrastruktur einer Blockchain zu arbeiten. Intelligente Verträge sind das dezentrale Äquivalent zum Anwendungscode. Anstatt an einem zentralen Ort ausgeführt zu werden, werden sie auf mehreren Knoten (Peers) in der Blockchain ausgeführt und erstellen oder validieren die Transaktionen. Bei Ethereum werden die Smart Contracts in einer vertragsorientierten Hochsprache namens „Solidity“ geschrieben, eine eigens dafür entwickelte Hochsprache. Hyperledger Fabric dagegen ist recht flexibel in der Auswahl und Implementierung der Smart Contracts, der als „Chaincode“ in Fabric bezeichnet wird. Hier werden insbesondere die drei Programmiersprachen Java, Go und Node.js unterstützt.

²⁵ Melnik, I. (2020). Comparison: Ethereum vs Hyperledger Fabric vs Corda. In: <https://merehead.com/blog/comparison-ethereum-hyperledger-fabric-r3-corda/> (Zugriff: 19.10.2020)

Corda dagegen bietet Kompatibilität zu einer ganzen Bandbreite an Programmiersprachen an. Bei MultiChain werden die Transaktionsregeln über sogenannte „Intelligente Filter“ gesteuert. Die Funktionsweise der Smart Contracts unterscheidet sich bei den gegenübergestellten Blockchain-Modellen deutlich. In Ethereum (public) werden die Smart Contracts auf allen Peers des Netzwerks ausgeführt, sobald diese einen Block mit den entsprechenden Transaktionen erhalten. In Hyperledger Fabric hingegen werden sie nur von den Peers ausgeführt, die zur Billigung einer Transaktion notwendig sind, bevor die Transaktion in einen Block aufgenommen wird. Bei R3 Corda ist es ähnlich. Hier wird die Ausführung nur von den Knoten vollzogen, die eine Transaktion zwingend überprüfen müssen. Private Blockchain-Frameworks wie Hyperledger Fabric und R3 Corda können, im Vergleich zu Ethereum (public) durch spezielle Transaktionen auch den Code von Smart Contracts aktualisieren, wenn alle zustimmungspflichtigen Knoten dem Update einwilligen. Wenn es um Transaktionsregeln geht, gibt es eine Möglichkeit, in der sich MultiChain speziell von den anderen Blockchains unterscheidet. MultiChain verfügt über mehrere „integrierte Abstraktionen“, die einige grundlegende Bausteine für Blockchain gesteuerte Anwendungen bereitstellen. Das Schreiben von Programmlogik für Entwickler entfällt. Trotz der unterschiedlichen Bezeichnungen für den Programmcode (Smart Contract) beziehen sich alle auf die gleiche Funktion, dem anwendungsspezifischen Code, der die Regeln einer Blockkette definiert.

Datenschutz / Sicherheit

Die Anforderungen an die Blockchain-Technologie, basierend auf der Datenschutzgrundverordnung (DSGVO), leiten sich hauptsächlich aus den Artikeln 15-18 und 20 ab:

In Artikel 15 der DSGVO wird das jederzeitige Auskunftsrecht sowie die Forderung nach Nachvollziehbarkeit und Transparenz der Datenerhebung und Verarbeitung gefordert. Die Forderung der Einsichtnahme eigener Transaktionen wird von beiden Varianten der Blockchain-Frameworks (öffentlich/privat) ermöglicht. Informationen in Bezug auf das Auskunftsrecht in Form von Angaben über die verarbeitende Stelle, Empfänger, Verarbeitungszwecke, etc. können bei öffentlichen, genehmigungsfreien Blockchains, wie Ethereum, aufgrund der fehlenden Regulierung nur schwer in Erfahrung gebracht werden. In privaten, zulassungsbeschränkten Blockchain-Netzen ist dies aufgrund der fest installierten, regulierenden Stelle möglich.

Der Artikel 16, „Recht auf Berichtigung“ und der Artikel 17, „Recht auf Löschung“ der DSGVO stellen die schwierigste Aufgabe an die Blockchain-Technologie dar. Dies ist darauf begründet, dass die Daten bei der Blockchain-Technologie unveränderbar und revisionssicher im Hauptbuch (Ledger) gespeichert werden. Jegliche Änderungen bzw. Löschung von Transaktionsdaten widerspricht dem grundlegenden Ziel der Manipulationssicherheit einer Blockchain. Eine Datenänderung kann nur über die Generierung einer neuen Transaktion realisiert werden. Dabei bleiben die alten Werte im „Hauptbuch“ bestehen.

Von den betrachteten Blockchain-Modellen erfüllt das Hyperledger Fabric Blockchain-Framework mit seinen umfangreichen technischen Features am weitesten die Vorgaben der Datenschutzgrundverordnung, gefolgt von R3 Corda. Ethereum als öffentliche Blockchain erfüllt die Anforderung der DSGVO am wenigsten. Corda und MultiChain regeln den Datenschutz über den Zulassungsmechanismus und stellen sicher, dass der Zugriff auf die Transaktionen eingeschränkt werden kann. Bei Corda kann die Beschränkung der Sicht soweit herunter skaliert werden, dass nur noch die an der Transaktion direkt beteiligten Teilnehmer die Daten einsehen können. Fabric hingegen steuert den Zugriff über eine spezielle Kanalarchitektur (Channel). Die Kanäle in Hyperledger Fabric werden mit Zugriffsrichtlinien konfiguriert, die den Zugriff auf die Ressourcen des Kanals (Chaincode, Transaktionen und Ledger-Status) regeln, wodurch die Privatsphäre und Vertraulichkeit von Informationen innerhalb der Knoten im Kanal gewährleistet ist. Zusätzlich bietet Hyperledger Fabric auf einer weiteren, feineren Ebene die Form der „Privaten Transaktionen“. Die privat deklarierten Daten werden dabei in einer getrennten Datenbank gespeichert und im öffentlichen Hauptbuch über den entsprechenden Hash-Wert referenziert. Tatsächlich dienen die Hashes im öffentlichen Hauptbuch als überprüfbarer Beweis für die Daten. Dabei können die privaten Transaktionen zusätzlich mit einer anonymen Client-

Authentifizierung kombiniert werden, um zu vermeiden, dass die Verbindung zwischen der Identität des Erstellers der Transaktion und den gespeicherten Hash- Daten des Ledgers unterbrochen wird. Dem Datenschutz wird hierbei Rechnung getragen, indem die Kontrolle darüber besteht, wer auf die tatsächlich sensiblen Daten zugreifen kann. Die in Hyperledger Fabric als privat gespeicherten Daten können mit einer Gültigkeitsdauer (transient) versehen, sowie bearbeitet und gelöscht werden. Dies unterscheidet Fabric in einem entscheidenden Punkt von allen anderen Blockchain-Modellen.

In Artikel 18 der DSGVO wird die Einschränkung der Verarbeitung gefordert. In den privaten, zulassungsbeschränkten Blockchains können den Teilnehmern unterschiedliche Berechtigungsstufen zugewiesen werden, sodass der Zugriff eingeschränkt werden kann und Informationen zum Schutz der Vertraulichkeit zusätzlich verschlüsselt werden können. Bei öffentlichen Blockchains, wie z.B. Ethereum, stellt der freie Zugang sowie die absolute Transparenz auf Datenbankebene im Hinblick auf den Schutz von sensiblen und vertraulichen Informationen ein großes Problem dar. In der öffentlichen Blockchain Ethereum müssen sich alle Knoten auf ein gemeinsames Hauptbuch einigen und alle Knoten haben somit Zugriff auf alle Einträge. Es ist nicht möglich, Informationen nur einer bestimmten Teilnehmergruppe sichtbar zu machen.

Das Recht auf Datenübertragbarkeit in Artikel 20 der DSGVO fordert, dass sich die zu einer Person gehörenden Daten in einem strukturierten gängigen Format zusammenstellen lassen und der betreffenden Person zur Verfügung gestellt werden können. Diese Forderung lässt sich bei den privaten, zulassungsbeschränkten Modellen der Blockchain einfacher umsetzen als in den weit verteilten, großen Netzwerken der öffentlichen Blockchain-Varianten.

Modularität

Alle betrachteten Blockchain-Frameworks weisen eine modulare Architektur auf. Bei Hyperledger Fabric liegt eine besonders hohe modulare Struktur vor, die es ermöglicht, dass Entwickler jede Komponente (jedes Modul) für benutzerdefinierte Anforderungen ersetzen oder hinzufügen können, ohne den Rest des Systems zu beeinträchtigen. Gerade im Hinblick auf die vom Bundesamt für Informationstechnik (BSI)²⁶ geforderte Beachtung der Langzeitsicherheit bei der Blockchain-Architektur ist dies eine zielgerichtete Funktionalität.

Dokumentation / Community

Für die zuvor beschriebenen Blockchain-Modelle gibt es umfassende Informationen in Form von Anleitungen, Beiträgen, und Beispielen (z.B. Github) im World Wide Web. Alle zuvor beschriebenen Blockchain-Modelle verfügen alle über eine große und globale Community.

²⁶ Berghoff, C. et al. (2019). Blockchain sicher gestalten, Bundesamt für Sicherheit in der Informationstechnik (BSI) S. 61 f.. In: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Krypto/Blockchain_Analyse.pdf?__blob=publicationFile&v=5 (Zugriff: 16.08.2020)

5.4.6 Auswahl Blockchain-Modell für das Projektvorhaben

Nach Abwägung aller Kriterien kommt für das vorliegende Projektvorhaben nur das Blockchain-Framework „Hyperledger Fabric“ der Linux-Foundation in Frage. Fabric ist besonders im Bereich des Datenschutzes allen anderen Blockchain-Varianten überlegen.

Als private, zulassungsbeschränkte Blockchain regelt Fabric den Datenschutz über die Kombination von Kanälen (Channels) und privaten Transaktionen. Das Hauptkriterium für die Auswahl war dabei der integrierte Lösungsansatz von Fabric, personenbezogene Daten gem. Artikel 16 und 17 der DSGVO bearbeiten und löschen zu können. Keines der anderen zuvor aufgeführten Blockchain-Modelle bietet dieses Features. Da die zu prüfenden technischen Ansätze dieser Studie bei positiver Bewertung später in Unternehmen oder andere Organisationen Anwendung finden sollen und davon auszugehen ist, dass dabei auch „personenbezogene Daten“ zu Verarbeitung kommen, ist die Berücksichtigung der Anforderungen der Datenschutzgrundverordnung (DSGVO) unumgänglich. Die im Abschnitt „Prüfung rechtliche Anforderungen“ beschriebenen Ansätze zur Berichtigung bzw. Löschung von Daten innerhalb einer Blockchain, in Form von „Rollback-Verfahren“, Off-Chain-Datenspeicherung“, Einsatz von „Chameleon-Hashes“, Abspaltung der Blockchain (Fork) oder Mutable Blockchains, kommen als Lösungsansatz aufgrund der Komplexität und des Aufwands für das Projektvorhaben nicht in Frage.

Bei Fabric als privates Blockchain-Netz ist der Konsensmechanismus weniger zeit- und kostenintensiv als bei einer öffentlichen Blockchain-Variante, da hier kein aufwändiges Mining-Verfahren zum Einsatz kommt. Anzumerken ist hier, dass der Konsensmechanismus von Fabric, aus drei Schritten besteht (Ausführen, Reichenfolge, Validieren) und sich damit von allen anderen Blockchain-Modellen abgrenzt.²⁷ In Hyperledger Fabric können Kanäle genutzt werden, um die in CRM-Systemen häufig vorhandenen Benutzergruppen, wie z.B. Buchhaltung, Vertrieb, Verwaltung abzubilden.

Das Hyperledger Fabric Framework ist im Vergleich zu den privaten Blockchain-Modellen wie MultiChain oder Corda, hoch modular entwickelt. Das hat den Vorteil, dass Entwickler jede Komponente für benutzerdefinierte Anforderungen ersetzen oder hinzufügen können, ohne den Rest des Systems in irgendeiner Form zu beeinträchtigen. Inwieweit sich die hohe Modularität von Fabric bei der Umsetzung des Projektvorhabens auf den Umsetzungsaufwand und die Entwicklungszeiten auswirkt, zeigte sich im weiteren Verlauf der Studie.

Für die Erstellung der Smart Contracts (Chaincode), die für die Speicherung und das Lesen der sensiblen Daten (Transaktionen) benötigt werden, ist bei Fabric keine neue Programmiersprache zu erlernen. Es kann auf die gängigen Programmiersprachen wie Java, Go oder Node.js zurückgegriffen werden. Für diese Sprachen existieren bereits Werkzeuge und Entwickler, die diese Sprachen beherrschen. Da der komplette Funktionsumfang dieser Sprachen genutzt werden kann, ermöglicht dies den Einsatz von Bibliotheken, die nicht extra für die Nutzung in Fabric Chaincode entwickelt werden müssen. Dies entsprach einem der Kernziele des Projektvorhabens, den Aufwand für die Umsetzung der technischen Implementierung im Testverfahren und später bei der anvisierten Produktentwicklung niedrig zu halten.

Hyperledger Fabric ist nicht für eine bestimmte Branche entwickelt worden, wie z.B. Corda oder MultiChain, die schwerpunktmäßig für die Finanzbranche ausgelegt sind. Fabric wurde als Universallösung für Businessapplikationen auf Distributed-Ledger-Technologie für Unternehmen entwickelt. Hyperledger Fabric verfügt über eine sehr große Community und verspricht unter dem Schirm der Linux Foundation eine verlässliche und langfristige Weiterentwicklung. An der Entwicklung von Hyperledger Fabric beteiligen sich namhafte Unternehmen wie IBM, SAP, Digital Asset, Hyperchain und viele mehr. Das modular aufgebaute System ist Softwareentwicklern auf der ganzen Welt bekannt. Es wird nicht nur von Unternehmen, sondern auch von Regierungen und Organisationen genutzt. Als Open Source-Projekt liegt der gesamte Programmcode auf Github oder Gerrit. Das Backend für Blockchain-Angebote von IBM, Oracle und teilweise SAP basiert auf Hyperledger Fabric. Ein besonderes Merkmal von Hyperledger Fabric ist die Erfahrung der Linux Foundation in der Entwicklung und Wartung von großen und komplexen Open Source-

²⁷ Le Hors, A. (2019). Demistifying Hyperledger Fabric Ordering and decentralization. In: <https://developer.ibm.com/articles/blockchain-hyperledger-fabric-ordering-decentralization/> (Zugriff: 20.11.2020)

Projekten. Die Verankerung von Hyperledger Fabric unter dem Schirm der Linux Foundation verspricht eine verlässliche und langfristige Weiterentwicklung.

5.4.7 Hyperledger Fabric – Technische Details

Im Folgenden werden die technischen Details von Hyperledger Fabric näher beschrieben. Zum Zeitpunkt der Studie lag die aktuelle Version 2.3 von Hyperledger Fabric vor.²⁸

Die Architektur von Fabric beschreibt drei Services:

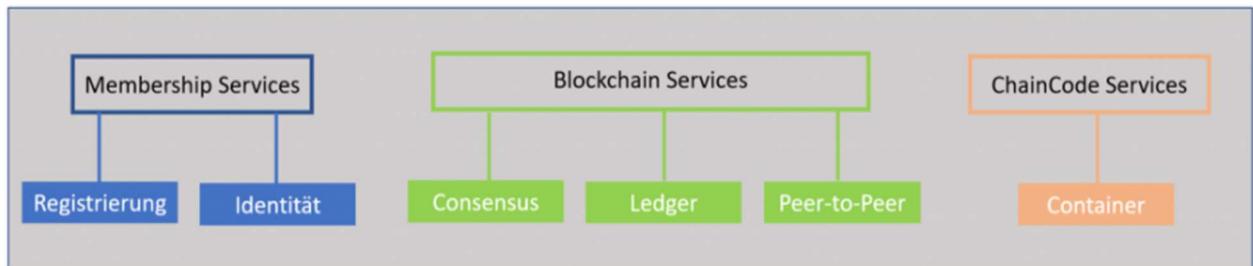


Abbildung 4: Komponenten von Hyperledger Fabric²⁹

In Hyperledger Fabric kann die Geschäftslogik (Chaincode) nur dann ausgeführt werden, wenn die beteiligten Mitglieder (Members) bekannt sind und eine fälschungssichere Identität haben. Der reibungslose Ablauf muss über die Membership Services sichergestellt werden. Die Hauptaufgabe dieser Services besteht in der Erzeugung und Zuweisung von Zertifikaten. Gleichzeitig wird gewährleistet, dass alle Transaktionen über sogenannte Transaktions-Zertifikate autorisiert werden können und einem Mitglied (Member) eindeutig zugewiesen sind. Diese Zertifikate werden in der Blockchain gespeichert und gewährleisten, dass Transaktionen über mehrere Blöcke verlinkt werden können. Die Aufgabe der Blockchain Services ist die Implementierung eines Peer-To-Peer Protokolls, damit die an der Blockchain beteiligten Knoten über das Internet miteinander kommunizieren können. Auf diesen Services können dann verschiedene Konsens-Protokolle aufsetzen. Über die Blockchain Services wird das Hauptbuch (Ledger) verwaltet. Die Chaincode-Services gewährleisten, dass der ChainCode (Smart Contract), in einer sicheren Umgebung ausgeführt wird. Technische Implementierungen hierfür sind Docker Container und Runtime Umgebungen wie Java, Go oder Node.js.

²⁸ Herrnberger, S. (2020). Blockchain-Entwicklungen auf Hyperledger Fabric v2.0 mit modularem Konsens In: <https://blockchainwelt.de/blockchain-entwicklungen-auf-hyperledger-fabric-v2-0-mit-modularem-konsens/> (Zugriff: 03.12.2020)

²⁹ <https://blog.dataone.de/2019/03/fabric-services-der-hyperledger-blockchain-verstehen/>

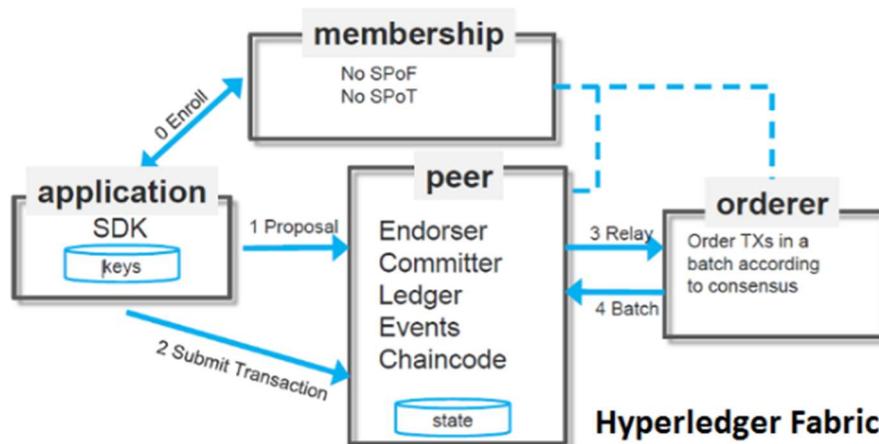


Abbildung 5: Hyperledger Fabric Infrastruktur³⁰

In Fabric laufen die Peer- und Orderer-Prozesse vollständig getrennt. Somit ist das Orderer-Netzwerk vom eigentlichen Datennetz getrennt.

Software Development Kits (SDK)

Hyperledger Fabric unterstützt sogenannte Software Development Kits (SDKs). Durch den Import der SDKs steht eine Konfiguration bereit, die Informationen wie MSP, TLS-Zertifikate, Peers und Endpunkte sowie Kanäle für Bestelldienste enthält.

Organisationen

Hyperledger Fabric verfügt über ein verteiltes Berechtigungskonzept, das jede Organisation selbst verwaltet. Eine Organisation in Fabric entspricht einer realen Organisation in einem Unternehmen oder einer sonstigen Vereinigung. Jede Organisation in Hyperledger Fabric muss über eine Public Key Infrastructure (PKI) verfügen, die aus einer Zertifizierungsstelle (CA) bestehen muss. Die Certificate Authority (CA) erstellt die digitalen Zertifikate für die Identitäten der Organisation (z. B. Benutzer, Clientanwendungen, Peers, Besteller (Orderer)). Es lässt sich aber auch jede bereits vorhandene PKI einer Organisation verwenden.

Mitgliedschaftsdienstleister (MSP)

Jede Organisation im Hyperledger Fabric Network wird anhand ihrer MSP-ID (Membership Service Provider Identification) identifiziert. Die Hyperledger Fabric Zertifizierungsstelle (CA, Certificate Authority) fungiert dabei als MSP-Dienstleister und erhöht die Sicherheit mithilfe von kryptografisch signierten Zertifikaten. Jede gelesene oder aktualisierte Transaktion muss mit einem Zertifikat signiert werden. Der Fabric CA-Server kann verwendet werden, um Zertifikate auszustellen und sie verschiedenen Entitäten des Netzwerks, wie Peer-Knoten oder Bestellknoten zuzuordnen oder zum Erstellen oder Registrieren eines Benutzers. Er wird durch eine Instanz einer PKI-Infrastruktur gebildet, die dann Zertifikate für den MSP ausstellen kann. Die Zertifizierungsstelle übernimmt neben der Registrierung von Mitgliedern, die Zertifizierung und Regulierung von Knoten. Alle Knoten, Benutzer und Clients verwenden digitale Zertifikate, um sich gegenseitig zu überprüfen und miteinander zu kommunizieren. Daher spielen Zertifizierungsstellen eine zentrale Rolle, um Identitäten im Netzwerk zu verfolgen und veraltete Konten zu eliminieren. Der Hyperledger Fabric CA-Server kommuniziert mit dem Fabric-Netzwerk über REST-APIs, die ebenfalls entweder mit einem Fabric SDK interagieren oder mit einer speziellen Instanz des Fabric CA-Clients. Jede Komponente in Hyperledger Fabric hat eine Identität. Je nach Identität können die genauen Berechtigungen und die Rolle der Komponente

³⁰ <https://de.0xzx.com/2019060913246.html> (Zugriff: 03.12.2020)

festgelegt werden. Fabric benötigt von allen Transaktionen ein kryptografisches Zertifikat, das die vertraulichen Daten eines Benutzers beinhaltet und im Netzwerk registriert ist. Die Identitäten der Transaktionspartner sind verborgen, um die Privatsphäre innerhalb des Netzwerks zu sichern.

Client

Clients sind Anwendungen, die im Auftrag eines Benutzers Transaktionen an das Netzwerk senden.

In Hyperledger Fabric existieren verschiedene Knoten als logische Entitäten:

Peer (Knoten)

Die Validierung und Aktualisierung der Blockchain-Daten erfolgt über sogenannte Peers. Auf einem Peer ist das Hauptbuch (Ledger) gespeichert. Somit verwaltet jeder Peer eine eigene Kopie des gemeinsam genutzten Hauptbuchs. Ein Peer kann auch über mehrere Ledger verfügen, indem er an mehreren Kanälen (Channels) beteiligt ist. Transaktionen innerhalb eines Kanals werden an alle Peers verteilt, um sie in einem vertrauenswürdigen und konsistenten Zustand zu halten. Auf den Peers können mehrere Versionen eines Chaincodes installiert und gespeichert sein und in einem oder mehreren Kanälen instanziiert werden. Anwendungen stellen eine Verbindung zu Peers her, um Daten aus dem Hauptbuch abzurufen oder abzulegen. Die Peers in Fabric müssen bei der zur Installation und Inbetriebnahme von Chaincode zustimmen.

Endorser Peer

In Fabric können Peers als „Endorser“ tätig sein, um die von einem Client angestoßene Transaktion zu simulieren und zu überprüfen. Endorser simulieren die Transaktionsausführung auf isolierten Containern. Ausgehend von dieser Simulation erstellt der Endorser einen Transaktionsvorschlag, den er dann an den aufrufenden Peer sendet. Endorsing Peers haben die Berechtigung, Chaincode auszuführen, der Teil des Konsensmechanismus von Hyperledger ist. In Hyperledger Fabric wird eine Endorser-Richtlinie (Endorsement-Policy) definiert, um einen Peer anzuweisen, einen Chaincode auszuführen. Dabei wird der Validation System Chaincode (VSCC) aufgerufen, um die Transaktion zu validieren. Die Endorsement-Richtlinie wird über die Boolesche Logiksyntax über Principals definiert werden. Fabric verfügt somit über einen hohen Sicherheitsstandard. Anders als bei anderen Blockchains benötigen fast alle Schritte in Fabric eine entsprechende Policy oder Berechtigung, weshalb von einem hohen Sicherheitsstandard auszugehen ist.

Besteller (Orderer)

Der Bestellservice kann aus einem oder mehreren Bestellknoten bestehen. Die Knoten empfangen gleichzeitig Transaktionen von verschiedenen Anwendungsclients. Diese Bestellknoten arbeiten zusammen, um gemeinsam den Bestellservice abzuwickeln. Ihre Aufgabe ist es, Stapel übermittelter Transaktionen in einer genau definierten Reihenfolge anzuordnen und in Blöcke zu packen. Die Peers müssen diese Reihenfolge beim Validieren und Festschreiben von Transaktionen verwenden. Bei anderen Blockchains (z.B. Bitcoin oder Ethereum) kann dieselbe Transaktion in mehrere, verschiedene Blöcke gepackt werden, die dann miteinander konkurrieren, um die Blockchain zu erweitern. Die von Fabric erstellten Blöcke sind endgültig, d.h. es werden keine validierten Transaktionen jemals zurückgesetzt oder gelöscht. Dadurch können bei Fabric keine sogenannten „Forks“ entstehen. In der aktuellen Version von Fabric 2.3 ist das Raft-Protokoll für die Bestelldienste als Standard implementiert. Optional kann auch das Solo- oder Kafka-Protokoll aus früheren Versionen von Fabric eingesetzt werden. Beim Raft-Protokoll wird ein „Leader und Follower“-Modell umgesetzt, bei dem ein Leader dynamisch unter den Bestellern in einem Kanal ausgewählt wird. Dieser Leader repliziert die Vorgehensweise über Nachrichten an die Follower-Besteller. Durch den Einsatz von mehreren Bestellern (Orderer) kann das System den Verlust von Bestell-Knoten (z.B. durch technischen Ausfall etc.) aushalten, solange die Mehrheit der Bestell-Knoten weiter zur Verfügung steht. Fällt z.B. ein Bestell-Knoten von insgesamt 3 Bestell-Knoten aus (zwei bleiben übrig = Mehrheit), so bleibt der Bestellservice

intakt. Dieser Mechanismus des Raft-Protokolls wird als „Crash Fault Tolerant“ (CFT) bezeichnet. Auch das Kafka-Protokoll arbeitet nach dem CFT-Mechanismus.

Committer Peer

Die Committer Peers sind verantwortlich für die Aufrechterhaltung der Blockchain- und Ledger-Struktur und für die Festschreibung der Transaktionen. In regelmäßigen Abständen erhalten sie die sortierte Batch-Transaktionsblockstruktur vom Besteller (Orderer) und führen eine letzte Überprüfung dieser Transaktionen durch, bevor die Bestellung veranlasst wird.

Anker Peer

Ein Anker Peer dient als Vermittler zwischen Peers aus der Organisation und Peers aus einem externen Unternehmen. Er stellt eine Peer-Rolle in einem Kanal dar, die von allen anderen Peers in allen Organisationen erkannt werden kann. Verfügt eine Organisation über keinen Anker Peer, können ihre Peers nur die Peers ihrer internen Organisation sehen.

Leader Peer

Die Leader Peers übernehmen die Verantwortung für die Verteilung der Transaktionen vom Besteller (Orderer) an die Committer Peers.

Smart Contract (Chaincode)

Der Anwendungscode „Smart Contract“, wird bei Fabric „Chaincode“ genannt. Er ist ein Anwendungscode, der die Logik zum Aufrufen von Transaktionen darstellt. Im herkömmlichen Sinne kann der Chaincode als eine Art „Datenbank-Trigger“ oder „Stored Procedure“ verstanden werden. Fabric unterscheidet in User- und System-Chaincode. Über den User-Chaincode der auf den Peers (Docker-Container) ausgeführt wird, können die Daten aus dem Hauptbuch gelesen und aktualisiert werden. Fabric unterstützt beim User-Chaincode gängige Programmiersprachen wie z.B. Java, Go und Node.js. Der Chaincode muss Teil des Kanals (Channel) sein und kann nur auf das Hauptbuch dieses Kanals angewendet werden. Um einen komplexen Prozess abzubilden, können Peers eines Kanals mehrere Chaincodes verwenden. Der Chaincode wird paketiert und nach Installation auf den Peers ist er anschließend zu instanzieren. Die Instanziierung umfasst eine Verknüpfung einer bestimmten Chaincode-Version mit dem Kanal. Chaincode kann auf einem Peer in mehreren Versionen existieren. Die Chaincode-Instanziierung erfordert eine Endorsement-Richtlinie, die vorschlägt, welcher Peer die Chaincode-Transaktionen unterstützen kann. Dabei kann jeder Chaincode eine eigene Richtlinie haben.

User-Chaincodes haben vier Lebenszyklen:³¹

1. Installieren: Ordnet dem Chaincode einen Ort oder Pfad zu
2. Instanzieren: Ein Docker-Container-Image wird erstellt (Objekterzeugung)
3. Aufrufen: Schreiben und Einfügen von Chaincode, Einfügen der Daten auf der Blockchain
4. Abfragen: Abrufen und Erhalten von Daten aus der Blockchain

Die System-Chaincodes umfassen Logik, die als Teil des Peer-Prozesses ausgeführt wird. Sie haben mehr Zugriff auf Ressourcen im Peer und können zum Implementieren von Funktionen verwendet werden. Beispiele für System-Chaincodes sind QSCC (Query System Chaincode) für Ledger- und andere Fabric bezogene Abfragen, CSCC (Configuration System Chaincode) zur Regulierung der Zugriffskontrolle und LSCC (Lifecycle System Chaincode). Die System-Chaincodes werden vom Peer beim Start registriert und bereitgestellt.

³¹ Herrberger, S. (2020). Blockchain-Entwicklungen auf Hyperledger Fabric v2.0 mit modularem Konsens In: <https://blockchainwelt.de/blockchain-entwicklungen-auf-hyperledger-fabric-v2-0-mit-modularem-konsens/> (Zugriff: 17.12.2020)

Kanal (Channel)

Für die private Kommunikation der Parteien des Unternehmensnetzwerks bietet Fabric die spezielle Struktur der Kanäle (Channels) an.³² In den Kanälen sind die Transaktionen nur für ihre Teilnehmer sichtbar. Jeder Kanal ist unabhängig von einem anderen und verfügt über eigene Sicherheitsrichtlinien und einen eigenen Chaincode. Peers (Knoten) können an mehreren Kanälen beteiligt sein. Die Channel-Teilnehmer (Peers) müssen sich gegenseitig kennen. Nur autorisierte Peers können daher Teil des Kanals sein. Jeder Kanal kann als unabhängige Blockchain betrachtet werden. Ein Kanal verwaltet sein eigenes Hauptbuch (Transaktionen) und seine eigene Konfiguration.

Verteiltes Hauptbuch (Ledger)

In Hyperledger Fabric verwalten Peers nach dem Beitritt zu einem Kanal eine Kopie des Hauptbuchs (Ledger).³³ Im Hauptbuch werden alle Details von Transaktion, die jemals in einem Fabric-Kanal stattgefunden haben, aufgezeichnet. In Fabric besteht das Hauptbuch aus zwei Teilen. Der erste Teil ist eine Blockchain-Datenstruktur, die die Blöcke (von Transaktionen) enthält. Der zweite Teil umfasst eine „Worldstate-Datenbank“, die den neuesten Status abbildet, nachdem ein Block festgeschrieben wurde. Die Peers schreiben die vom Bestellservice erhaltenen neuen Blöcke nach erfolgreicher Validierung in das Hauptbuch. Dies umfasst das Übernehmen des Blocks in das Hauptbuch und das Aktualisieren des Worldstates. Die Blöcke im Hauptbuch führen nicht nur Aufzeichnungen über alle Transaktionen, die bestätigt wurden, sondern auch über diejenigen, die während des Bestätigungsprozesses abgelehnt wurden.

Private Daten / Kanal

Bei Hyperledger Fabric werden in einem Kanal alle Daten von allen Kanalteilnehmern gemeinsam genutzt. Ein Teil dieser Daten kann als „private Daten“ über die Private Data Collection von Fabric verwaltet werden, ohne einen weiteren Kanal zu erstellen.

Der Zugriff auf diese Daten ist per Definition beschränkt. Die Sichtbarkeit der Daten in einem Kanal kann dadurch weiter eingeschränkt werden. Die privaten Transaktionen/Daten werden neben dem Hauptbuch (Ledger) in einer separaten Datenbank gespeichert. Diese Datenbank wird zusammen mit dem öffentlichen Hauptbuch aktualisiert, wenn Transaktionen mit Verweisen auf private Daten festgeschrieben werden. Nicht autorisierte Teilnehmer haben einen Hash der privaten Daten im Hauptbuch als Nachweis für die Transaktionsdaten. Aus Gründen des Datenschutzes werden Hashes der privaten Daten über den Bestellservice und nicht über die privaten Daten selbst gesendet, sodass private Daten vor dem Besteller vertraulich behandelt werden. Über spezielle Einstellungen kann der Verfügbarkeitszeitraum der privaten Daten definiert werden. Private Transaktionen sollten mit Vorsicht verwendet werden, wenn das Muster der Aktualisierung privater Daten auch vertrauliche Informationen beinhaltet und zur Ableitung der tatsächlichen privaten Daten verwendet werden kann. Private Transaktionen können z.B. mit einer anonymen Clientauthentifizierung kombiniert werden, um zu vermeiden, dass die Verbindung zwischen der Identität des Erstellers der Transaktion und den gespeicherten Hash-Daten des Ledgers offensichtlich wird.

³² Androulaki, E. et al. (2018). Private and confidential transactions with Hyperledger Fabric. <https://developer.ibm.com/technologies/blockchain/tutorials/cl-blockchain-private-confidential-transactions-hyperledger-fabric-zero-knowledge-proof/> (Zugriff: 14.12.2020)

³³ <https://www.oak-tree.tech/blog/hyperledger-overview> (Zugriff: 18.12.2020)

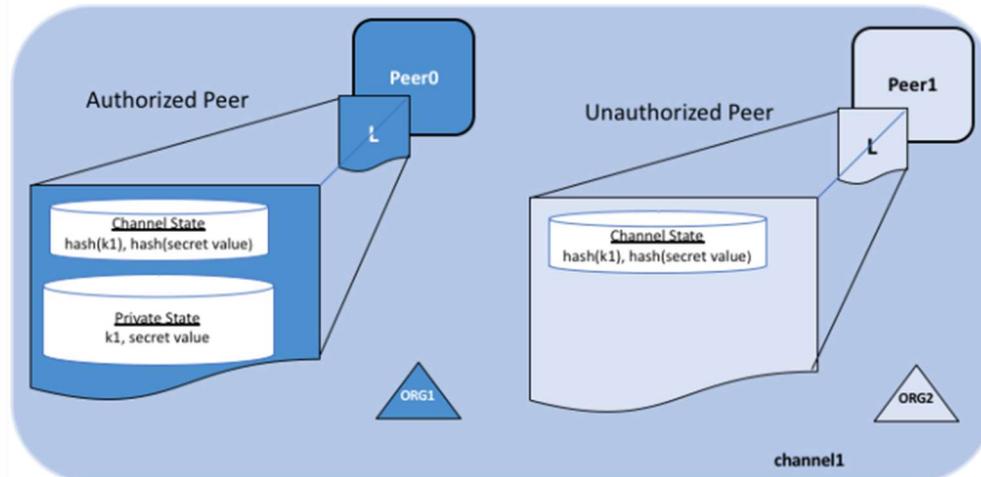


Abbildung 6: Speicherung privater Daten in Hyperledger Fabric³⁴

Konsens

In Hyperledger Fabric erfolgt der Konsens in einem dreistufigen Prozess. Zuerst erfolgt das Endorsement, dann das Ordering (Bestellung) und im Anschluss die Validation.³⁵ Beim Ordering unterstützt Hyperledger Fabric derzeit drei verschiedene Mechanismen oder Implementierungen von Konsens - SOLO, Kafka und Raft. In der Version Hyperledger Fabric 2.0 ist „Raft“ der empfohlene Konsens-Service und als nativ implementiert. Nur Raft ermöglicht einen Bestellservice, der aus Knoten verschiedener Organisationen besteht. Raft basiert auf dem CFT-Konsensmechanismus (Crash Fault Tolerant) und fungiert als Leader mit Followern. Ab Version 2.0 ermöglicht das „pluggable consensus protocol“ die Plattform effektiv an bestimmte Anwendungsfälle und Vertrauensmodelle (Konsens) anzupassen. In Abhängigkeit des Anwendungsfalles kann ein fehlertoleranter byzantinischer Konsens (Byzantine Fault Tolerance BFT) nötig sein. Der BFT Konsens ist weitaus komplexer und wird vor allem in Systemen angewendet, bei denen man davon ausgehen muss, dass auch nicht vertrauenswürdige Teilnehmer im Netzwerk vorhanden sind.

Transaktionsfluss in Hyperledger Fabric

Um einen Konsens zu erzielen, verwendet Hyperledger Fabric einen Ablauf, der auf Berechtigungen basiert. Die Endorsement-Richtlinie definiert dabei den Ablauf, wie die einzelnen Peers verwendet werden sollen, und welches Gewicht jeder Peer in Bezug auf die Gültigkeit einer Transaktion erhält.

Der Transaktionsfluss in Hyperledger Fabric umfasst folgende Teilschritte:

1. **Vorschlag Transaktion:** Ein Client erstellt einen Transaktionsvorschlag und sendet ihn, wie in der Endorsement-Richtlinie definiert, an Endorsement-Peers. Der Vorschlag enthält Informationen zur Identität des Antragstellers und eine Transaktionskennung.
2. **Ausführen (Bestätigung):** Die Endorser simulieren die Transaktion. Dabei erstellen sie einen Schreibratz, der die Schlüssel und ihre geänderten Werte enthält, sowie einen Lesesatz. Sie überprüfen auch die Richtigkeit der Transaktionsausführung. Das Ergebnis wird als Antwort auf den Vorschlag gesendet und enthält den Schreibratz, den Lesesatz, die Transaktions-ID, die Endorser-ID und die Unterschrift des Endorsers. Wenn der Client ausreichend Antworten von Endorsern erhalten hat, die dasselbe Ausführungsergebnis enthalten, leitet er die Transaktion an den Bestellservice (Orderer) weiter.

³⁴ Private Data. In: <https://hyperledger-fabric.readthedocs.io/en/release-2.2/private-data/private-data.html> (Zugriff: 04.12.2020)

³⁵ Herrnberger, S. (2020). Blockchain-Entwicklungen auf Hyperledger Fabric v2.0 mit modularem Konsens In: <https://blockchainwelt.de/blockchain-entwicklungen-auf-hyperledger-fabric-v2-0-mit-modularem-konsens/> (Zugriff: 17.12.2020)

3. **Bestellung:** Der Bestellservice (Orderer) prüft zunächst, ob der Client, der den Transaktionsvorschlag gesendet hat, auf einem bestimmten Kanal über die entsprechenden Berechtigungen verfügt. Im Anschluss erzeugt der Bestellservice Blöcke, die indossierte Transaktionen in einer geordneten Reihenfolge pro Kanal enthalten. Durch die Bestellung kann das Blockchain-Netzwerk einen Konsens erhalten. Nach der Blockerstellung sendet der Bestellservice die Blöcke entweder an definierte Peer-Leader oder direkt an alle Peers.
4. **Validieren:** Jeder Peer validiert die empfangenen Transaktionen, indem er prüft, ob eine Transaktion der entsprechenden Endorsement-Richtlinie entspricht. Danach wird nacheinander eine Lese- / Schreibkonfliktprüfung für alle Transaktionen im Block vorgenommen. Für jede Transaktion werden die Versionen der Schlüssel im Lesesatz mit denen verglichen, die sich aktuell im Hauptbuch befinden. Es wird geprüft, ob die Werte gleich sind. Falls sie nicht übereinstimmen, werden die Transaktionen verworfen. Im Anschluss wird das Ledger aktualisiert, in dem es den erstellten Block an seine Kette anfügt. Die Ergebnisse der Gültigkeitsprüfungen einschließlich der ungültigen Transaktionen werden im Ledger ebenfalls gespeichert.

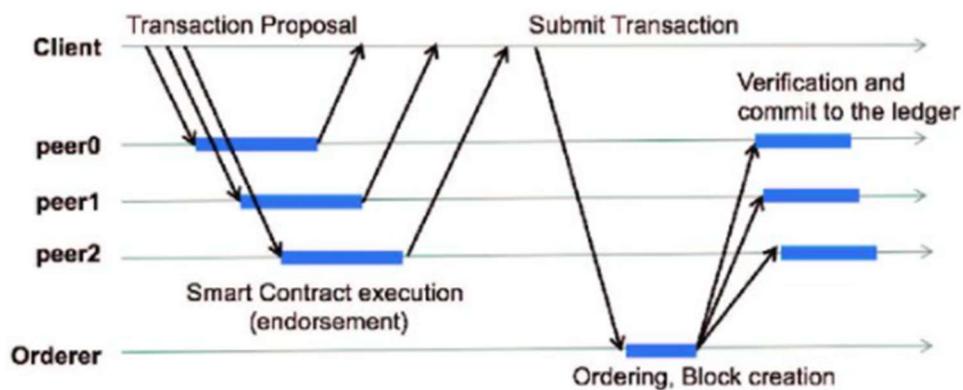


Abbildung 7: Konsensablauf von Hyperledger Fabric³⁶

5.5 Installation, Test Blockchain-Modell

Aufsetzen der Hyperledger Fabric Blockchain auf einem Verbund eigener Rechner/Server

Im Nachfolgenden werden die Schritte für das Aufsetzen der Hyperledger Fabric Blockchain auf einem Verbund eigener Rechner und Server beschrieben.

1. Rechner- und Server-Setup

Zum Betrieb der Hyperledger Fabric Instanz benötigte jeder Rechner und Server initial einen Docker-Container und die entsprechenden Images. Zuvor waren einige Applikationen zu installieren. Dazu zählten z.B. GO, windows-build-tools sowie Docker-Composer und Docker. Auf den Systemen war die Installation des Dockers nach Download des „Installers“ und dessen Ausführung erledigt. Anschließend mussten im globalen Dockerhub die Images für Hyperledger gesucht und hinzugefügt werden:

³⁶ Yoshihama, S. et al. (2018). Studie zu Integritäts- und Datenschutzanforderungen verteilter Hauptbuchtechnologien. In: https://www.researchgate.net/publication/328334995_Study_on_Integrity_and_Privacy_Requirements_of_Distributed_Ledger_Technologies. (Zugriff: 04.12.2020)

- Hyperledger/fabric-tools
Hierzu zählen alle Funktionen des hyperledger command line interfaces (cli).
- Hyperledger/fabric-peer
Hierzu zählen alle Funktionen des Peers.
- Hyperledger/fabric-orderer
Nahezu alle Funktionalitäten des Orderers in diesem Image zusammen.
- Hyperledger/ccenv
Setzt eine Umgebung für Chaincodes (CC) auf und bietet die Möglichkeiten für den Lifecycle Prozess, zu dem auch die Installation, Instanziierung und das Endorsement von Chaincode zählen.
- Hyperledger/fabric-java-env
Da der Chaincode mit Java entwickelt werden sollte, wurde ebenfalls ein Java-Environment zur Installation und Instanziierung des Chaincodes benötigt.

2. Erstellung von Krypto-Materialien / Membership-Registrierung (Public Key Infrastruktur)

Hyperledger arbeitet mit einer PKI (Public Key Infrastructure), die eine Certificate Authority (CA) bereitstellt. Die Certificate Authority (CA) ist eine „Organisation“, die für Nutzer ein Zertifikat bereitstellt und zugleich verwaltet.

Ablauf:

- Der Nutzer erstellt einen Zertifikats-Request den er an die CA sendet. Diese erstellt mit Angabe von Parametern und einem private Key ein Zertifikat und sendet dieses zurück.
- Die CA legt das Zertifikat in ihrer internen Struktur ab.
- Eine Organisation, die mit dem Partner handeln möchte, holt sich dieses Zertifikat und kann die Daten, die der Nutzer gesendet hat mithilfe des Zertifikats validieren.

Gleichzeitig verwaltet die CA auch die Informationen darüber, ob ein Zertifikat bereits invalide geworden ist. Dazu verwaltet sie eine sogenannte „Revocation List“. Der im Internet wohl am häufigsten genutzte Zertifikatsstandard ist X.509, der auch in Hyperledger Fabric Anwendung findet.

Da zunächst weder eine CA noch sonstige Service-Provider bekannt sind, mussten diese zunächst erzeugt werden. Hierzu diente ein Tool, das Hyperledger als „Binary“ ausliefert: „Cryptogen“. Über dieses Tool war es möglich, über eine vorher erstellte Datei, die die Anzahl der Peers und Nutzer definiert, die entsprechenden Zertifikate zu erzeugen.

Erstellt wurden hierbei:

Peer-Zertifikate, 1 User-Zertifikat und standardmäßig pro Peer ein Admin-Zertifikat. Auch die CA erhielt das entsprechende „Root-Zertifikat“, um später weitere Nutzer registrieren zu können.

```
version: '2'

networks:
  behave:

services:
  peer0.org1.example.com:
    extends:
      file: docker-compose-base.yml
      service: peer
    container_name: peer0.org1.example.com
    environment:
      - CORE_PEER_CHAINCODELISTENADDRESS=peer0.org1.example.com:7052
      - CORE_PEER_ID=peer0.org1.example.com
      - CORE_PEER_ADDRESS=peer0.org1.example.com:7051
      - CORE_PEER_GOSSIP_BOOTSTRAP=peer1.org1.example.com:8051 peer9.org1.example.com:16051
      - CORE_PEER_GOSSIP_EXTERNALENDPOINT=peer0.org1.example.com:7051
      - CORE_PEER_GOSSIP_ORGLEADER=${CORE_PEER_GOSSIP_ORGLEADER_PEER0_ORG1}
      - CORE_PEER_GOSSIP_USELEADERELECTION=${CORE_PEER_GOSSIP_USELEADERELECTION_PEER0_ORG1}
      - CORE_PEER_LOCALMSPID=Org1MSP
      - CORE_PEER_TLS_CLIENTROOTCAS_FILES=/var/hyperledger/users/Admin@org1.example.com/tls/ca.crt
      - CORE_PEER_TLS_CLIENTCERT_FILE=/var/hyperledger/users/Admin@org1.example.com/tls/client.crt
      - CORE_PEER_TLS_CLIENTKEY_FILE=/var/hyperledger/users/Admin@org1.example.com/tls/client.key
    volumes:
      - ../crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/msp:/var/hyperledger/msp
      - ../crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls:/var/hyperledger/tls
      - ../crypto-config/peerOrganizations/org1.example.com/users:/var/hyperledger/users
      - ../network-config:/var/hyperledger/configs
      - ../blockchainData/peer0:/var/hyperledger/production
    extra_hosts:
      - "orderer0.example.com:10.11.12.60"
      - "orderer1.example.com:10.11.12.60"
      - "orderer2.example.com:10.11.12.60"
      - "ca.example.com:10.11.12.60"
      - "peer1.org1.example.com:10.9.61.135"
      # - "peer2.org1.example.com:10.9.61."
      # - "peer3.org1.example.com:10.9.61."
      # - "peer4.org1.example.com:10.9.61."
      # - "peer5.org1.example.com:10.9.61."
      # - "peer6.org1.example.com:10.9.61."
      # - "peer7.org1.example.com:10.9.61."
      # - "peer8.org1.esqmp1e.com:10.9.61."
      - "peer9.org1.example.com:10.11.12.60"

  networks:
    behave:
      aliases:
        - ${CORE_PEER_NETWORKID}
  ports:
    - 7051:7051
    - 7053:7053
```

Abbildung 8: Beispiel-Code für die Erstellung eines Docker-Compose File

3. Peer Einrichtung auf Client-PC und -Server

Anschließend erfolgte die Installation der Orderer, CA und Peers auf den jeweiligen Client-Rechnern. Für das vorliegende Projekt wurden sowohl drei Orderer + Kafka-Clients als auch die Certificate Authority (CA) sowie ein Peer auf einem zentralen Server installiert.

Es wurde der Ansatz verfolgt, eine maximale Sicherheit zu gewährleisten, weshalb alle Endorser Peers bei der ersten Teststellung auf die Client-Rechner verteilt wurden.

Alle weiteren Peers wurden auf den jeweiligen Rechnern installiert. Die Installation der Peers erfolgte durch das Kopieren der entsprechenden Config-Docker-Files, die nachdem sie mit „Port“ und „Domain“ aktualisiert wurden mit dem Befehl: `docker-compose -f docker-compose-peerX.yml up -d` gestartet und dabei erstellt wurden. Pro Client Rechner wurde jeweils eine CLI und der entsprechende Peer an sich erstellt. Nachdem alle Peers installiert waren, wurde der Channel, definiert und dem Orderer bekannt gemacht. Im Anschluss daran konnten alle Peers dem Channel beitreten.

Hierbei ist zu beachten, dass ein Channel durch ein vorher erzeugtes „channel.tx file“ erzeugt wird. „Tx“ ist in diesem Fall eine Transaktionsdatei, die bei der Generierung des Channels am Orderer in den Block geschrieben wird. Damit ein Peer diesem Channel beitreten kann, muss dieser diesen den Block zunächst einmal vom Orderer ziehen („fetchen“) und anschließend mit diesem eine Channel-Join-Transaktion erstellen. Erfolgt dies alles korrekt, ist der Peer mit Status 200 dem Channel hinzugefügt.

Während seiner „Up-Time“ kommuniziert der Peer über das „GOSSIP-Protokoll“ mit anderen Peers, und tauscht so Informationen darüber aus, ob z.B. gerade ein Peer das Netzwerk verlassen hat. Die Blockgröße für das Ledger muss in den Orderer-Peers festgelegt und beibehalten werden. Eine spätere Änderung der Blockgröße ist nicht möglich. Eine nachträgliche Änderung der Blockgröße hätte zur Folge, dass das Hauptbuch „invalide“ wird.

4. Anchor-Peers festlegen

Zusätzlich wurden in diesem Schritt durch ein Channel-Update die Anchor-Peers des Netzwerkes festgelegt. Anchor-Peers haben einen besonderen Rang, indem Sie über Informationen verfügen, die Auskunft darüber geben, welche Peers sich gerade im Netz befinden, aber auch welche Peers Chaincode „endorsen“ dürfen. Die Anker-Peers werden auch für den Einsatz der Endorsement-Policy benötigt, damit die anfragenden Peers die Endorser-Peers finden. In der Endorsement-Policy wird festgelegt wie viele Endorser von allen verfügbaren Endorser für die Freigabe der Transaktion mindestens zustimmen müssen (z.B. mindestes 3 Endorser von 5).

5. Chaincode Entwicklung und Installation

Die Entwicklung des Chaincodes kann in der Programmiersprache Java, Go oder Node.js erfolgen. Dies geschieht am besten über eine IDE (Integrated Development Environment), wie Visual Studio Code, um mithilfe des IBM-Blockchain-Plugins den Chaincode bereits innerhalb einer vorgefertigten Struktur zu implementieren. Der Chaincode läuft in einem eigenen Docker-Container, was den Vorteil hat, dass die komplette Businesslogik in diesem gekapselt ist und damit unabhängig vom Peer läuft. Um den Chaincode für den Endorser Peer zu installieren sind mehrere Schritte nötig. Als erstes muss das Projekt-Directory mit „Gradle“ bereits im Vorfeld ausgestattet worden sein. Durch einen Kommandozeilen-Befehl wird dieser Ordner in ein ZIP konvertiert bzw. verpackt. Anschließend lässt sich der Chaincode auf dem entsprechenden Peer installieren. Ist dieser erfolgreich auf allen „Endorsern“ definiert, kann dieser jeweils pro Organisation „approved“ werden. Im Anschluss kann der Chaincode „committed“ werden. Erst in diesem Status ist der Chaincode im Channel und zwischen den Organisationen bekannt und kann Key-Value-Paare auf dem Ledger erzeugen. Im Docker zeigt sich nun die Existenz des Chaincode-Containers. Abschließend muss der Chaincode der Peers auf den Channels noch instanziiert werden.

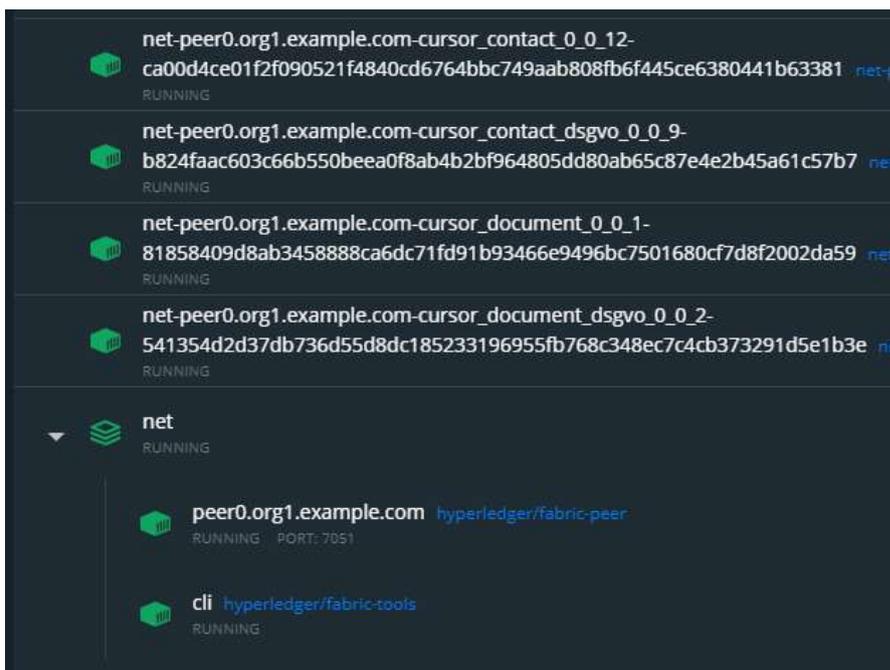


Abbildung 9: Übersicht der laufenden Docker-Container

```
58
59 @Transaction()
60 public String updateCrmDocument(Context ctx, String documentPk) {
61
62     Map<String, byte[]> map = ctx.getStub().getTransient();
63     byte[] content = map.get("content");
64
65     char[] password = new String(keyBytes, StandardCharsets.UTF_8).toCharArray();
66     String fileName = new String(map.get("fileName"));
67     String fileType = new String(map.get("fileType"));
68
69     CRMDocument asset = new CRMDocument();
70     try {
71         asset.setContent(new String(content.toString()));
72         asset.setFileName(fileName);
73         asset.setFileType(fileType);
74     } catch (Exception e) {
75         return "{\"result\":\"ERROR during encryption\"}";
76     }
77
78     ctx.getStub().putState(documentPk,
79         asset.toJSONString().getBytes(UTF_8));
80
81     return "{\"result\":\"The asset " + documentPk
82         + " was updated\"}";
83
84 }
```

Abbildung 10: Beispiel für eine Chaincode-Implementierung

In der obigen Abbildung wird beispielhaft ein Chaincode gezeigt, der zum Speichern von Dokumenten benötigt wird. Dabei werden Dateiname, Datei-Typ und Datei-Inhalt gespeichert. Genutzt wird hier als Übergabe „transient Data“, wodurch sichergestellt ist, dass die Daten nicht direkt im Ledger sichtbar sind.

Nach Ausführung der zuvor beschriebenen Schritte sind die Grundstrukturen des Hyperledger Fabric Netzwerks soweit aufgesetzt. Dazu zählen ein Peer-to-Peer Netzwerk, ein Channel und der Chaincode zur Anlage, zum Auslesen und zum Updaten von Daten in Form von Key-Value-Paaren.

Anlage „Private Data Collection“ in Hyperledger Fabric

Die Private Data Collection von Hyperledger Fabric lässt sich über den „Lifecycle Chaincode“ installieren und updaten. Fabric erstellt dafür selbständig im Hintergrund die entsprechenden Docker-Container. Anders als bei normalen Ledger-Daten besitzen Private Data Assets immer auch eine „Collection“. Eine Collection ist ähnlich zu einer „Policy“, jedoch regelt diese nicht die Zustimmung, sondern wer etwas in dem jeweiligen Chaincode ändern oder hinzufügen darf. So könnte z.B. definiert werden, dass nur ein bestimmter User der Organisation A die Daten lesen oder bearbeiten darf.

```
[
  {
    "name": "CollectionOne",
    "policy": {
      "identities": [
        {
          "role": {
            "name": "member",
            "mspId": "Org1MSP"
          }
        }
      ],
      "policy": {
        "l-of": [
          {
            "signed-by": 0
          }
        ]
      }
    },
    "requiredPeerCount": 1,
    "maxPeerCount": 1,
    "blockToLive": 0,
    "memberOnlyRead": true,
    "memberOnlyWrite": true
  }
]
```

Abbildung 11: Auszug aus einer Private Data Collection

In der obigen Abbildung ist beispielhaft der Auszug aus einer Collection zu sehen. Diese beginnt initial immer mit der Definition eines Namens. Danach erfolgt eine Auflistung der Peers, die das Private-Data-Feld halten darf. Im obigen Beispiel sind alle Identities erlaubt, die die Rolle „Member“ der Organisation Org1MSG besitzen.

Danach folgten weitere Einstellungsfelder:

requiredPeerCount: Bestimmt die minimale Anzahl an Peers innerhalb der Organisationen, an die der Endorser die Transaktion senden muss, ehe die Transaktion signiert wird. Der Status „0“ bedeutet, dass keine Verteilung nötig ist, sie allerdings trotzdem durchgeführt werden kann, wenn $\text{maxPeerCount} > 0$ ist.

maxPeerCount: Jeder autorisierte Endorsing Peer wird versuchen, die Transaktion an die festgelegte Anzahl zu verteilen.

BlockToLive: Bei der BlockToLive handelt es sich um ein DSGVO Feature. Es setzt fest, wie lange eine Information in Blöcken gehalten wird.

memberOnlyRead / memberOnlyWrite: Als weiteres Security Feature können hier die Lese- und Schreibrechte einer Organisation und dessen Clients festgelegt werden.

Technisch gesehen verändert sich im Gegensatz zu „normalen“ Ledger Daten nichts. Angemerkt werden muss jedoch, dass eine Nutzung von Parametern für die einzelnen Methoden dazu führt, dass diese Information im Block zugehörig zur Transaktion gespeichert wird.

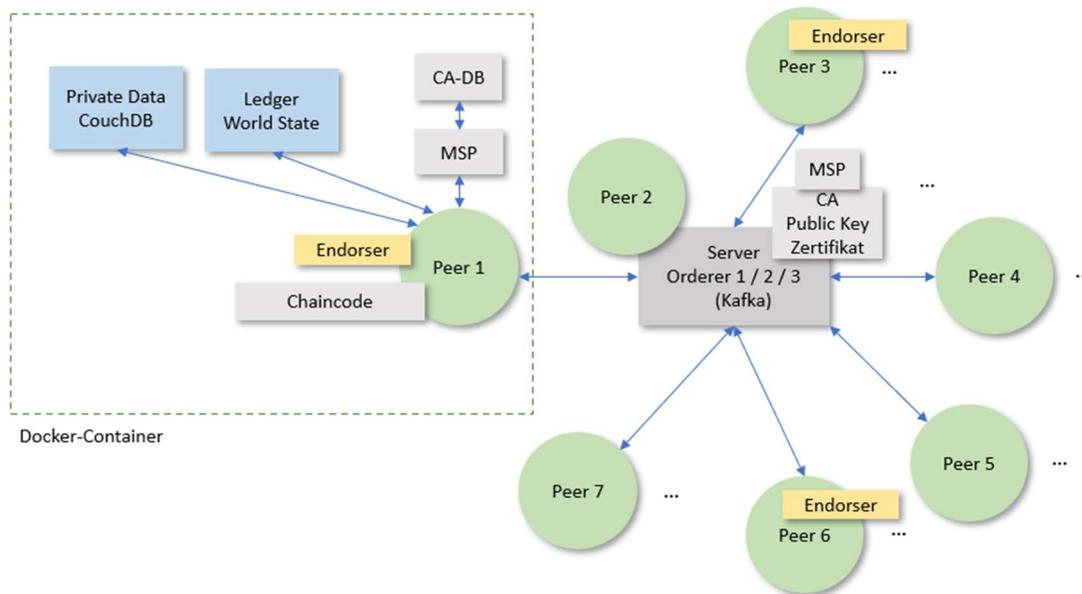


Abbildung 12: Hyperledger Fabric auf einem Verbund eigener Rechner/Server

Aufsetzen der Hyperledger Fabric Blockchain auf der IBM Cloud

Die Hyperledger Fabric Blockchain wird von IBM auch als Cloud-Variante angeboten. IBM bietet einen umfassenden und verwalteten Stack-Blockchain-as-a-Service (BaaS) an. Über diesen Service können Blockchain-Komponenten in einer selbst zu konfigurierenden Umgebung zusammengestellt und betrieben werden. Als technische Grundlage verwendet IBM die Blockchain-Technologie „Hyperledger Fabric“

IBM wirbt für den Blockchain Cloud-Einsatz mit verschiedenen Argumenten:³⁷

- Sehr kurze Vorlaufzeit für die Inbetriebnahme von eigenen Blockchains
- Niedrige Kosten, da nur Transaktions- oder Ressourcenkosten entstehen
- Vollständig realisiertes Dev/Test-/Run-Toolkit einschließlich DevOps
- Keine Investitionen in blockkettenspezifische System- oder Netzwerkexperten
- Updates und Sicherheitspatches werden überwacht und schnell angewendet
- Hohes Sicherheits- und Vertrauensniveau

Zu Testzwecken wurde bei IBM eine Public Blockchain-Infrastruktur in der Cloud angemietet und konfiguriert. Der garantierte Serverstandort war Frankfurt am Main. Der Support erfolgte über Dallas/USA. Die Netzwerkkomponenten konnten über ein User-Interface konfiguriert und bedient werden.

³⁷ <https://www.ibm.com/cloud/blockchain-platform> (Zugriff: 17.10.2020)

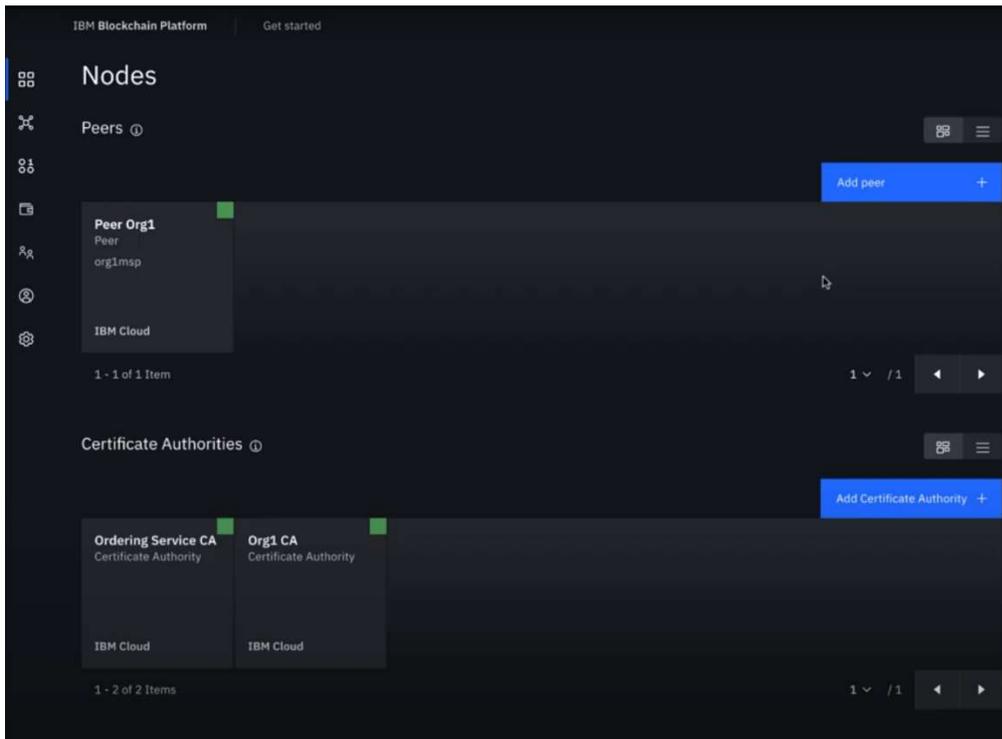


Abbildung 13: Konfiguration der IBM Public Cloud Hyperledger Fabric

Es wurden einige Peers angelegt und Daten in das Ledger geschrieben und gelesen.

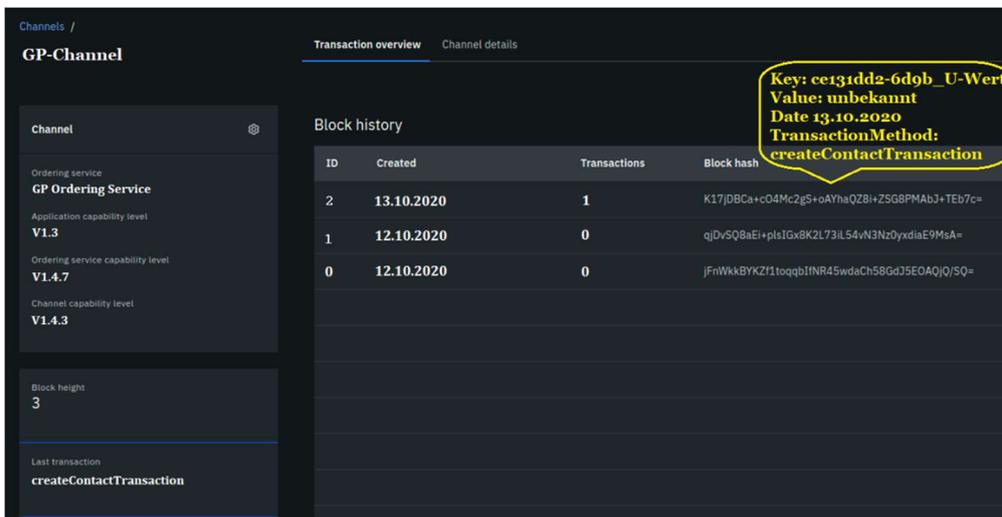


Abbildung 14: Auszug aus der IBM Public Cloud Block-History für Hyperledger Fabric

Beim Einrichten der Blockchain-Umgebung in der Cloud wurde deutlich, dass trotz allen zur Verfügung gestellten Vereinfachungen und Hilfestellungen des Anbieters spezielles Know-how zum Einrichten und Betreiben erforderlich ist. Bei den ersten Tests im Rahmen der Inbetriebnahme lokaler Peers in Verbindung mit dem Blockchain-Netzwerk in der IBM Cloud kam es zu verschiedenen Zertifikatsproblemen. Die Analyse der Probleme vollzog sich sehr aufwändig. Das Aufsetzen und die Inbetriebnahme einer Hyperledger Fabric Blockchain auf einem eigenen Linux Server gestaltete sich hingegen weniger aufwändig als ursprünglich angenommen. Die Analysemöglichkeiten (insb. auch das Debugging, Versionstests usw.) waren

hier wesentlich einfacher und flexibler zu handhaben. Neben den grundsätzlichen Funktionalitäten ist auch die Performance detailliert zu betrachten und hier bietet eine lokale Systemumgebung (Netzwerk, Peers, Server) wesentlich mehr Möglichkeiten zur Analyse als die Kommunikation mit Remotesystemen, bei welchen in der Regel weniger Zugriffsrechte zur Verfügung stehen. Eines der Ziele für das vorliegende Modell war die Kosten für die Umsetzung und den Einsatz möglichst gering zu halten, da in der Regel viele kleine und mittlere Unternehmen sowie andere Organisationen nur über begrenzte IT-Budgets verfügen. Die Kosten für die kleinste Stufe der IBM Public Cloud für Hyperledger Fabric bezifferte sich auf ca. 700 EUR/Monat. Das Implementieren der Blockchain auf einer IBM Private Cloud erforderte ein individuelles Angebot durch IBM. Die Kosten dürften erfahrungsgemäß noch höher ausfallen als bei der Public-Variante.

Die Umfrage von Bitkom Research und KMPG „Cloud-Monitor (2020)“³⁸ zeigt deutlich, dass der Cloud Einsatz zwar zugenommen hat, aber Unternehmen weiterhin gerade bei der Public-Cloud Variante Sicherheitsbedenken äußern. Die Mehrzahl der Unternehmen fürchten einen unberechtigten Zugriff auf sensible Unternehmensinformationen und bewerten die Rechtslage als weiterhin unklar.

Aus diesen Gründen und im Hinblick auf die Zielsetzung der vorliegenden Studie wurde beschlossen, im weiteren Verlauf der vorliegenden Studie nur die Variante Hyperledger Fabric auf einem Verbund eigener Rechner und Server zu verfolgen.

³⁸ <https://www.bitkom.org/Presse/Presseinformation/Drei-von-vier-Unternehmen-nutzen-Cloud-Computing#:~:text=Berlin%2C%202023.,Jahr%202017%20erst%2066%20Prozent.> (Zugriff: 04.12.2020)

5.6 Konzeption Prototypen-Schnittstellen-Logik

Die Speicherung sensibler Daten aus CRM-Systemen in ein Blockchain-Framework erfordert die Konzeption und Entwicklung einer Schnittstellen-Logik, die sich relativ herstellerunabhängig und aufwands- bzw. kosteneffizient zwischen den beiden Funktionseinheiten integrieren lässt. Ziel der Konzeption war es aus fachlicher und technischer Sicht, alle wesentlichen und relevanten Fakten (Eigenschaften) bezüglich der zu entwickelnden Schnittstellen-Logik zu erforschen und in einer strukturierten und vollständigen Form zu sammeln sowie den Soll-Zustand zu visualisieren. Für die späteren Testszenarien zur Speicherung der Daten zwischen den beiden Funktionseinheiten CRM-System und Blockchain werden im Nachfolgenden zwei technische Varianten als mögliche Schnittstellen-Logik aufgezeigt.

Variante 1 – Schnittstellen-Logik proprietär im CRM-Client

Die native Implementierung der Schnittstellen-Logik im CRM-Client setzt eine offene Systemarchitektur des CRM-Clients voraus, um programmtechnische Anpassungen vorzunehmen. In der Regel verfügen die meisten am Markt vorhandenen Systeme über diese Option.

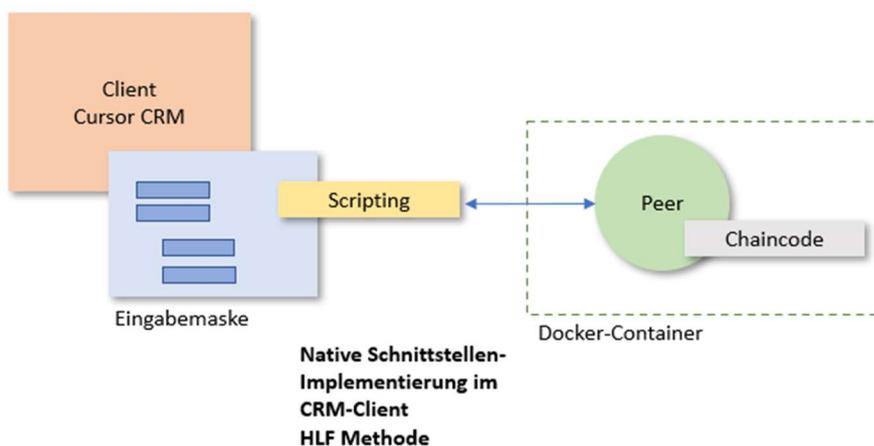


Abbildung 15: Schnittstellen-Logik über die native Einbindung im CRM-Client

In der obigen Abbildung sind zwei Funktionseinheiten dargestellt. Ein CRM-Client und ein Peer aus dem Blockchain-Netzwerk. Das skizzierte CRM-System bietet hierbei die Möglichkeit der Nutzung einer dynamischen Script-Sprache (z.B. Groovy) oder einer Workflow-Engine (BPM) an. Durch die Erstellung der Programmlogik lassen sich bereits aus dem CRM-System heraus, Funktionen auf der Blockchain ausführen. Zum Einsatz kommt hierbei das Hyperledger Fabric Framework, das zunächst im CRM als Library hinterlegt werden muss. Die Komponenten des Blockchain-Frameworks Hyperledger Fabric müssen ebenfalls installiert werden. Dazu zählen: Peers, Orderer, CA, Raft oder Kafka Nodes. Die Peers werden hierbei auf jedem Client installiert. Die Installation erfolgt in einem „Docker-Container“.

Vorteil

Die User-Interfaces (UI) in den CRM-Anwendungsclients bieten in der Regel sehr gute Funktionalitäten, um die Kommunikation über eine integrierte Schnittstellen-Logik abzubilden. Der Lösungsansatz auf Basis des User-Interface reduziert die Komplexität dahingehend, dass keine weiteren, zusätzlichen Software-komponenten auf den Clients installiert werden müssen, um den Schnittstellen-Prozess abzubilden.

Nachteil:

Zukünftig anstehende Releasewechsel seitens des CRM-Clients sind bei dieser Variante als nachteilig anzusehen. Releasewechsel des CRM-Clients können Anpassungen der programmtechnischen Schnittstellen-Logik auf jedem Client erfordern können, was den Wartungsaufwand deutlich erhöhen kann.

Variante 2 – Schnittstellen-Logik über Tomcat-Webserver

Bei diesem Konzeptionsansatz ist die Schnittstellen-Logik technisch nahe am Hyperledger Fabric Peer Clients implementiert.

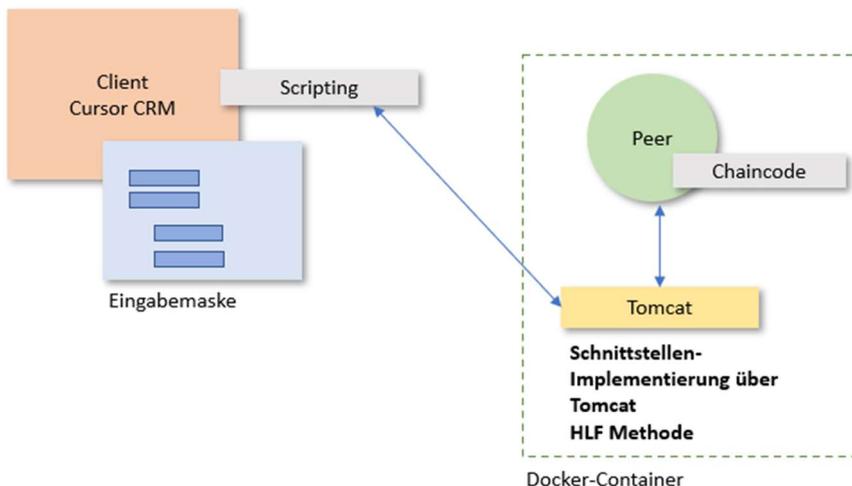


Abbildung 16: Schnittstellen-Logik über Tomcat-Webservice

Die 2. Variante der Schnittstellen-Logik basiert auf der Nutzung des Tomcat-Webserver³⁹. Er übernimmt dabei die Kommunikation zwischen dem CRM-System und der Blockchain. Apache Tomcat ist ein Open-Source-Webserver und Webcontainer, der die Spezifikation für Java Servlets und JavaServer Pages (JSP) implementiert und es damit erlaubt, in Java geschriebene Web-Anwendungen auf Servlet- bzw. JSP-Basis auszuführen. Zum Aufruf des Webservices kann im CRM-System ein Scripting-Aufruf oder eine alternative Möglichkeit eingesetzt werden. Der Webservices stellt im Grunde ein entkoppeltes Modul dar, das unabhängig von CRM-Systemen installiert werden kann. Der Aufruf des Webservice kann mittels „SOAP“ oder „REST“ erfolgen. Beide Aufruf-Varianten sind relativ schnell und einfach anwendbar. Im Falle von SOAP enthält der Aufruf den Methodennamen und die Parameter. Bei REST handelt es sich um eine URL mit Body und Quer-Parameter. Der Webservice kann lokal auf jedem Client oder je nach Konfiguration des Netzwerkes auch an einem zentralen Punkt installiert werden.

Vorteile

Sehr einfache und flexible Integration in verschiedene Systeme
Weitreichend unabhängig vom angebundenen CRM-System

Nachteil

Weiterer Dienst „Tomcat-Webserver“, der zu installieren und zu warten ist.

³⁹ <http://tomcat.apache.org/>

5.7 Entwicklung Prototypen

Im Nachfolgenden wird der Entwicklungsprozess der technischen Ansätze für die Schnittstellen-Logik beschrieben. Die technischen Ansätze wurden nur auf einem rudimentären Niveau entwickelt mit dem Ziel, die Funktionstüchtigkeit zu überprüfen.

5.7.1 Entwicklung, Test Prototypen Schnittstellen-Logik (3 Sprints)

1. Sprint

Im 1. Sprint wurde die technische Umsetzung der Schnittstellen-Variante 1 - „Schnittstellen-Logik proprietär im CRM-Client“ an den zwei ausgewählten CRM-Systemen „Cursor CRM und MS Dynamics CRM“ überprüft. Hyperledger Fabric bietet hier zur Kommunikation Anbindungen per Java, Go oder Nodes.js an. Um die Integration zu prüfen, wurde die Programmiersprache Java ausgewählt. Ausschlaggebend hierfür war das bereits im Team vorhandene Know-how der Entwickler.

Das Cursor CRM-System basiert auf Java und unterstützt dieses sowohl im Fat- als auch im Webclient. Eine Anbindung über das Fabric SDK konnte über die Java-Klassen einfach realisiert und erfolgreich getestet werden. Eine Integration der gleichen Lösungsmethodik in MS Dynamics konnte in der aufgesetzten Demoumgebung nicht erfolgreich realisiert werden. MS Dynamics setzt in der aktuellen Version als Webclient eher auf die Möglichkeiten von Webservices als auf die native Integration von Java-Klassen.

Aufgrund dieser Erkenntnis und der Anforderung, dass die Integration möglichst neutral vom CRM-System erfolgen sollte, wurde ein Konzept erarbeitet wie die Schnittstellenlogik in einen Webservice ausgelagert werden könnte. Zum Betrieb der Schnittstelle wurde der „Apache-Tomcat“ ausgewählt. Hierbei sind insbesondere bei Systemen, die in einer Cloud gehostet werden, verschiedene Installationsszenarien möglich:

Szenario 1 - MS Dynamics (Cloud)

Eine Kommunikation aus der Web-Applikation vom MS Dynamics CRM zu der lokalen Schnittstelle Tomcat konnte aufgrund von technischen Beschränkungen (Server zu Clientnetz) nicht erfolgen.

- ⇒ Lösungsvariante 1 wäre hier ein ausgelagerter Schnittstellen-Service, von welchem eine Kommunikation ins Clientnetz erfolgt und der vom MS Dynamics CRM verfügbar ist. Diese Variante ist jedoch nur eingeschränkt zu empfehlen, da hier eine Schwachstelle in der Sicherheitsinfrastruktur entstehen könnte, die durch weitere Maßnahmen abzusichern wäre.
- ⇒ Lösungsvariante 2 wäre die Implementierung des Schnittstellenaufrufs über ein clientseitiges JavaScript. Hier könnte der Benutzer den lokalen Schnittstellenservice direkt aufrufen. Bei diesem Vorgehen kann es allerdings zu Belastungsspitzen bei Prozessor und Arbeitsspeicher auf Clientseite kommen.

Szenario 2 - MS Dynamics (on-Premise)

- ⇒ Lösungsvariante 1 wäre der ausgelagerte Schnittstellenservice. Es kann angenommen werden, dass im Rahmen der Hosting-Infrastruktur von Unternehmen bereits ein System vorhanden ist, das den Service mit übernehmen könnte und die passende Freigabe für die Netzwerkkommunikation bereits vorliegt.
- ⇒ Bei Lösungsvariante 2 gibt es keine weiteren Einschränkungen zu dem äquivalent im Cloud-Bereich.

Der Aufruf könnte über die Möglichkeiten der Javascript Extensions für Forms erfolgen. Hier könnte beispielsweise das Formular „ onSave“ Ereignis oder Dataverse beispielsweise mit Plug-Ins oder Flows genutzt werden. Wichtig wäre, dass die Feldinformation direkt nach dem Speichern der Information in MS Dynamics CRM und deren Verarbeitung über den Schnittstellenservice wieder „gelöscht“ bzw. anonymisiert überschrieben wird.

Beispiel für clientseitiges Javascript mit Dataverse => <https://docs.microsoft.com/de-de/powerapps/developer/data-platform/webapi/web-api-samples-client-side-javascript>

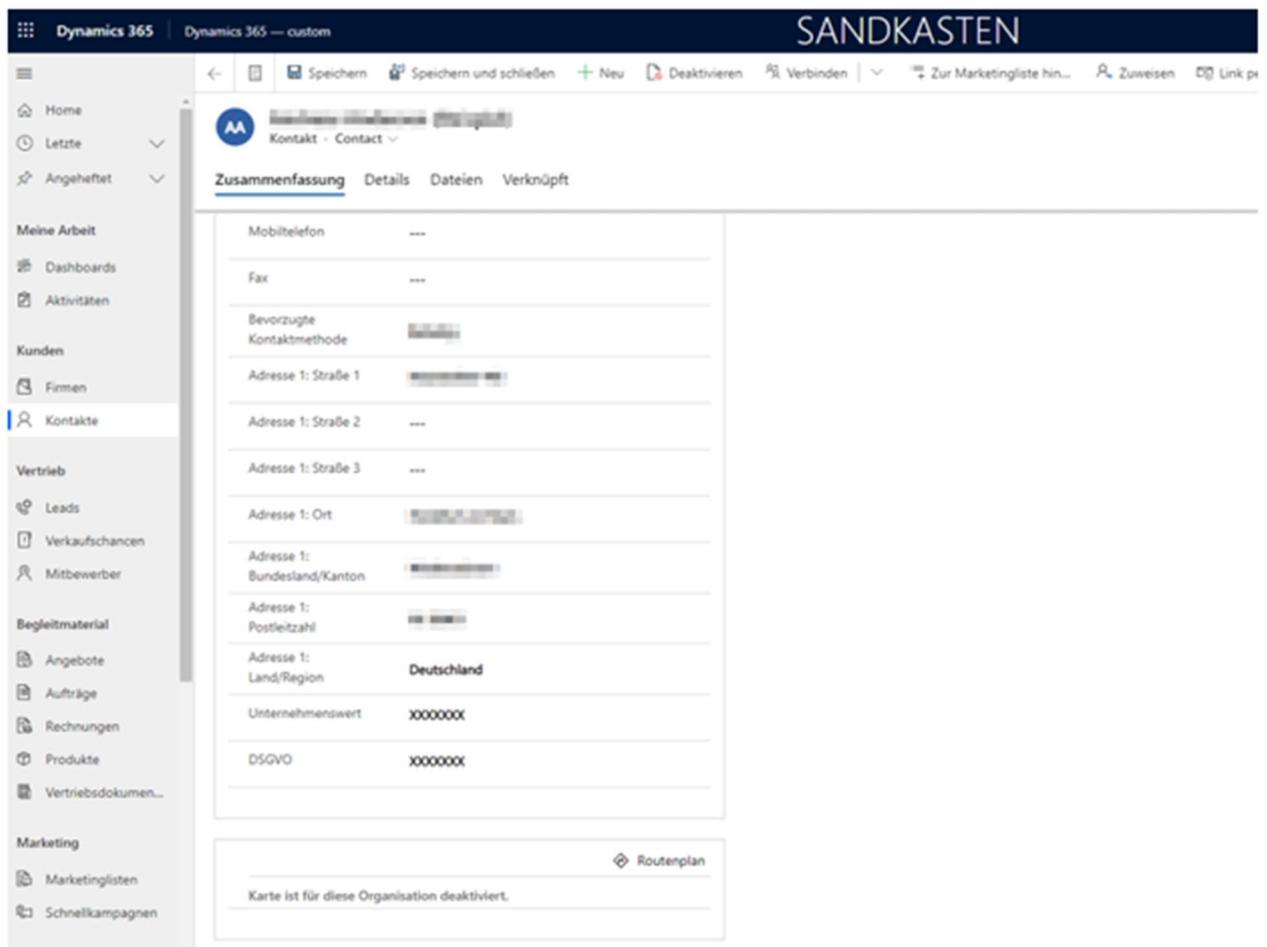


Abbildung 17: Beispiel für die konfigurierten Felder in Dynamics 365

Zusammenfassung:

Bei der technischen Überprüfung der Schnittstellen-Logik „Variante 1“ über das SDK in Fabric stellte sich heraus, dass sich die Anbindung im Cursor CRM-System über die native Java-Implementierung relativ einfach vollziehen lässt. Bei dem MS Dynamics CRM-System hingegen war die Implementierung komplexer und technisch schwieriger umsetzbar.

Aufgrund der zuvor beschriebenen technischen Schwierigkeiten bestand bei Variante 1 die Gefahr, dass die Lösung bei weiterer Anpassung der Schnittstellen-Logik zu proprietär ausfallen könnte. Da die Zielanforderung des vorliegenden Projektvorhabens in der Entwicklung einer einfachen und relativ herstellerunabhängigen Schnittstellen-Logik lag, wurde die Variante 1 zunächst zurückgestellt und im folgenden Sprint die Überprüfung der Variante 2 vorgenommen.

2. Sprint

Im 2. Sprint fand die technische Überprüfung der Schnittstellen-Logik Variante 2 – „Schnittstellen-Logik über „Tomcat“ statt.

Bei diesem Ansatz kommuniziert das CRM-System per SOAP mit einem Schnittstellenservice lokal auf dem Rechner der Benutzer. Der Schnittstellenservice implementiert die Blockchain mit Hilfe der von Hyperledger zur Verfügung gestellten Java-Klassen.

Zur weiteren Absicherung gegen schadhafte Angriffe und zur Vereinfachung des Deployment-Prozesses der Schnittstellen-Logik wurde der „Tomcat-Dienst“ nun in die Docker Container der Clients/Peers verlegt.

Schnittstellen-Logik Variante 2

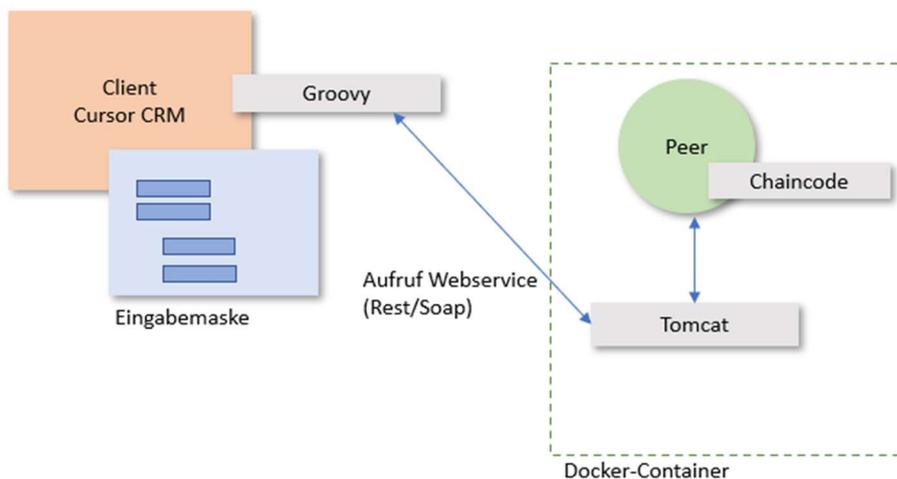


Abbildung 18: Verlegung des Tomcat-Dienstes in den Docker-Container

Da eine Integration von SOAP-Webservices für beide CRM-Systeme (CURSOR und MS Dynamics) sehr ähnlich durchzuführen wäre und nur ein geringer Erkenntnisgewinn hinsichtlich des Betriebs einer solchen mehrfach implementierten Lösung zu erwarten ist, wurde nur eine Implementierung durchgeführt.

Die Wahl fiel hier auf das CURSOR CRM, da auch hier die Variante 1 erfolgreich umgesetzt werden konnte und damit ein inhaltlicher Vergleich der beiden Varianten möglich wird.

3. Sprint

Im 3. Sprint wurden verschiedene technische Erweiterungen und Optimierungen an der bisher implementierten Lösung vorgenommen.

Integrierung der Private Data Collection

Zur Verarbeitung von Daten, die der Datenschutzgrundverordnung (DSGVO) unterliegen, wurde in diesem Sprint die „Private Data Collection“ von Fabric integriert. Ein speziell entwickelter Chaincode übernimmt dabei die Datenkommunikation der transienten Daten zwischen der CouchDB von Fabric und dem CRM-Anwender-Client. Aus dem CRM-System werden sensible Daten, die der DSGVO unterliegen, über die Schnittstelle automatisiert nicht im Ledger, sondern

in einer getrennten Datenbank (CouchDB) gespeichert. Ist dieser Prozess erfolgreich abgeschlossen, werden die sensiblen Informationsobjekte auf dem Server-File-System des CRMs gelöscht. Im Falle des Cursors CRM-Systems übernimmt dies ein BPM-Prozess.

Datenverschlüsselung

Um die Daten, die im Ledger oder in der CouchDB von Hyperledger Fabric im Standard unverschlüsselt gespeichert werden, vor unberechtigter Einsichtnahme zu schützen, wurde eine Datenverschlüsselung implementiert. Zwei mögliche technische Varianten zur Verschlüsselung wurden dafür konzipiert.

Variante 1

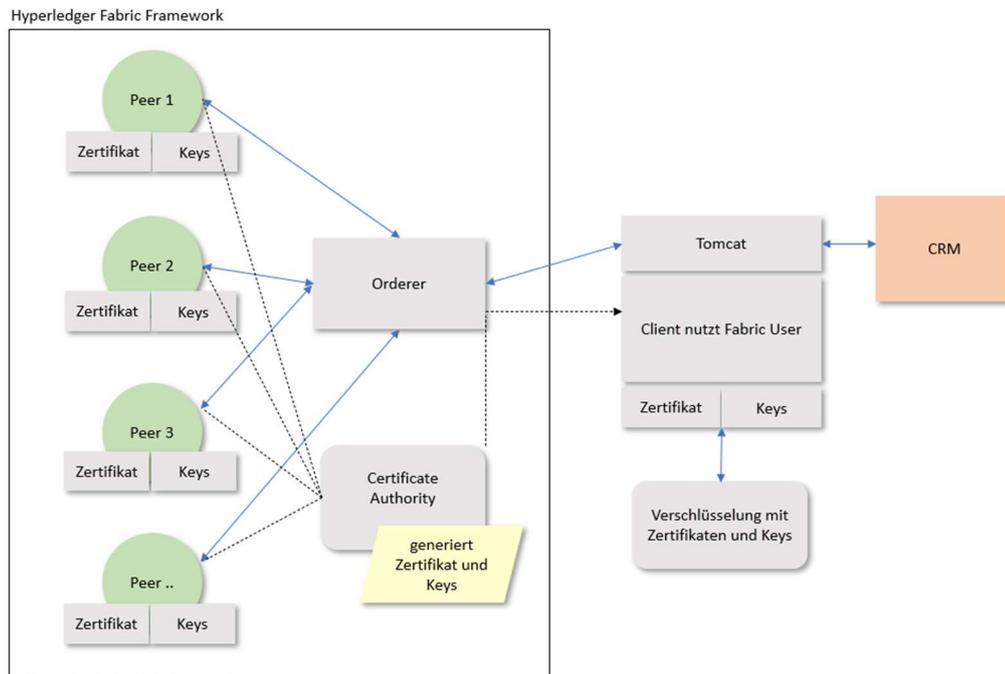


Abbildung 19: Verschlüsselung durch Schlüssel aus CA Fabric

Die 1. Variante umfasst die Verschlüsselung mit öffentlichen und privaten Schlüsseln durch die direkte Nutzung aus dem Hyperledger Framework heraus. In der obigen Abbildung wird dies exemplarisch aufgezeigt durch die Nutzung des Tomcat-Webservices, der hierbei als Fabric-Client fungiert. Über die Fabric Certificate Authority (CA) werden die benötigten Zertifikate sowie öffentliche und private Schlüssel generiert. Diese werden für das asymmetrische Kryptosystem benötigt. Der private Schlüssel eines Fabric-Users wird hierbei zur Verschlüsselung genutzt. Später kann jeder andere Nutzer mit dem Zertifikat, das den öffentlichen Schlüssel enthält, die Daten entschlüsseln. Auf Anfrage stellt die CA hierbei das Zertifikat zur Verfügung.

Vorteil:

Eine weitere Certificate Authority (CA) wird nicht benötigt.
Das Hyperledger-Framework kann direkt genutzt werden.
Es kann schnell eingesetzt werden.

Nachteil:

Um Organisationen miteinander verbinden zu können, muss die CA für diese zugänglich und auch abgesichert sein.

Variante 2

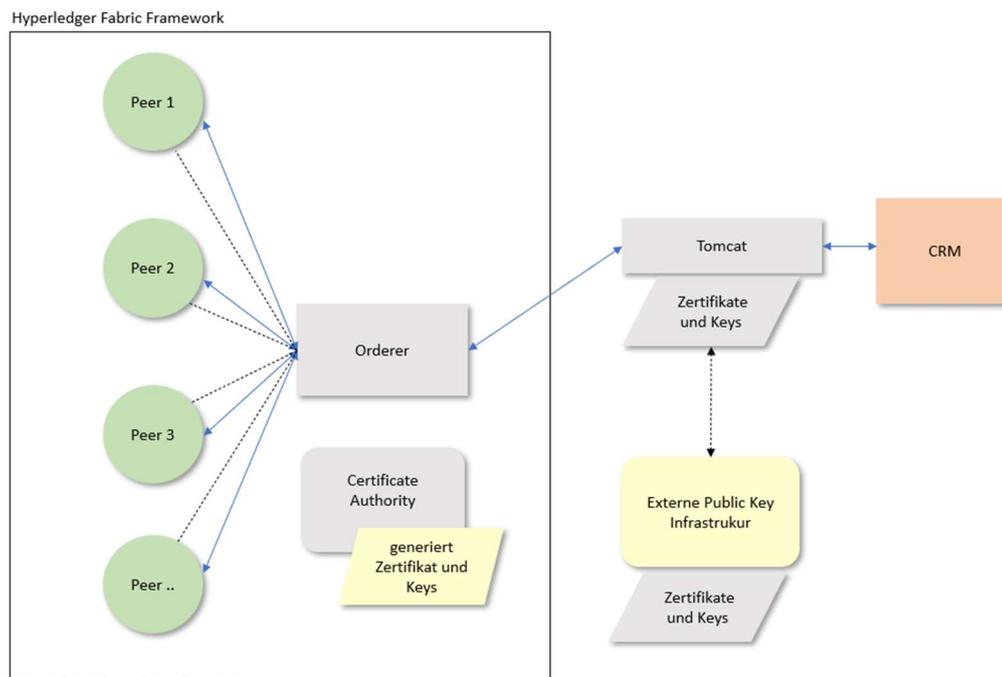


Abbildung 20: Verschlüsselung durch externe Public Key Infrastruktur

Die 2. Variante zur Daten-Verschlüsselung verwendet eine externe „Public Key Infrastruktur“. Das Verfahren bleibt hierbei gleich wie bei Variante 1. Die CA, die in der Public Key Infrastruktur vorhanden ist, verteilt das Zertifikat und den privaten Schlüssel.

Der private Schlüssel liegt auch hier im Tomcat zur Verschlüsselung bereit. Der öffentliche Schlüssel, der im Zertifikat enthalten ist, dient zur Entschlüsselung

Vorteil:

Wenn eine Organisation eine bestehende CA bereits im Einsatz hat, dann kann diese hier genutzt werden und es entstehen keine weiteren Insel-CA's. Dadurch ist ebenfalls eine Trennung zwischen der Authentifizierung im Hyperledger-Netz und der Verschlüsselung möglich.

Nachteil:

Das Generieren von Zertifikaten über die externe Certificate Authority ist kostspielig und aufwändig.

Für die vorliegende Studie kam die „Variante 2“ zur Verschlüsselung der Transaktionsdaten zum Einsatz. Dies ermöglichte einen schnellen und kostengünstigen Test-Einsatz. Dadurch konnten die Transaktionsdaten verschlüsselt im Ledger und im Private Data Bereich (CouchDB) von Fabric abgelegt werden und somit vor unberechtigter Einsichtnahme geschützt werden.

Speicherung von Dokumenten

Um neben den sensiblen Inhalten aus Feldern der Eingabemasken von CRM-Systemen zusätzlich auch sensible Dokumente (wie z.B. Word, Excel, PDF, etc.) auf der Fabric- Blockchain speichern zu können, wurde die Schnittstellen-Logik technisch erweitert. Diese Erweiterung begründete sich auf die zu Beginn des Projekts geführten Experten-Interviews.

Zur Speicherung von DSGVO konformen Dokumenten auf der Blockchain wurde zunächst der Bytestream gelesen und in einen Hex-String umgewandelt, um diesen im Anschluss auf der

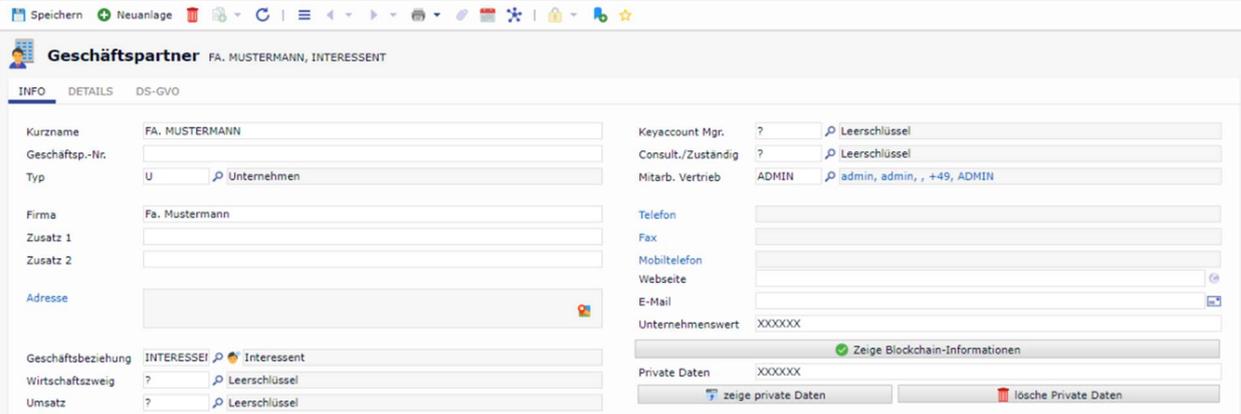
Blockchain speichern zu können. Die Reduzierung der Dateigröße war erforderlich, da im Austausch-Protokoll (gRPC) von Fabric eine Buffer-Begrenzung implementiert ist. Durch diese Konvertierung war es möglich, über diesen Ansatz im Testlauf ein Dokument von bis zu 1,8 MB speichern zu können. Im Falle des Lesens über die Schnittstelle erfolgte die Konvertierung in einen Byte-Stream und die Erstellung/Ablage des Dokuments im File-Systems des Users. Ist dieser Prozess erfolgreich abgeschlossen, werden die sensiblen Informationsobjekte auf dem Server-File-System des CRMs gelöscht. Im Falle des Cursors CRM-Systems übernimmt dies ein BPM-Prozess.

Einbau Logging zum Messen der Latenzen

Um die Zeiten (Latenzen) und den Durchsatz beim Lesen bzw. Schreiben von Transaktionen messen zu können, wurde eine spezielle Programmroutine entwickelt. Dieses „Mess-Programm“ wurde auf allen CRM-Testclients installiert. Das Logging der Messdaten erfolgte in einer lokalen Datei.

5.7.2 Testläufe Datenspeicherung

Nach Entwicklung der technischen Ansätze zur Schnittstellen-Logik über 3 Sprints erfolgten die ersten umfangreicheren Tests zur Datenspeicherung mit der final ausgewählten Schnittstellen-Logik über Tomcat-Webserver. Zu Beginn der Tests wurde der Start von Transaktionen (lesen/schreiben) aus der Cursor CRM-Anwendung direkt von den Test-Anwendern in der Eingabemaske ausgelöst.



The screenshot shows a web browser window displaying a CRM application. The browser's address bar shows the URL 'Speichern' and 'Neuanlage'. The page title is 'Geschäftspartner FA. MUSTERMANN, INTERESSENT'. The page has three tabs: 'INFO', 'DETAILS', and 'DS-GVO'. The 'INFO' tab is active, showing a form with the following fields:

Kurzname	FA. MUSTERMANN	Keyaccount Mgr.	? Leerschlüssel
Geschäftsp.-Nr.		Consult./Zuständig	? Leerschlüssel
Typ	U Unternehmen	Mitarb. Vertrieb	ADMIN admin, admin, +49, ADMIN
Firma	Fa. Mustermann	Telefon	
Zusatz 1		Fax	
Zusatz 2		Mobiltelefon	
Adresse		Webseite	
Geschäftsbeziehung	INTERESSENT Interessent	E-Mail	
Wirtschaftszweig	? Leerschlüssel	Unternehmenswert	XXXXXX
Umsatz	? Leerschlüssel	Private Daten	XXXXXX

At the bottom of the form, there are two buttons: 'zeige private Daten' and 'lösche Private Daten'. There is also a green checkmark icon and the text 'Zeige Blockchain-Informationen'.

Abbildung 21: Beispiel einer CRM-Eingabemaske

Bei dieser Vorgehensweise stellte sich schnell heraus, dass sich der organisatorische Aufwand und die enge Abstimmung unter den Test-Teilnehmern, z.B. beim zeitgleichen Auslösen von Transaktionen, schwierig gestaltete. Infolgedessen wurde für die weiteren Tests ein spezielles „Auslöse-Programm“ entwickelt, welches die jeweiligen Transaktionen automatisch und zeitgleich starten konnte. Über das zeitgleiche Auslösen von Transaktionen sollten die Auswirkungen von möglichen Parallelitätskonflikten beobachtet werden. Das Programm zum automatischen Auslösen von Transaktionen wurde auf allen Test-Clients neben der CRM-Anwendung installiert. Messungen ergaben, dass sich durch den Einsatz dieses Programms keine signifikanten Zeitunterschiede zur manuellen Auslösung über die Test-User einstellen.

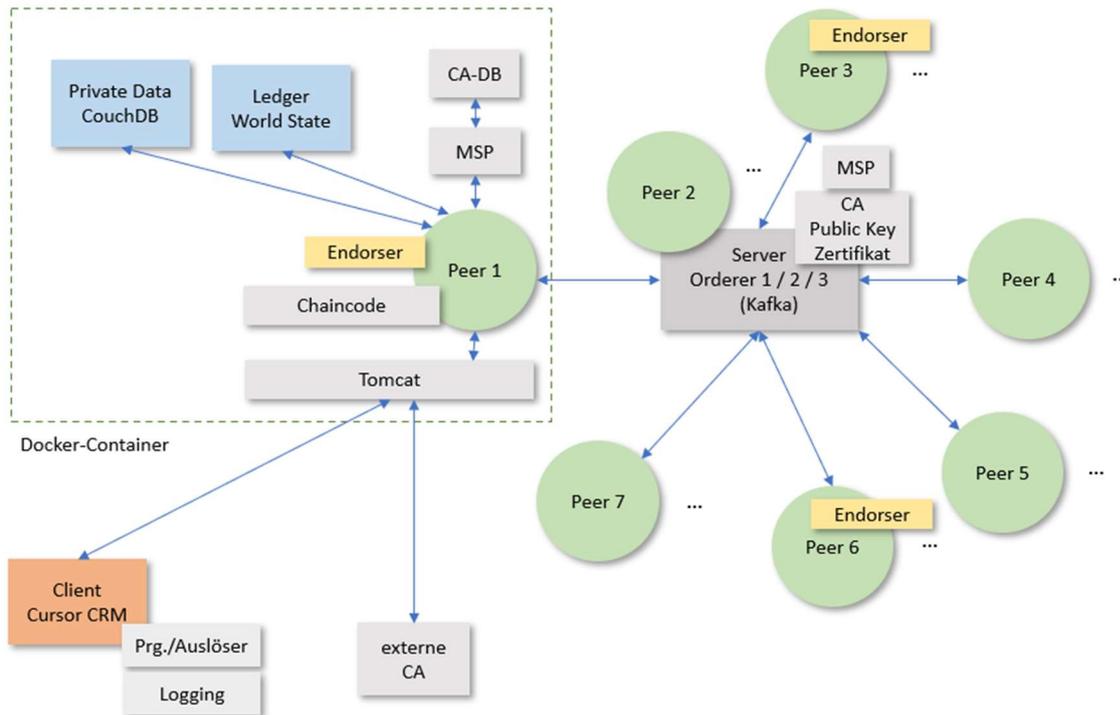


Abbildung 22: Teststellung für erste Testläufe

Die ersten Testläufe mit der zuvor dargestellten Testumgebung ergaben Latenzen für das Schreiben und Lesen von Feldwerten (ca. 1 KB) zwischen dem CRM-System (Eingabemaske) und der Hyperledger Fabric Blockchain von teilweise 20 bis 30 sec pro Transaktion. Latenzen für Transaktionen von Hyperledger Fabric die von anderen Untersuchungen veröffentlicht wurden liegen im Vergleich deutlich unter den hier gemessenen Werten.⁴⁰ Die Messergebnisse sind jedoch nicht direkt miteinander vergleichbar. Die gemessenen Latenzen für den vorliegenden Modellansatz umfassten eine erweiterte Mess-Schleife (End-to-End). Ausgehend vom Start der Transaktion aus dem CRM-System über den Tomcat-WebServer zur Blockchain und wieder zurück bis zum Startpunkt. Zusätzlich wurden die Transaktionsdaten vor Speicherung in der Blockchain beim vorliegenden Modell verschlüsselt und beim Lesen wieder entschlüsselt. Die Latenzwerte, die in anderen Studien zur Performance von Hyperledger Fabric veröffentlicht wurden, basieren in der Regel auf Messungen mit dem Hyperledger Fabric eigenen Tool „Caliper“. Die Mess-Schleife von Caliper ist auf die Messung der Latenzen nur innerhalb der Hyperledger Fabric Blockchain ausgerichtet.

Fehlerraten beim ersten Testlauf

Die Fehlerraten waren zu Beginn des ersten Testlaufs relativ hoch. Die Fehler begründeten sich z.B. auf „Timeouts“ wegen Nicht-Erreichbarkeit der Endorser auf den Client-Rechnern was zum Abbruch von Transaktionen führte. Bei Nicht-Erreichbarkeit der Endorser konnte die Endorsement-Policy nicht erfolgreich umgesetzt werden was zum Transaktionsabbruch führte. Darüber hinaus lösten unzureichende Portfreigaben im Netzwerk vermehrt Fehler aus. Neben vereinzelt Zertifikatsfehlern („Certificate in not avialable“) kam es aufgrund von Read-Write-Konflikten auch zu Fehlern im implementierten Chaincode. Hier wurden Verbindungen nicht geschlossen, was folglich nach einiger Testlaufzeit zu OutOfMemoryExceptions führte. Die Fehlerursachen wurden analysiert und konnten weitgehend behoben werden.

⁴⁰ Lincoln, N.K.. IBM Blockchain Developer Tools. Hyperledger Fabric 1.4.0 Performance Information Report. In: https://hyperledger.github.io/caliper-benchmarks/fabric/resources/pdf/Fabric_1.4.0_javascript_node.pdf (Zugriff: 04.12.2020)

Zusätzliche Implementierung des Hyperledger Fabric-Tools „Caliper“

Um die Latenzen innerhalb der vorliegenden Blockchain-Teststellung besser mit den Latenzen anderer Studien vergleichen zu können, wurde das Hyperledger Fabric Tool „Caliper“ zusätzlich in die Testumgebung integriert und im Anschluss einem Testlauf unterzogen.

Name	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
createLedger	1094	72	3.9	2.38	0.28	0.67	3.9
readLedger	1310	0	4.4	1.29	0.23	0.59	4.4
createDocument	1299	0	4.3	2.08	0.22	0.61	4.3
readDocument	3871	0	12.9	0.88	0.01	0.22	12.9

Abbildung 23: Ergebnisse Latenzen des Caliper-Messtools – Feldwert (1 KB)

Die obige Abbildung erhält einen Auszug der Messwerte (Latenzen) für den intern durchgeführten Testlauf mit dem Hyperledger Tool „Caliper“. Die Ergebnisse zeigen ähnliche Latenzwerte wie in anderen veröffentlichten Untersuchungen.⁴¹

Die Hauptgründe für die zuvor stark abweichenden Latenzen des ersten Testlaufs zu den Latenzen anderer, externer Studien lassen sich durch den Einsatz der erweiterten Mess-Schleife (End-to-End Messung) begründen. Zusätzlich spielt die Anordnung der Endorser-Peers, die Endorsement-Policy und der Tomcat-Dienst sowie die zusätzliche Ver- und Entschlüsselung der Transaktionsdaten eine Rolle. Der Tomcat-Dienst auf den Client-Rechnern benötigte mehr Zeit für die Verarbeitung als ursprünglich erwartet.

Um die Latenzen / Performance der Teststellung weiter zu verbessern, wurden verschiedene Möglichkeiten geprüft und umgesetzt.

Verlagerung des Tomcat-Dienstes und der Endorser

Um die Latenzen weiter zu verbessern, wurde beschlossen, zwei Endorser-Peers sowie den Tomcat-Dienst von den Peer-Clients auf den Server zu portieren.

⁴¹ Lincoln, N.K.. IBM Blockchain Developer Tools. Hyperledger Fabric 1.4.0 Performance Information Report. In: https://hyperledger.github.io/caliper-benchmarks/fabric/resources/pdf/Fabric_1.4.0_javascript_node.pdf (Zugriff: 04.12.2020)

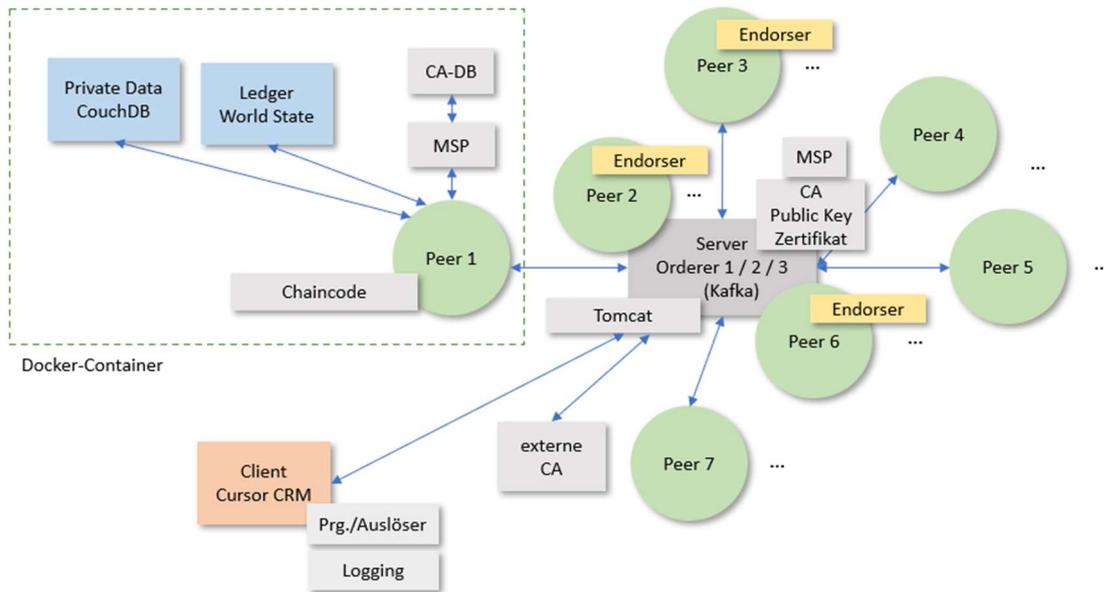


Abbildung 24: Verlagerung der Endorser-Peers und Tomcat-Dienstes auf den Server

Nach erfolgter Umstellung und erneutem Testlauf konnte festgestellt werden, dass sich die Latenzen signifikant verbessert hatten. Auch die Fehlerraten nahmen deutlich ab. Mit dieser Testumgebung wurde im Folgenden eine umfangreiche Evaluierung des Gesamtsystems durchgeführt.

5.7.3 Evaluation Prototypen Leistung und Sicherheit

Dieser Abschnitt beschreibt die Evaluierung der entwickelten technischen Ansätze als Gesamtsystem in Hinblick auf die Kriterien Leistung und Sicherheit.

Über diverse Metriken wurde die Performance des Gesamtmodells in unterschiedlichen Konstellationen gemessen. Im Anschluss erfolgte eine Sicherheitsbeurteilung der Systemarchitektur des Gesamtmodells.

Leistungs-Bewertung

Viele Leistungsmerkmale von Blockchain-Modellen, wie Hyperledger Fabric, sind aufgrund der Leistungskomplexität immer noch nicht gut verstanden. Dies bestätigte sich auch durch die geführten Experten-Interviews. In vielen anderen Untersuchungen wurde die Leistungsmessung vorrangig auf den Durchsatz von „Tausenden“ Transaktionen pro Sekunde ausgerichtet. Dabei wurde untersucht wie sich unterschiedlichste Konstellationen auf das Latenzproblem zugelassener Blockchain-Modelle auswirken. Die Latenzleistung wurde bei diesen Studien mithilfe verschiedener Systemkonfigurationen gemessen. Aufgrund der zunehmenden Anzahl von Konfigurationsoptionen und der nicht bekannten Einschränkungen sind diese Ergebnisse jedoch nicht direkt vergleichbar. Die zugrundeliegenden Netzwerke für die Bereitstellung von Blockchains mit Berechtigungen haben einen großen Einfluss auf die Analyseergebnisse.⁴² In den vielen empirische Studien wurde die Leistung auf unterschiedlichen Hardwareplattformen analysiert. Diese experimentellen Ergebnisse sind infolge dessen schwer miteinander vergleichbar, da sie stark von den zugrundeliegenden Hardware- und Netzwerk-Komponenten beeinflusst werden.⁴³

Für das vorliegende Projektvorhaben, der Speicherung und Änderungsverfolgung sensibler Informationsobjekte aus CRM-Systemen über das Hyperledger Fabric Blockchain-Netzwerk, war die Ausrichtung auf maximale Leistungsspitzen kein primäres Ziel. Mit dem vorliegenden Ansatz soll lediglich der Anteil sensibler Informationen aus CRM-Systemen auf der Blockchain gespeichert werden unter der Annahme, dass das Datenaufkommen von wirklich „sensiblen Informationen“ nur ein Bruchteil an den Gesamtdaten darstellt. Die Gesamtlösung sollte in Bezug auf die Ausführungs- bzw. Antwortzeiten bei der Speicherung bzw. beim Lesen von Informationen in einem für den Anwender zumutbaren Zeitintervall liegen. Die Evaluation sollte dabei zeigen, ob und inwieweit der entwickelte technische Ansatz geeignet erscheint die angestrebte Zielsetzung zu erfüllen. Für die Evaluation wurden die Daten methodisch erhoben, dokumentiert und ausgewertet, um das Vorgehen und die Ergebnisse nachvollziehbar und überprüfbar zu machen.

⁴² Xiaojiong, X.. et al. (2021). Latency performance modeling and analysis for hyperledger fabric blockchain network (ScienceDirect). In: <https://www.sciencedirect.com/science/article/pii/S0306457320309298> (Zugriff: 15.01.2021)

⁴³ Xiaojiong, X.. et al. (2021). Modellierung und Analyse der Latenzleistung für das Hyperledger Fabric Blockchain-Netzwerk. In: https://www.researchgate.net/publication/347833428_Latency_performance_modeling_and_analysis_for_hyperledger_fabric_blockchain_network (Zugriff: 08.01.2021)

Versuchsaufbau

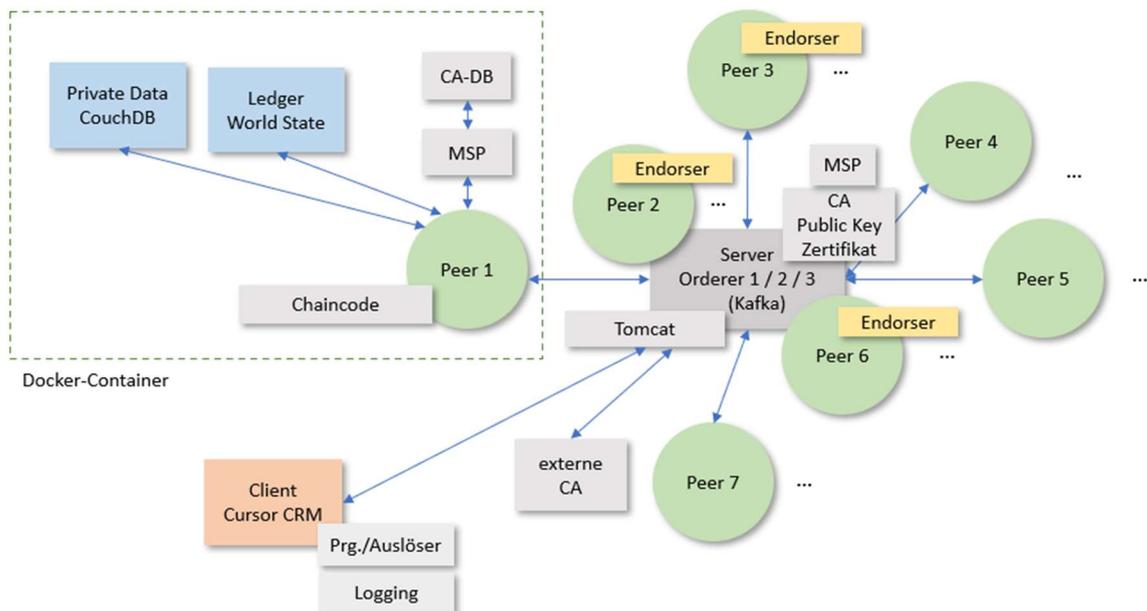


Abbildung 25: Teststellung für die Evaluierung

Die Teststellung umfasste 5 Cursor CRM-Clients, die Schnittstellen-Logik über den Tomcat-Dienst und das private zulassungsbeschränkte Hyperledger Fabric Netzwerk mit einem Channel, 7 Peers, 3 Endorser und 3 Orderer (Kafka). Das Mess-Programm wurde auf den 5 CRM-Clients implementiert. Die Cursor CRM-Software und die Peers von Fabric wurden auf einem Verbund eigener Rechner installiert. Zwei Endorser-Peers sowie die Orderer und der Tomcat-Webservice wurden auf einem Server aufgesetzt.

Beschreibung der Testumgebung Hard-/Software

Blockchain

Hyperledger Fabric Version 2.3.0

Client-Rechner/Peer

Das Netzwerk wurde auf unterschiedlich ausgestatteten Rechnern installiert.

Windows-System, 8-16 Gigabyte RAM und 2,5-3,5 GHz CPU

Auf jedem Client wurde ein Cursor CRM-Client installiert.

Die Peers laufen in einem Docker-Container.

Netzwerk-Typ

Zentral verwaltetes Gigabit Netzwerk auf Basis einer Baum-Topologie und einer strukturierten Verkabelung.

Unterschiedliche Subnetze mit jeweils separaten VLANs für Management, Netzwerk, Server, Clients, DMZ.

Vergabe dynamischer IP-Adressen für Clients.

Feste IP-Adressen für Server, Management, Netzwerk, DMZ.

Firewall

Fortigate 100E im HA Cluster (redundant)

Virens Scanner

F-Secure Business Suite für Server & Clients (zentral verwaltet)

Linux-Server

Zwei Peers/Endorser befanden sich auf einem Linux-Server, der als „Virtuelle Maschine“ aufgesetzt war. Einzelne Peers waren während des Tests im Homeoffice und über einen VPN-Zugang mit dem Unternehmensnetz verbunden. Auf dem Linux-Server wurden 3 Orderer, 2 Endorser und der Tomcat-Dienst installiert.

Endorsement-Richtlinie

Die Endorsement-Richtlinie wurde mit dem mit dem Typ "AND" verwendet.
Zwei von drei Endorser mussten zustimmen.

Orderer

3 Orderer auf einem Linux-Server (Konsensprotokoll: Kafka)

Blockgröße

Die Blockgröße des Ledgers betrug 2 MB.

Messstelle / Messverfahren

Die Latenz wurde pro Transaktion berechnet, ausgehend vom Zeitpunkt der Auslösung vom CRM-System über den Tomcat-Webservice zum Blockchain-Netzwerk und wieder zurück zum Startpunkt. Für die Messung der Latenz wurde eine speziell entwickelte Programmroutine eingesetzt, die die Messdaten protokollierte und in ein Log-File sicherte.

Beschreibung der Testdaten

Die Testdaten umfassten Feldwerte aus einer Cursor CRM-Eingabemaske, Größe 1 KB sowie Dokument-Objekte Größe 400 KB und 1.77 MB. Die Testdaten wurden in das „Ledger“ sowie in den „Private Data“ Bereich der Hyperledger Fabric Blockchain geschrieben und gelesen.

Evaluation Leistung

Über den zuvor dargestellten Versuchsaufbau wurden mehrere Testläufe durchgeführt und die Leistung der Teststellung evaluiert.

Feldwert schreiben ins Ledger (Größe 1 KB)

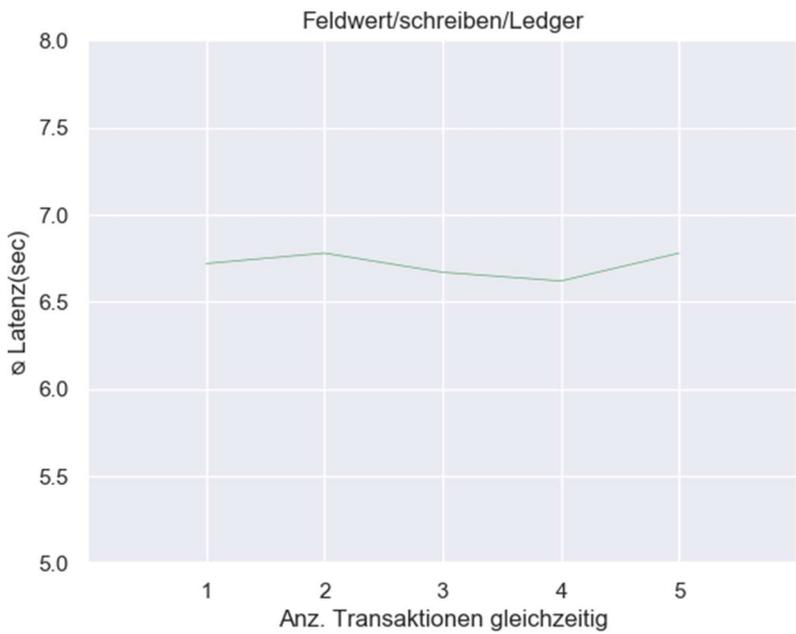


Abbildung 26: Liniengraphik - Feldwert schreiben ins Ledger (1KB)

Ø Latenz(sec)	6,72	6,78	6,67	6,62	6,78
Anz. Transaktionen gleichzeitig	1	2	3	4	5
Alle Transaktionen					
Ø Latenz(sec)	6,70				
Min Latenz(sec)	6,14				
Max Latenz(sec)	7,36				

Abbildung 27: Übersicht Latenzen - Feldwert schreiben ins Ledger (1KB)

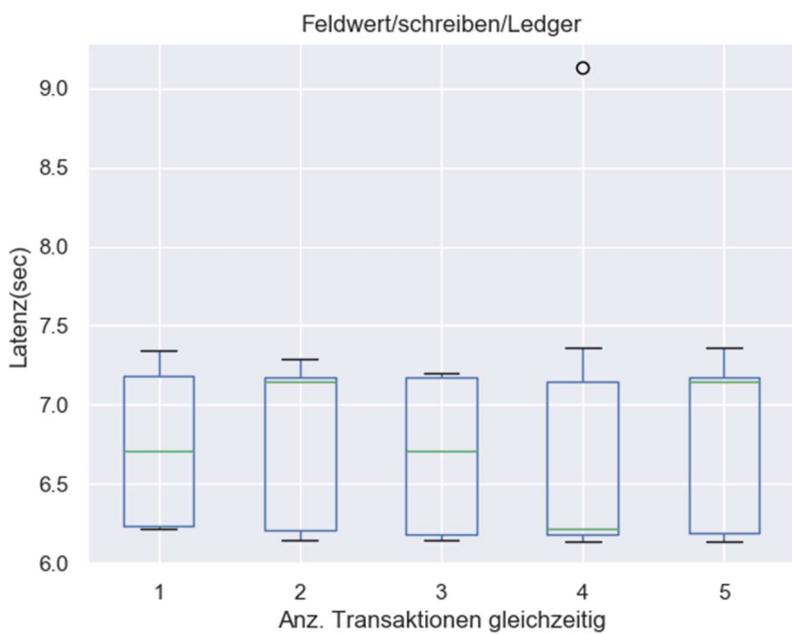


Abbildung 28: Boxplot – Feldwert schreiben ins Ledger (1KB)

Feldwert lesen aus Ledger (Größe 1 KB)

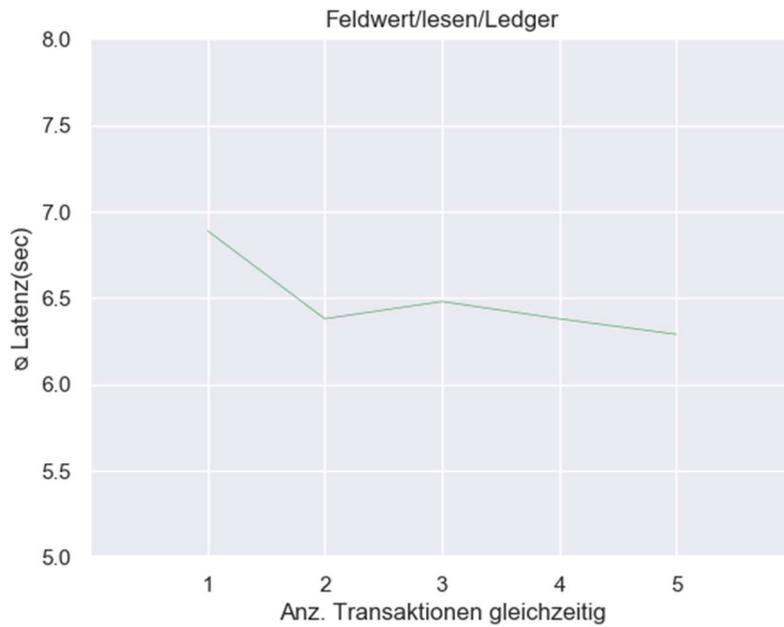


Abbildung 29: Liniengraphik - Feldwert lesen aus Ledger (1KB)

Ø Latenz(sec)	6,89	6,38	6,48	6,38	6,29
Anz. Transaktionen gleichzeitig	1	2	3	4	5
Alle Transaktionen					
Ø Latenz(sec)	6,48				
Min Latenz(sec)	5,25				
Max Latenz(sec)	7,36				

Abbildung 30: Übersicht Latenzen - Feldwert lesen aus Ledger (1KB)



Abbildung 31: Boxplot – Feldwert lesen aus Ledger (1KB)

Feldwert schreiben in Private Data Bereich (Größe 1 KB)

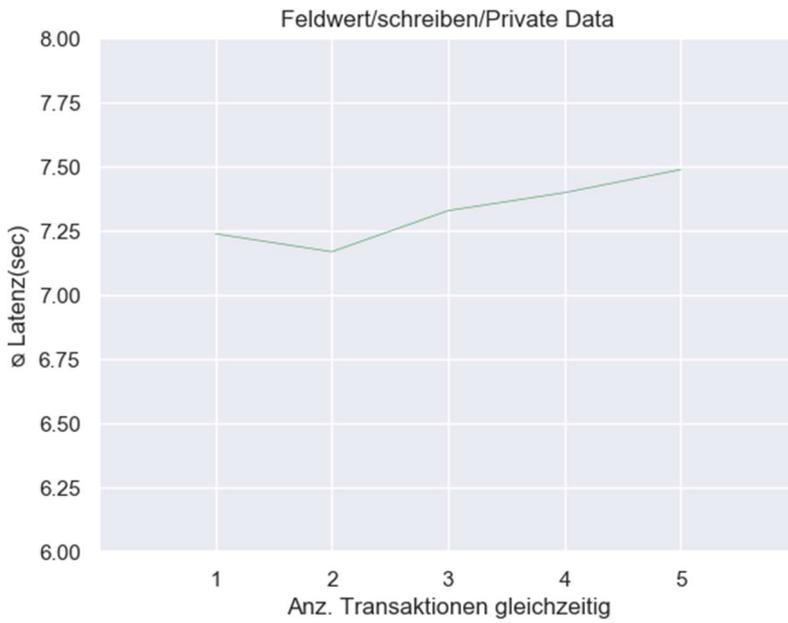


Abbildung 32: Liniengraphik - Feldwert schreiben in Private Data Bereich (1KB)

Ø Latenz(sec)	7,24	7,17	7,33	7,40	7,49
Anz. Transaktionen gleichzeitig	1	2	3	4	5
Alle Transaktionen					
Ø Latenz(sec)	7,33				
Min Latenz(sec)	6,17				
Max Latenz(sec)	9,43				

Abbildung 33: Übersicht Latenzen - Feldwert schreiben in Private Data (1KB)

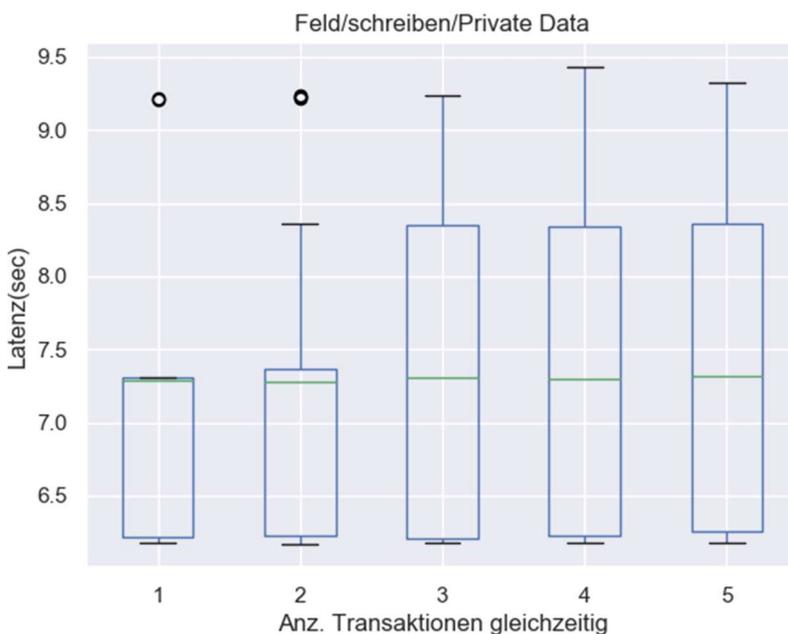


Abbildung 34: Boxplot – Feldwert schreiben in Private Data Bereich (1KB)

Feldwert lesen aus Private Data Bereich (Größe 1 KB)

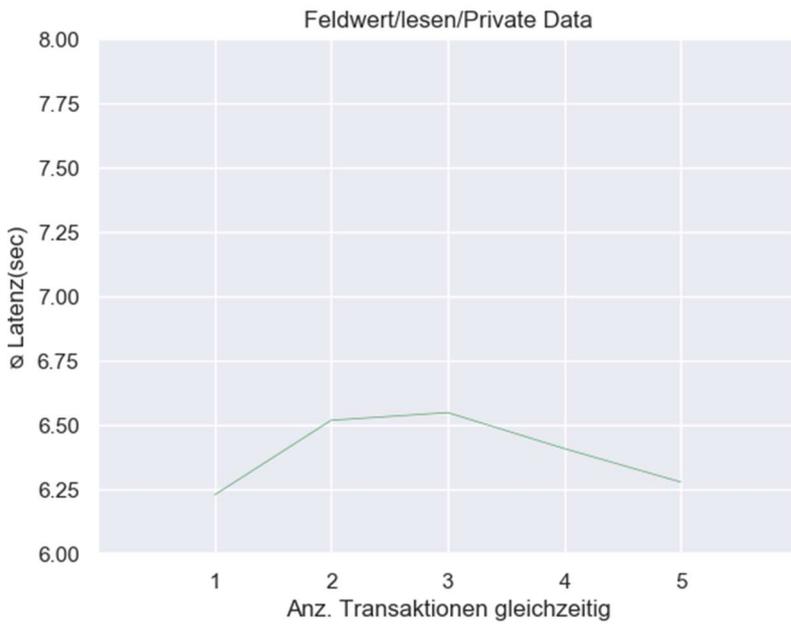


Abbildung 35: Liniengraphik - Feldwert lesen aus Private Data Bereich (1KB)

Ø Latenz(sec)	6,23	6,52	6,55	6,41	6,28
Anz. Transaktionen gleichzeitig	1	2	3	4	5
Alle Transaktionen					
Ø Latenz(sec)	6,39				
Min Latenz(sec)	5,15				
Max Latenz(sec)	7,62				

Abbildung 36: Übersicht Latenzen - Feldwert lesen aus Private Data (1KB)

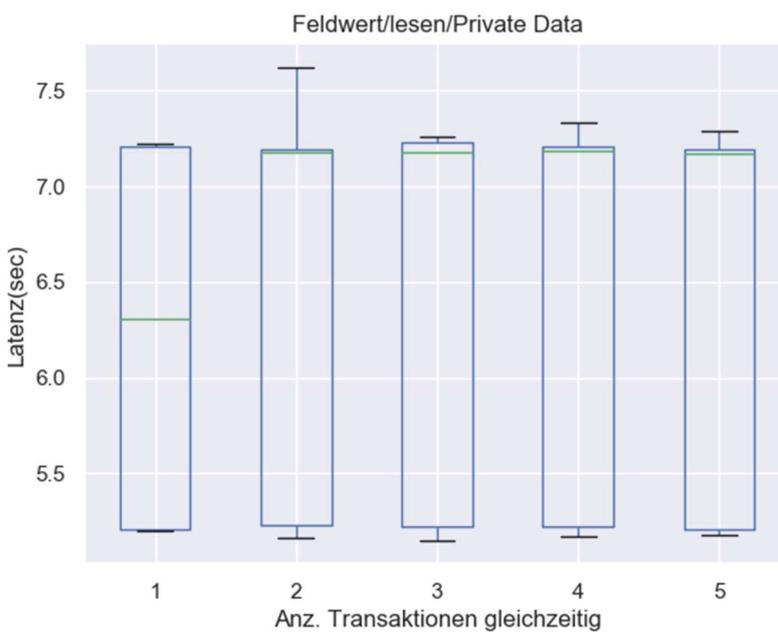


Abbildung 37: Boxplot – Feldwert lesen aus Private Data Bereich (1KB)

Dokument schreiben ins Ledger (Größe 400 KB)

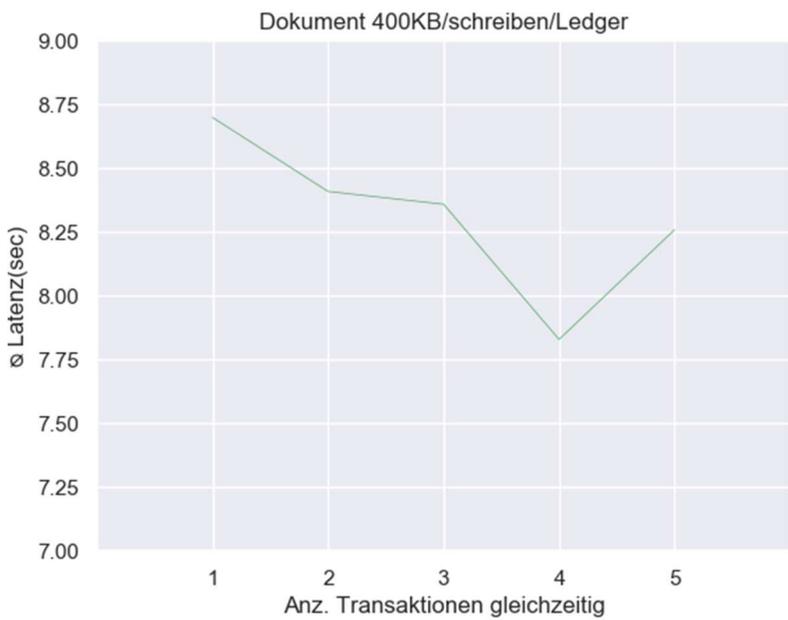


Abbildung 38: Liniengraphik - Dokument schreiben ins Ledger (400KB)

Ø Latenz(sec)	8,70	8,41	8,36	7,83	8,26
Anz. Transaktionen gleichzeitig	1	2	3	4	5
Alle Transaktionen					
Ø Latenz(sec)	8,31				
Min Latenz(sec)	6,57				
Max Latenz(sec)	12,76				

Abbildung 39: Übersicht Latenzen – Dokument schreiben in Private Data (400KB)

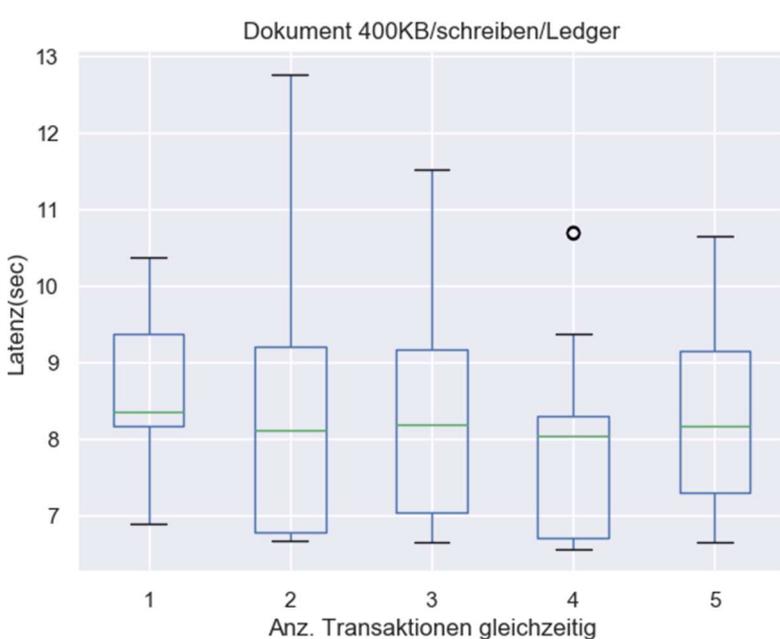


Abbildung 40: Boxplot – Dokument schreiben ins Ledger (400KB)

Dokument lesen aus Ledger (Größe 400 KB)

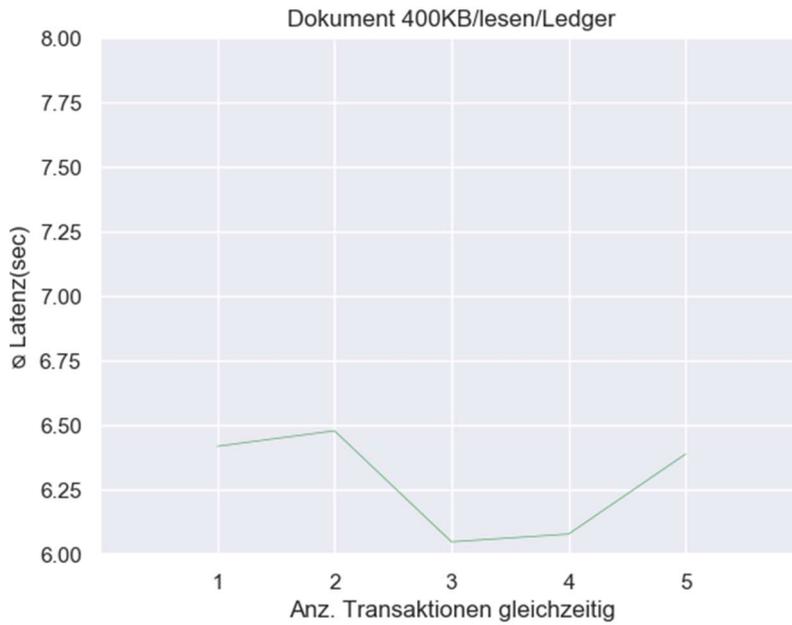


Abbildung 41: Liniengraphik - Dokument lesen aus Ledger (400KB)

Ø Latenz(sec)	6,42	6,48	6,05	6,08	6,39
Anz. Transaktionen gleichzeitig	1	2	3	4	5
Alle Transaktionen					
Ø Latenz(sec)	6,28				
Min Latenz(sec)	5,14				
Max Latenz(sec)	7,25				

Abbildung 42: Übersicht Latenzen – Dokument lesen aus Ledger (400KB)

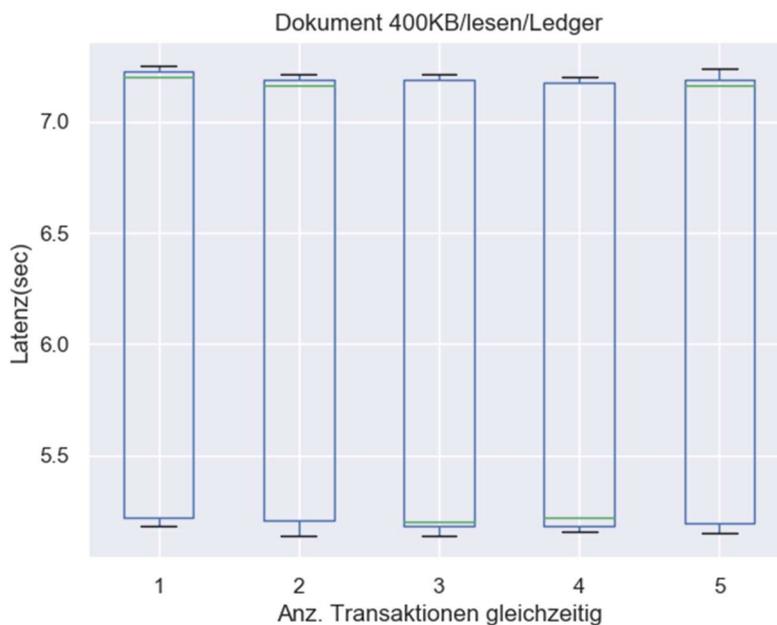


Abbildung 43: Boxplot – Dokument lesen aus Ledger (400KB)

Dokument schreiben in Private Data (Größe 400KB)

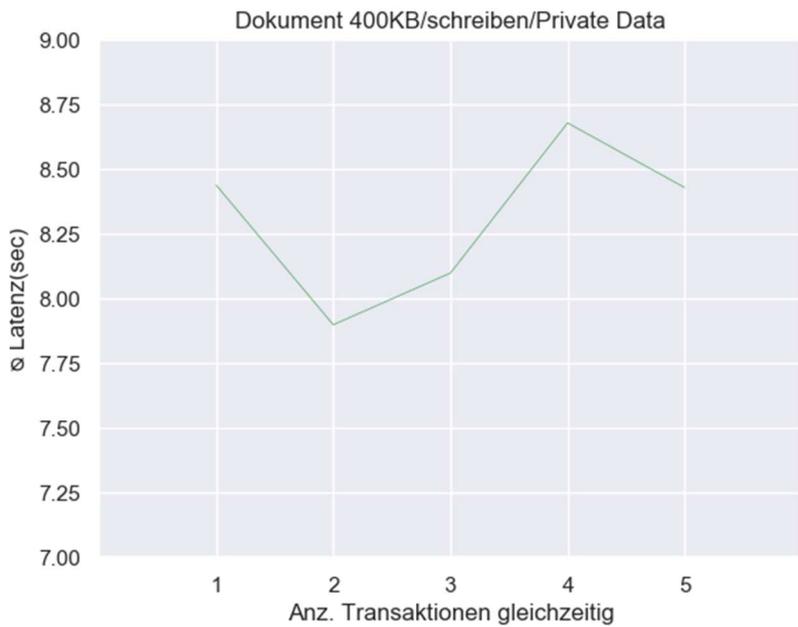


Abbildung 44: Liniengraphik - Dokument schreiben in Private Data (400KB)

Ø Latenz(sec)	8,44	7,90	8,10	8,68	8,43
Anz. Transaktionen gleichzeitig	1	2	3	4	5
Alle Transaktionen					
Ø Latenz(sec)	8,31				
Min Latenz(sec)	6,67				
Max Latenz(sec)	11,01				

Abbildung 45: Übersicht Latenzen – Dokument schreiben in Private Data (400KB)

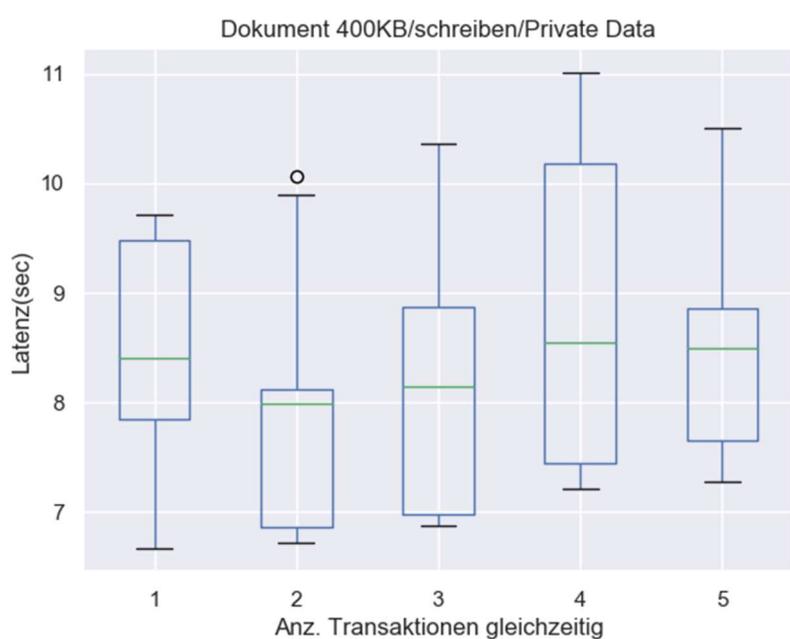


Abbildung 46: Boxplot – Dokument schreiben in Private Data (400KB)

Dokument lesen aus Private Data (Größe 400KB)

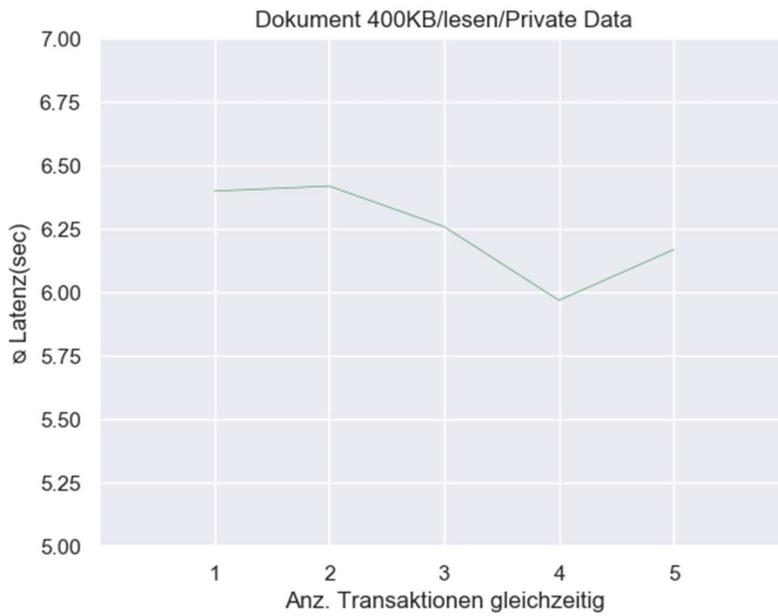


Abbildung 47: Liniengraphik - Dokument lesen aus Private Data (400KB)

Ø Latenz(sec)	6,40	6,42	6,26	5,97	6,17
Anz. Transaktionen gleichzeitig	1	2	3	4	5
Alle Transaktionen					
Ø Latenz(sec)	6,24				
Min Latenz(sec)	5,15				
Max Latenz(sec)	7,4				

Abbildung 48: Übersicht Latenzen – Dokument lesen aus Private Data (400KB)

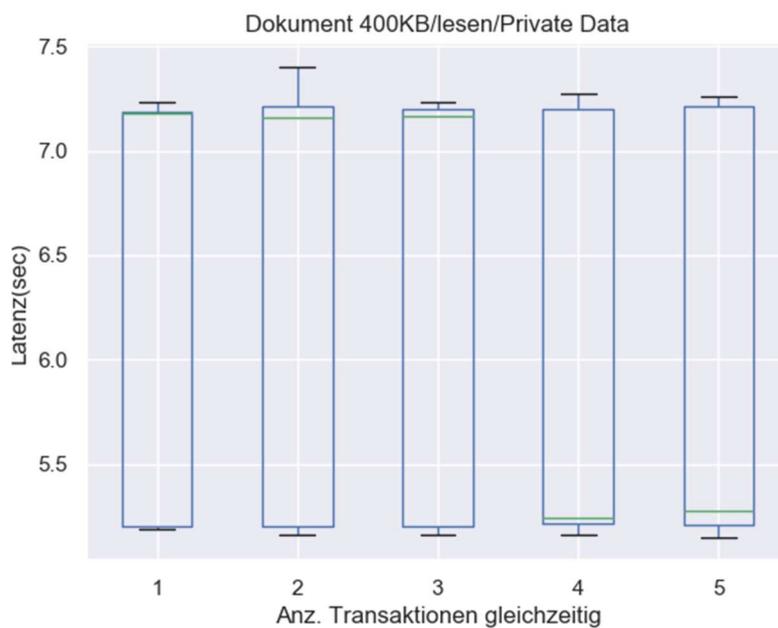


Abbildung 49: Boxplot – Dokument lesen aus Private Data (400KB)

Dokument schreiben ins Ledger (Größe 1.77MB)

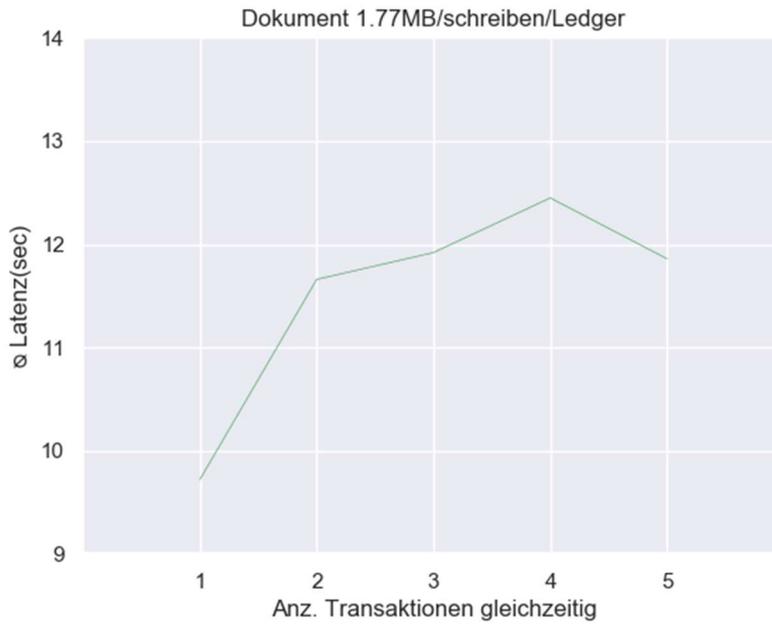


Abbildung 50: Liniengraphik - Dokument schreiben ins Ledger (1.77MB)

Ø Latenz(sec)	9,97	11,66	11,92	12,45	11,86
Anz. Transaktionen gleichzeitig	1	2	3	4	5
Alle Transaktionen					
Ø Latenz(sec)	11,57				
Min Latenz(sec)	7,53				
Max Latenz(sec)	21,94				

Abbildung 51: Übersicht Latenzen – Dokument schreiben ins Ledger (1.77MB)

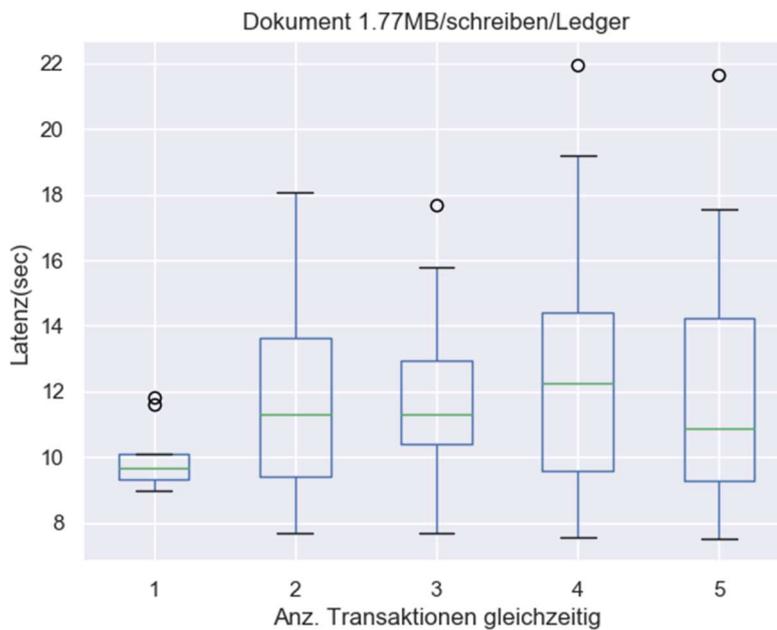


Abbildung 52: Boxplot – Dokument schreiben ins Ledger (1.77MB)

Dokument lesen aus Ledger (Größe 1.77MB)

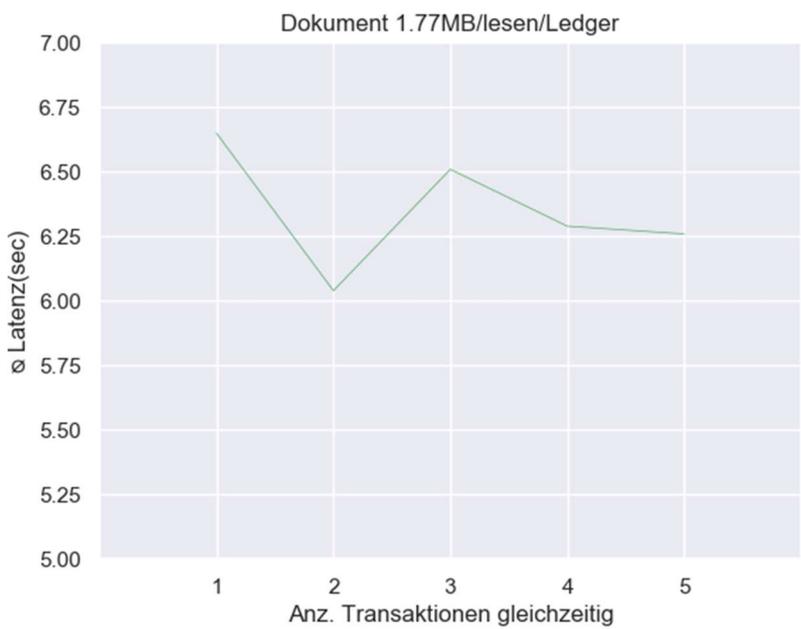


Abbildung 53: Liniengraphik - Dokument lesen aus Ledger (1.77MB)

Ø Latenz(sec)	6,65	6,04	6,51	6,29	6,26
Anz. Transaktionen gleichzeitig	1	2	3	4	5
Alle Transaktionen					
Ø Latenz(sec)	6,35				
Min Latenz(sec)	5,17				
Max Latenz(sec)	7,49				

Abbildung 54: Übersicht Latenzen – Dokument lesen aus Ledger (1.77MB)

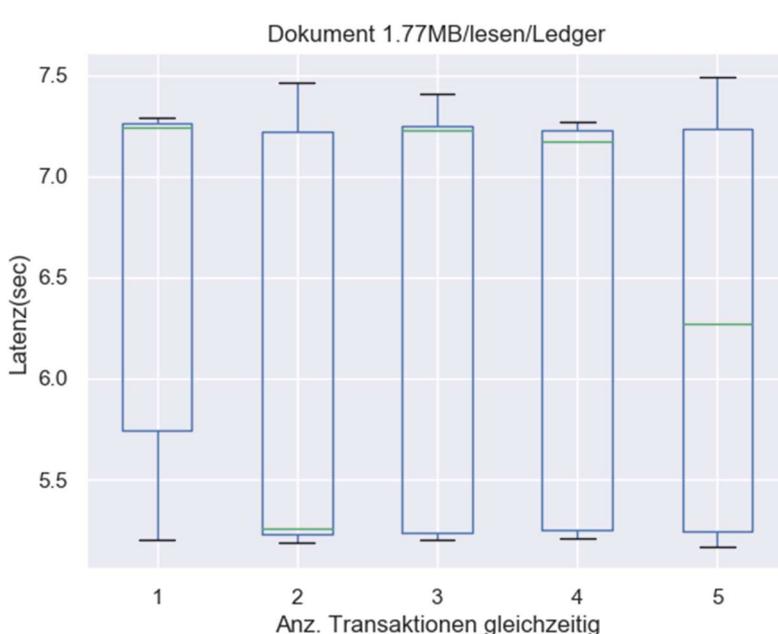


Abbildung 55: Boxplot - Dokument lesen aus Ledger (1.77MB)

Dokument schreiben in Private Data (Größe 1.77MB)



Abbildung 56: Liniengraphik - Dokument schreiben in Private Data (1.77MB)

Ø Latenz(sec)	8,52	11,19	13,41	12,29	11,84
Anz. Transaktionen gleichzeitig	1	2	3	4	5
Alle Transaktionen					
Ø Latenz(sec)	11,45				
Min Latenz(sec)	7,5				
Max Latenz(sec)	20,2				

Abbildung 57: Übersicht Latenzen – Dokument schreiben in Private Data (1.77MB)

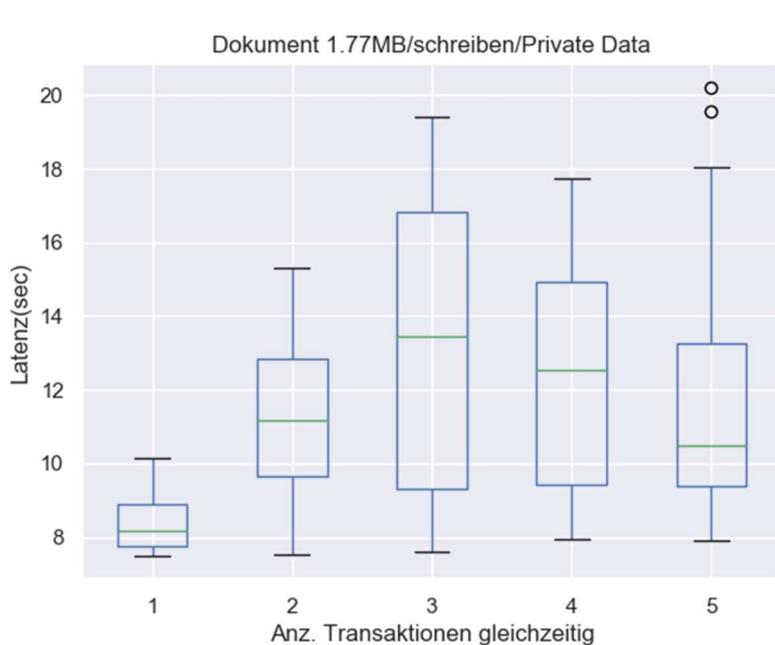


Abbildung 58: Boxplot - Dokument schreiben in Private Data (1.77MB)

Dokument lesen aus Private Data (Größe 1.77MB)

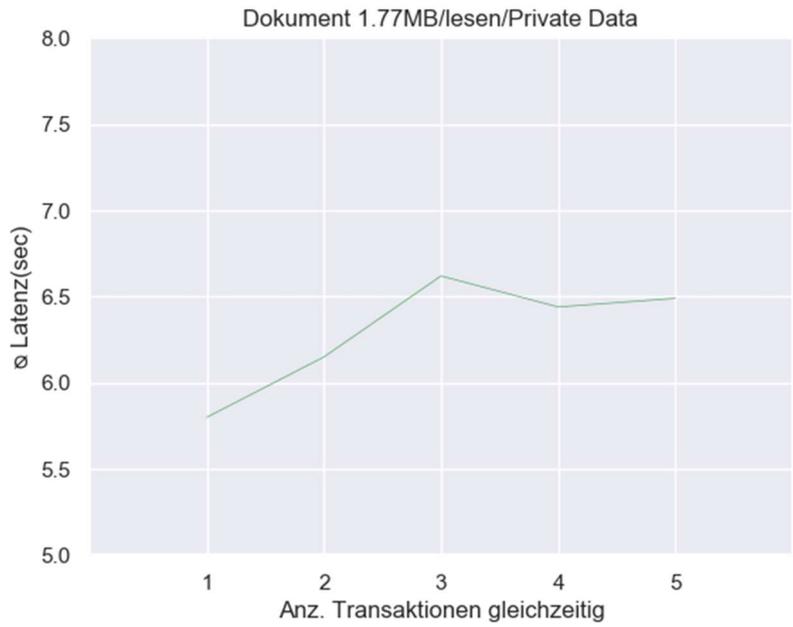


Abbildung 59: Liniengraphik - Dokument lesen aus Private Data (1.77MB)

Ø Latenz(sec)	5,80	6,15	6,62	6,44	6,49
Anz. Transaktionen gleichzeitig	1	2	3	4	5
Alle Transaktionen					
Ø Latenz(sec)	6,30				
Min Latenz(sec)	5,18				
Max Latenz(sec)	7,56				

Abbildung 60: Übersicht Latenzen – Dokument lesen aus Private Data (1.77MB)

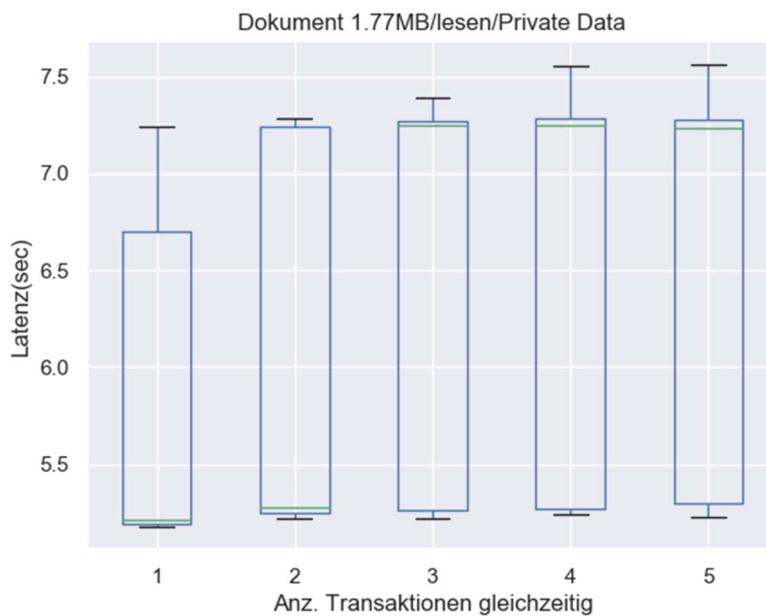


Abbildung 61: Boxplot - Dokument lesen aus Private Data (1.77MB)

Zusammenfassung der Mess-Ergebnisse

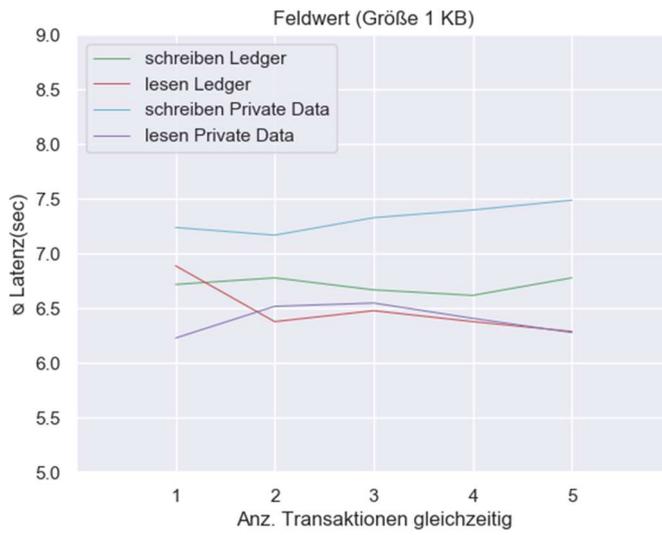


Abbildung 62: Liniengraphik - Zusammenfassung Latenzen Feldwert (1KB)

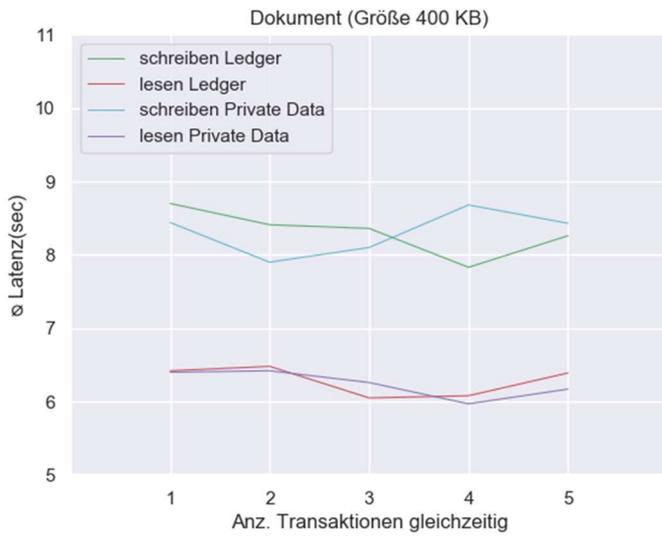


Abbildung 63: Liniengraphik - Zusammenfassung Latenzen Dokument (400KB)



Abbildung 64: Liniengraphik - Zusammenfassung Latenzen Dokument (1.77MB)

Transaktion	Latenz(sec)
Feldwert schreiben ins Ledger (1KB)	6,70
Feldwert lesen aus Ledger (1KB)	6,48
Feldwert schreiben in Private Data (1KB)	7,33
Feldwert lesen aus Private Data (1KB)	6,39
Dokument schreiben ins Ledger (400KB)	8,31
Dokument lesen aus Ledger (400KB)	6,28
Dokument schreiben in Private Data (400KB)	8,31
Dokument lesen aus Private Data (400KB)	6,24
Dokument schreiben ins Ledger (1.77MB)	11,57
Dokument lesen aus Ledger (1.77MB)	6,35
Dokument schreiben in Private Data (1.77MB)	11,45
Dokument lesen aus Private Data (1.77MB)	6,30

Abbildung 65: Ergebnisse der Latenz-Messung im Überblick

Bewertung der Mess-Ergebnisse

Die zuvor aufgeführten Ergebnisse basieren auf der Auswertung von Transaktionsdaten mehrerer Testläufe. Pro Mess-Konstellation wurden ca. 150 Transaktionen verarbeitet.

Feldwerte 1 KB

Die Mess-Ergebnisse zeigten, dass die Latenzen für das Schreiben von Feldwerten „tendenziell“ über den Zeiten für das Lesen lagen. Die Abweichung der Latenzzeiten wird darauf zurückgeführt, dass die Lese-Operation in Hyperledger Fabric direkt im Endorsing-Peer umgesetzt wird, ohne dass eine weitergehende Validierung erfolgt. Beim Lesen wird die Transaktion nur simuliert. Beim Schreiben hingegen wird die Transaktion über die Einbindung aller Endorsing-Peers vollzogen. Durch Verteilung, Abstimmung und den Konsens-Mechanismus wird über „submit“ eine Transaktion geschrieben und ein Block-Inhalt erzeugt. Aus den Ergebnissen in Bezug auf das Schreiben und Lesen von Feldwerten (1 KB) lässt sich feststellen, dass die durchschnittliche Latenz für das Schreiben und Lesen in Ledger und Private Data (CouchDB) in der vorliegenden Teststellung zwischen 6 und 7 Sekunden lag. Zu erkennen war, dass bei der Zunahme der gleichzeitigen Transaktionen das Schreiben ins Ledger im Gegensatz zum Schreiben in den Private Data Bereich (CouchDB) weniger Zeit beanspruchte und auch die Streuung geringer war.

Dokumente 400 KB / 1.77 MB

Bei der Verarbeitung von Dokument-Dateien zeigte sich ebenfalls der zuvor beschriebene Effekt der höheren Latenz für Transaktionen beim Schreiben im Gegensatz zum Lesen. Eine Steigerung der Dokumentengröße von 400 KB auf 1.77 MB erhöhte die durchschnittliche Latenz beim Schreiben um ca. 40%. Diese Tendenz lässt vermuten, dass die Zunahme der Latenz bei Steigerung der Datei-Größe nicht linear (proportional) verläuft. Dieser Effekt müsste durch weitere Tests und einer größeren Anzahl von parallel ausgelösten Transaktionen überprüft werden. Die Transaktionszeiten für das Lesen eines Dokuments blieben trotz Steigerung der Datei-Größe von 400 KB auf 1.77 MB bei Zugriffen auf das Ledger sowie den Private Data Bereich (CouchDB) relativ konstant bei ca. 6 Sekunden. Die Schreibe- und Lese-Latenzen bei beiden Datei-Größen ergaben keine Unterschiede zwischen dem Ledger und Private Data Bereich.

Die durchschnittliche Lese-Latenz aller simulierter Transaktionen von Feldwert und Dokument-Dateien aus Ledger und Private Data lag sehr eng beieinander (≈ ca. 6,5 sec.).

Als Ursache für die unterschiedliche Bandbreite der Transaktionszeiten (Max-/Min-Werte) lässt sich vermuten, dass neben der allgemeinen Netzwerklatenz die Schwankungen auch auf den Umstand zurückzuführen sind, dass auf den Client-Rechnern, auf denen die Peers installiert waren, weiter gearbeitet wurde (CPU/RAM-Engpässe). Diese Hypothese müsste in weiteren Tests überprüft werden. Zwei der CRM-Clients waren zum Testzeitpunkt über VPN mit dem Testnetz verbunden.

Wie auch in anderen Studien⁴⁴ berichtet wurde, hängen die Latenzen stark von der Netzwerk- und Hardwareumgebung ab. Um die Latenzen zu verbessern, sind weitere Kombinationen von Netzwerkanordnungen (Peers) zu testen. Die Orderer und Endorser der Fabric Blockchain könnten z.B. auf mehrere Server verteilt werden, auch aus Gründen der Ausfallsicherheit und zum Schutz vor Kompromittierung. Der Tomcat-Dienst könnte durch ein Node.js-Programm ersetzt werden und einige Peers könnten von den Anwenderclients auf verschiedene Server verteilt werden, um die Auswirkung auf die Latenzen zu prüfen. Generell ist bei allen weiteren Maßnahmen zur Steigerung der Performance (Latenz) die Zielsetzung des vorliegenden Modell-Ansatzes zu berücksichtigen sowie das sich daraus ableitende Spannungsfeld Performance, Sicherheit und Kosten.

Um die Akzeptanz bei den Anwendern zu sichern, müssen die Latenzen (Systemantwortzeiten) in einem für die Anwender vertretbaren Rahmen liegen. Um bei längeren Transaktionszeiten den Anwender nicht am Weiterarbeiten mit dem System zu hindern, sollte die Speicherung der Daten auf die Blockchain „asynchron“ im Hintergrund erfolgen. Diese Funktionalität wurde technisch noch nicht umgesetzt.

Da es sich bei den zu verarbeitenden sensiblen Informationen nur um einen Teil der Gesamtdaten aus CRM-Systemen von KMU und anderen Organisationen handelt, ist die Durchsatzrate nicht mit denen anderer Blockchain-Projekte zu vergleichen. Andere Studien weisen wesentliche kürzere Latenzen aus. Die Messungen dieser Studien beziehen sich in der Regel aber lediglich auf Latenzen innerhalb von Hyperledger Fabric und wurden mit dem Hyperledger Tool „Caliper“ gemessen. Ein weiterer Aspekt ist das im Standard die Daten unverschlüsselt auf der Hyperledger Fabric Blockchain gespeichert werden.

Fehlerrate

Die Fehlerraten, die bei den Testläufen auftraten lagen bei ca. 5%. Dabei handelte es sich überwiegend um die Meldung „Endorser nicht gefunden“.

Sicherheits-Bewertung

Die klassischen Schutzziele der IT-Sicherheit umfassen im engeren Sinne die Integrität, Authentizität, Verfügbarkeit und Vertraulichkeit einer technischen Lösung.

Die Integrität der Daten wird bei Hyperledger Fabric als Blockchain-Modell darüber sichergestellt, dass die einzelnen Blöcke mittels einer Hashfunktion miteinander verkettet werden. Hyperledger Fabric ermöglicht als private und zulassungsbeschränkte Blockchain-Variante eine schnelle Herstellung der Daten-Integrität.

Die Authentizität der Transaktionen wird bei Hyperledger Fabric über digitale Signaturen mittels eines „Private/Public-Key“ erreicht. Die Fälschung einer Signatur wird als extrem schwierig eingestuft. In Fabric werden die Betreiber der Knoten bei Erzeugung der Schlüssel in geeigneter Weise identifiziert und die Zuordnung der Schlüssel dokumentiert (Public-Key-Infrastruktur (PKI)). Dem Schutz der privaten Schlüssel kommt hierbei eine besondere Rolle zu.

Durch das Modell der dezentralen Speicherung sind die Informationen in Fabric zu jedem Zeitpunkt mit hoher Wahrscheinlichkeit verfügbar. Bei ausreichender Anzahl von Knoten können auch Teilausfälle des zugrundeliegenden Netzes kompensiert werden. Eine größere Anzahl von

⁴⁴ Xiaojiong, X. et al. (2021). Latency performance modeling and analysis for hyperledger fabric blockchain network. In: <https://www.sciencedirect.com/science/article/pii/S0306457320309298#bib0015> (Zugriff: 20.04.2021)

Konten im Blockchain-Netz kann die Resistenz des Netzwerks gegen Ausfälle aufgrund von Angriffen oder sonstigen technischen Problemen verstärken. Der Vertraulichkeit der Daten in der privat zulassungsbeschränkten Blockchain Hyperledger Fabric wird konstruktionsbedingt durch die geringere Zahl von Knoten im Gegensatz zu den öffentlichen Blockchains entsprochen. Nur zulassungsberechtigte Teilnehmer haben Zugang zu den Daten, die weiter über sogenannte „Channels“ und den „Private Data Bereich“ in Fabric eingeschränkt werden können. Im vorliegenden Modell-Ansatz werden die Daten zusätzlich verschlüsselt abgelegt. Eine Anonymisierung/Pseudonymisierung der Teilnehmer ist für das vorliegende Modell nicht gewünscht. Die Identifizierung der Knoten/Teilnehmer im Hyperledger Fabric Netz als privates Unternehmens-Netz wird vorausgesetzt.

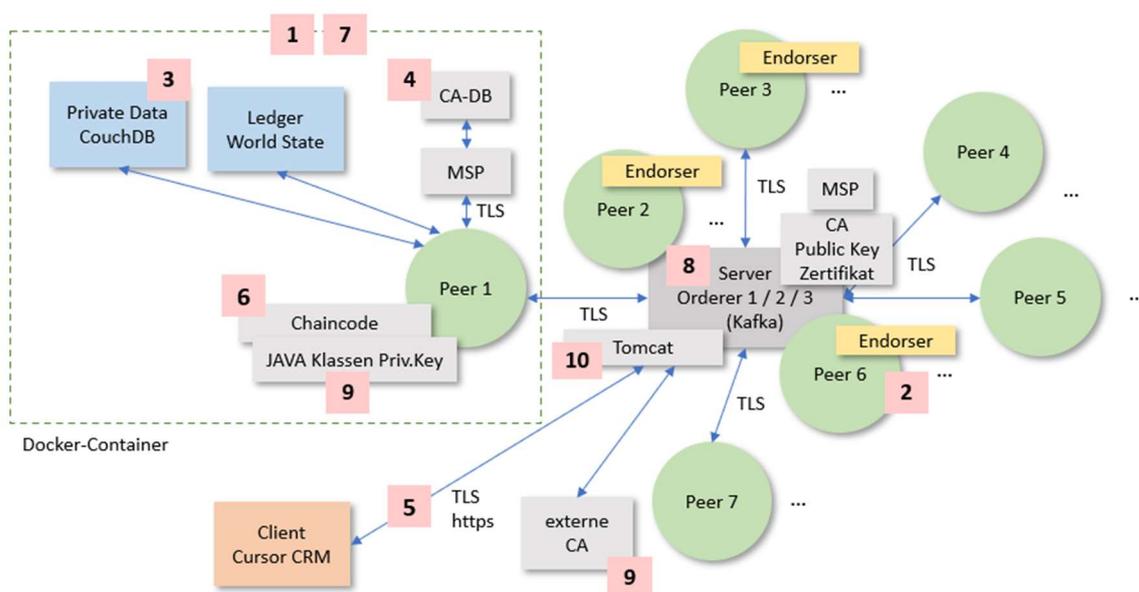


Abbildung 66: Beispiel - Technische Infrastruktur des entwickelten Modellansatzes

Die innerhalb des Hyperledger Fabric Peer-Netzwerks benötigten Netzwerkverbindungen und Ports zwischen den einzelnen Fabric-Komponenten sollten im Hinblick auf die Sicherheitsrichtlinien überprüft werden. Es sollte darauf geachtet werden, dass nur die Verbindungen und Ports nach außen geöffnet werden, die für einen erfolgreichen Betrieb unbedingt benötigt werden. Alle Komponenten des Gesamtsystems sollten regelmäßig auf Aktualität geprüft werden. Zeitnahe Installationen von Sicherheitspatches kann die Angriffsfläche auf kritische Bereiche reduzieren und erleichtert das „Troubleshooting“ im Fehlerfall. Das Hyperledger Fabric Blockchain-Framework ist sehr modular aufgebaut. Durch die hohe Modularität werden verschiedene Kombinationen von Protokollen auf verschiedene Weise genutzt. Dies erhöht die Komplexität, Angriffe zu identifizieren und entsprechend abzuwehren.

Die Sicherheitsanforderungen für das entwickelte Gesamtmodell sind mehrschichtig. Nachfolgend werden die wichtigsten Sicherheitsaspekte für den zuvor abgebildeten technischen Lösungsansatz anhand der Nummer-Indexierung beschrieben.

[1] Die Komponenten von Hyperledger Fabric (Certificate Authority, CouchDB, Kafka, ZooKeeper) können entweder als Single-Instanz oder aufgeteilt in mehrere, getrennte Instanzen bereitgestellt werden. Dafür können ein oder mehrere Docker Container verwendet werden. Eine direkte Implementierung der einzelnen Komponenten in ein Server-Betriebssystem ist ebenfalls möglich. Alle Instanzen sollten in einem eigenen Netzbereich betrieben werden. Das erleichtert die Konfiguration von Firewall-Regeln auf anderen Ebenen und reduziert den Einfluss anderer

Server-Komponenten, die üblicherweise in IT-Strukturen kleinerer Unternehmen zu finden sind (Active Directory, Anwendungsserver, Mailserver, Clients ohne Nutzungsrechte der Blockchain, etc.).

[2] Die Komponenten, wie Endorser, Oderer und Peers, lassen sich als Docker-Container Lösung bereitstellen und damit gut über zentral verwaltete Sicherheitslösungen (Client Firewall Regeln, Anti-Virus Policies) schützen. Jedoch arbeiten diese Komponenten oft mit dynamischen IP-Adressbereichen, sodass eine umfangreiche Konfiguration der Sicherheitslösungen zu berücksichtigen ist. In kleineren Peer-Netzwerken kann theoretisch auch mit statischen Adressen gearbeitet werden. Werden die Kern-Komponenten von Fabric als Docker Container betrieben, sind die dynamischen Netzbereiche zu berücksichtigen oder auf eine statische Konfiguration zu wechseln. Zudem sollte stets auf die generelle Empfehlung geachtet werden, alle Prozesse innerhalb der Container mit dedizierten Usern ohne Root-Rechte zu starten.

[3] Um Datenmanipulationen innerhalb der verwendeten CouchDB-Datenbank zu verhindern, sollte vor Inbetriebnahme des Chaincode der administrative Datenbank-Zugang angepasst und entsprechend gesichert werden. Zudem sollten keine Netzwerkzugriffe außerhalb des localhost-Bereichs erlaubt sein. Jeder Peer sollte einen individuellen Zugang zur jeweils lokal gespeicherten CouchDB erhalten. Diese Zugänge sollten ebenfalls entsprechend geschützt werden. Über die Endorsement Policy lässt sich Fabric zwar vor böswilligen, bereits manipulierten Peers schützen, jedoch bietet diese Policy keinen Schutz vor Manipulation der CouchDB Zugangsdaten.

[4] Die HyperLedger Certificate Authority (CA) stellt auf den unterschiedlichen Ebenen von Fabric grundlegende Sicherheitsmechanismen bereit, die sowohl die Netzwerk-Kommunikation zwischen Peers als auch die Datenspeicherung über den Chaincode selbst schützen. Je nach Einsatzzweck der Fabric-Komponenten (privat oder öffentlich), kann entweder die integrierte Fabric CA oder eine öffentlich betriebene Root CA genutzt werden. Letztere bietet den Vorteil auch SSL-Zertifikate anbieten zu können, die von den meisten Browsern und Clients weltweit akzeptiert werden.

Nach der Erstellung aller benötigten Zertifikate und Keys sollte auf eine solide Konfiguration der funktionsfähigen Transport Layer Security (TLS) Verbindung zwischen allen beteiligten Fabric-Komponenten geachtet werden. Die Zertifikatssperllisten (CRLs) aller Authorities (Root & Intermediate) sollten für jeden Client zu jeder Zeit abrufbar sein. Ohne diese Sperllisten lässt sich kein Zertifikat auf Gültigkeit prüfen. Ein weiterer Schritt zum Schutz vor Datenmanipulation stellt die digitale Signatur einer jeden Identität dar. Durch die Verwendung eines „Private/Public Key Paares“ zur Beglaubigung einer Transaktion, kann zu jeder Zeit nachvollzogen werden, ob Manipulationen während oder nach der Änderung stattgefunden haben. Die sichere Verwahrung des Private Keys stellt dabei einen zentralen Bestandteil des Sicherheitskonzeptes dar.⁴⁵ Ein weiterer möglicher Ablageort wäre z.B. ein separater Speicherort in einem bereits zur Verfügung stehenden Passwort Management System. Der Private Key könnte zum benötigten Zeitpunkt über eine standardisierte API in der Chaincode-Transaktion bereitgestellt werden. Eine weitere Möglichkeit wäre die Verwendung eines Hardware Security Moduls. Dieses Modul muss nur zum Zeitpunkt der Transaktionsbeglaubigung mit dem jeweiligen Client verbunden werden und könnte anschließend für den Rest der Zeit an einem sicheren Ort aufbewahrt werden. Diese Module bieten zuverlässige Verschlüsselungsmethoden zum Schutz des Private Keys bei Verlust oder Diebstahl.

Sollte der Membership Service Provider (MSP) auf Grund einer Sicherheitsverletzung durch einen Angriff kompromittiert werden, können größere Schäden entstehen.⁴⁶ Bei einem sogenannten „Sybille Angriff“ umgeht der Angreifer das Reputationssystem des Peer-to-Peer-Netzwerks durch die Erstellung einer größeren Anzahl pseudonymer Knoten. Er könnte so lange Knoten erzeugen bis er die Mehrheit hat und schließlich in der Lage ist die Endorsement-Policy

⁴⁵ Steinfeld, G. (2020). Hyperledger Fabric for the enterprise – privacy and security strategies. In: <https://www.linkedin.com/pulse/hyperledger-fabric-enterprise-privacy-security-grant-steinfeld> (Zugriff: 11.01.2021)

⁴⁶ Davenport, A. et al. (2018). Attack Surface Analysis of Permissioned Blockchain Platforms for Smart Cities. In: <https://par.nsf.gov/servlets/purl/10083311> (Zugriff: 08.01.2021)

von Fabric für betrügerische Aktionen zu nutzen.⁴⁷ Ein weiterer möglicher Angriff wäre der „ID Angriff“. Die MSP einer Organisation identifiziert die Certificate Authority (CA) die die vertrauenswürdigen Zertifikate ausstellt. Die Zertifikate erhalten ein (OU)-Feld, welches von der MSP zugewiesen wird. Über dieses Feld werden die Mitglieder einer Organisation identifiziert. Zusätzlich erfolgt über dieses Feld der Zugriff auf den Fabric-Channel. So könnte bei einem Angriff über dieses Feld Zugang zu einem Channel und dessen Transaktionen erlangt werden. Auch „Backlisting-Angriffe“ stellen eine Gefahr dar. Bei der Implementierung des MSP werden diverse Parameter gesetzt, um eine Identitätsprüfung zu ermöglichen. Ein Parameter hiervon ist eine Liste von „Certificate Revocation Lists (CRLs)“, welche jeweils einer MSP Certificate Authorities entsprechen. Diese CRLs deklarieren die Knoten, deren Rechte für das Netz gesperrt wurden. Über Zugang zum MSP könnten Angreifer über diese Parameter-Änderung die verschlüsselte Kommunikation kompromittieren.

[5] Je nach Art der Anwendung, die für eine Nutzung der Blockchain erweitert werden soll, kommen weitere Elemente in der Kommunikation hinzu. Bei der Nutzung eines Webservers für die Kommunikation zwischen den Client-Anwendungen und der Blockchain, sollte zwingend auf die Implementierung einer gesicherten Verbindung über TLS und die Verwendung eines sicheren Protokolls wie https geachtet werden. Alle SSL-Zertifikate für die Absicherung der Anwendungskommunikation sollten nicht von der Fabric CA, sondern einer separaten PKI stammen. Dies erhöht die Transparenz in der täglichen Verwaltung aller existierenden Zertifikate und verhindert ein versehentliches Vermischen von Identitäten bzw. Zugriffsrechten über die jeweilige Anwendungsgrenze hinaus. Die Verwendung einer mittels VPN abgesicherten Verbindung für Remote-Peer User bietet ebenfalls eine weitere Sicherheitsebene und schirmt den Datenstrom in öffentlichen Netzen zuverlässig ab. Da die Peers immer in der Lage sein müssen, alle anderen Komponenten von Fabric zu erreichen, muss sichergestellt werden, dass die Kommunikation im VPN auch zwischen den einzelnen Remote-Peer Usern reibungslos funktioniert.

[6] Der Chaincode in Hyperledger Fabric kann mit verschiedenen Programmiersprachen erstellt werden. Wie bei jeder anderen Programmiersprache sollte auf eine korrekte Fehlerbehandlung sowie eine nachvollziehbare Log Strategie geachtet werden. Neben der eindeutigen Definition von Fehlercodes sollte auch der Umfang der gespeicherten Fehlerdaten geprüft und angepasst werden. Sensible Daten sollten zu keiner Zeit in Fehlercodes aufgenommen werden. Gleiches gilt für die jeweiligen Log Files. Betrachtet man die innerhalb der Programmlogik verwendeten Authentifizierungsmechanismen, so sollte vor Inbetriebnahme der jeweiligen Komponenten eine Prüfung des verwendeten Quellcodes auf schadhafte Code durchgeführt werden. Bei allen Daten- Eingaben sollte sowohl eine Eingabebereinigung als auch eine Eingabebereinigung stattfinden. Die Validierung stellt sicher, dass zuvor definierte Bedingungen für Eingaben eingehalten werden und verweigert die Eingabe bei Missachtung der Vorgaben. Neben der eigentlichen Absicherung der Codebasis sollten zudem alle nicht verwendeten Funktionen und Dateien deaktiviert bzw. entfernt werden.

[7] Backup

Durch die dezentrale Struktur des Peer-Netzes bietet die Blockchain eine höhere Fehlertoleranz als klassische Client-/Server-Systeme. Fällt ein Peer aus, stehen in der Regel noch genügend Instanzen für einen störungsfreien Betrieb der Fabric-Blockchain zur Verfügung. Dennoch sollten zumindest die Server Instanz(en) regelmäßig gesichert werden. Idealerweise sollten alle Sicherungen mit Fabric-Daten in einem separaten Backup Repository gesichert und über einen Verschlüsselungsmechanismus der Backup-Software zusätzlich vor unerlaubten Zugriffen geschützt werden. Eine Sicherung der kompletten Server-Instanz(en), mit einer bereits durchgeführten Sicherheitskonfiguration von Fabric, erleichtert die Wiederherstellung der Fabric Komponenten im Störfall. Möchte man jedoch das Backup auf Ebene der Blockchain Daten

⁴⁷ Dabholkar, A. et al. (2019). Ripping the Fabric: Attacks and Mitigations on Hyperledger Fabric In: https://www.researchgate.net/publication/337276742_Ripping_the_Fabric_Attacks_and_Mitigations_on_Hyperledger_Fabric (Zugriff: 08.01.2021)

durchführen, um beispielsweise transiente Daten vom Backup auszuschließen, muss zwingend eine tiefere Analyse der Datenstrukturen erfolgen.

[8] Besteller (Orderer)

Für die Besteller (Orderer) besteht die Gefahr eines „Internal Fork-Angriffes“.⁴⁸ Bei diesem Angriff sendet der Besteller widersprüchliche Versionen von Blöcken an die Peers. Es kommt zu einer Verzögerung des Bestellprozesses (Konsenses) aufgrund der unnötigen Rechenleistung für die Verifizierung durch die Peer-Konten. Um dem Angriff vorzubeugen empfiehlt es sich die Anzahl der verworfenen Blöcke (Transaktionen) zu überwachen. Um die Ausfallsicherheit weiter zu erhöhen, könnten die Besteller auf mehrere Server verteilt werden.

[9] Verschlüsselung/Entschlüsselung

Beim vorliegenden Modell erfolgt die Ver- bzw. Entschlüsselung über den Chaincode. Die Verschlüsselung erfolgt mit Hilfe des öffentlichen Schlüssels und die Entschlüsselung über den privaten Schlüssel. Durch die Nutzung einer Certificate Authority, die durch ein Zertifikat einen öffentlichen Schlüssel mit passendem privatem Schlüssel ausstellt, wird für den Verschlüsselungsansatz keine weitere „Institution“ benötigt. Das Zertifikat wird über die Certificate Authority zur Verfügung gestellt. Der Private Key kann und darf nach IT-Sicherheits-Richtlinien nicht in einer Certificate Authority (CA) gespeichert werden. Für die Positionierung eines solchen Schlüssels bietet sich der Chaincode an, da ein privater Schlüssel sich nicht ändert und einer Organisation fest zugeordnet ist. Bei Hinterlegung des Schlüssels im Chaincode ist dieser durch den Kompilierungsvorgang im Falle von Java nur durch eine Dekompilierung auslesbar.

Der Chaincode wird innerhalb eines Secure-Docker-Containers deployed und durch den Secure-Docker-Containers eines Peers angestoßen. Der Chaincode-Container verfügt über keine Zugriffe auf ein unterliegendes Netzwerk oder das host-OS, wodurch ein Zugriff auf diesen Container von außerhalb erschwert wird. Auch bietet Hyperledger keine Möglichkeit der Dekompilierung des Chaincodes. Eine Schwachstelle wäre, wenn der Peer durch einen Angreifer übernommen wird. Dann könnte der Angreifer die Daten lesen. Dieses Problem existiert jedoch immer, wenn Peers von Angreifern übernommen werden.

[10] Tomcat

Der Tomcat-Webserver bietet in seiner Grundkonfiguration bereits ein gutes Sicherheitskonzept, welches für die meisten Anwendungsfälle ausreichend ist. In Bezug auf die Verwendung des Tomcat-Servers zusammen mit Fabric sollten jedoch zusätzliche Betrachtungen der Sicherheitsarchitektur durchgeführt werden. Tomcat-Prozesse sollten grundsätzlich mit dedizierten, Nicht-Root Userrechten ausgeführt werden. Zudem sollten diese User keine weiteren Rechte innerhalb der verwendeten Betriebssystem-Umgebung besitzen. Vor Inbetriebnahme des Servers müssen alle veralteten und als unsicher eingestuften Protokolle (SSLv3, TLS1.0, etc.) deaktiviert und alle nicht verwendeten Funktionen bzw. Beispielkonfigurationen entfernt werden. Zusätzliche Sicherheit bietet außerdem die Verwendung von Realms, die zur Steuerung von Zugriffsrechten innerhalb der Tomcat- Umgebung benutzt werden können.

Langzeitsicherheit

Die Erreichung einer Langzeitsicherheit gerade für sensible Daten in einer Blockchain ist besonders anspruchsvoll. Durch ein geeignetes Konzept zur Kryptoagilität sollte sichergestellt sein, dass die Sicherheitsmechanismen der Blockchain durch neuere technische Mechanismen zur Laufzeit austauschbar sind.⁴⁹ Die Gewährleistung der Sicherheit für langlebige Daten kann bei privat, zulassungsbeschränkten Blockchain-Modellen, wie Hyperledger Fabric, besser erzielt

⁴⁸ Putz, B. et al. (2020). Detecting Blockchain Security Threats In: <https://epub.uni-regensburg.de/43942/1/AcceptedManuscript.pdf> (Zugriff: 08.01.2021)

⁴⁹ Berghoff, C. et al. (2019). Blockchain sicher gestalten, Bundesamt für Sicherheit in der Informationstechnik (BSI) S. 42ff.. In: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Krypto/Blockchain_Analyse.pdf?__blob=publicationFile&v=5 (Zugriff: 16.08.2020)

werden, da sich die Anforderungen dazu über die administrative Stelle und einer überschaubaren Netzwerk-Größe besser erfüllen lassen.

5.7.4 Anpassung Prototypen erneuter Testlauf

Nach umfassender Evaluierung der Performance und Sicherheit der aufgezeigten Teststellung wurden im Anschluss weitere Optimierungsmaßnahmen erarbeitet und überprüft.

Dazu zählte z.B. die Variation der Speichereinstellung im Tomcat-Webserver. Die Veränderung der Speichereinstellung im Tomcat-Dienst ergab keine signifikante Auswirkung auf die Latenzen. Eine mögliche Einflussnahme des Viren-Scanners auf die Latenzen wurde durch die Inaktivierung des Viren-Scanners für die Bereiche Docker-Container und den Hyperledge Fabric Server überprüft.

Dies ergab ebenfalls keine messbaren Auswirkungen auf die Latenzen. Auch die Verlagerung der Client-Verbindungen aus dem VPN in das lokale Netz brachte keine nennenswerte Veränderung im Performancebereich. Die Auswertung der Firewall-Überwachung während der umfangreichen Testläufe im Rahmen der Evaluation zeigte keine auffälligen Last-Spitzen durch das Blockchain-Netzwerk.

5.8 Roll-out im Testunternehmen - Bundesverband deutscher Banken (BdB), Berlin

Machbarkeitsstudien werden oft von Vertretern aus Forschung und Entwicklung geleitet und entwickelt. Die Untersuchung erfolgt meist unter kontrollierten Bedingungen. Der Übergang zur Überprüfung in der Praxis mit möglichen Erwartungskorrekturen erfordert die Einbeziehung von Akteuren aus der Realwelt. Da sich die entwickelten technischen Ansätze noch auf einem rudimentären Niveau befinden, konnte der Praxistest nicht im vollen Umfang mit Anwendern durchgeführt werden. Während des Entwicklungsprozesses und der internen Testläufe zeichnete sich ab, dass noch einige Funktionalitäten zu entwickeln sind, um die Anwender Usability zu erhöhen und Fragen zum Prozessablauf bei unterschiedlichen Konstellationen der Speicherung zu klären sind. Zur Überprüfung der entwickelten technischen Ansätze in einem „ersten Praxistest“ konnte der „Bundesverband der deutschen Banken (BdB)“⁵⁰ in Berlin für eine „Teststellung“ gewonnen werden. Als wirtschaftspolitischer Spitzenverband übernimmt der Bundesverband deutscher Banken die Interessenvertretung der privaten Banken in Deutschland. Zu seinen Aufgaben gehören unter anderem die Information seiner Mitgliedsbanken und -verbände, Lobbyarbeit in der Politik sowie die Zusammenarbeit mit anderen Verbänden, beispielsweise in der Deutschen Kreditwirtschaft.

5.8.1 Identifikation Prototypen und Onboarding im Testunternehmen

Für den Testlauf beim Bundesverband deutscher Banken (BdB) sollte die Schnittstellen-Variante 2: „Schnittstellen-Logik über Tomcat“ und die privat zulassungsbeschränkte Hyperledger Fabric Blockchain zum Einsatz kommen. Als CRM-System stand beim BdB das Cursor-CRM zur Verfügung, welches sich zum Zeitpunkt des Testlaufs noch in der Einführungsphase befand. Aufgrund der aktuellen Pandemie und des Einführungsstatus des CRM-Systems verzögerte sich der Termin für den geplanten Testlauf.

Über einen Onboarding-Prozess wurden die Funktionsweise des technischen Ansatzes und die geplante Integration in die Systemlandschaft sowie der Termin für den Testlauf mit dem Leiter des IT-Managements des BdB abgestimmt. Aus dem CRM-System des BdB sollten über die gewählte Schnittstellen-Logik sensible Informationen aus der BdB-Datenlandschaft über das CRM-System auf die Blockchain geschrieben und gelesen werden. Neben Feldwerten mit unterschiedlichen Funktionen aus den CRM-Eingabemasken sollten auch Dokumente des Bankenverbands in den Testlauf einbezogen werden.

5.8.2 Roll-out und Integration der technischen Ansätze im Testunternehmen

Das Peer-to-Peer Netz für die Hyperledger Fabric Blockchain V. 2.3.0 des BdB wurde auf einem Verbund eigener Rechner/Server als Teststellung aufgesetzt. Die Implementierung der Schnittstelle erfolgte über den Tomcat-Webservice. Der Tomcat-Dienst wurde auf dem Server aufgesetzt. Es kamen zwei CRM-Testclients zum Einsatz. Das Peer-Netz umfasste 5 Peers von welchen 3 als Endorser fungierten. Zwei Endorser-Peers wurden auf dem Server betrieben. Die Endorsement-Policy umfasste die Vorgabe 2 von 3, d.h. mindestens 2 Endorser mussten für die Transaktionsbestätigung zustimmen. Für den Bestellservice wurden 3 Orderer (Kafka) auf dem Server installiert. Die Blockgröße des Ledgers betrug 2 MB. Die Verschlüsselung der Transaktionsdaten erfolgte über Schlüssel einer externen CA. Für das Logging wurde ein Messprogramm auf den CRM-Clients installiert.

⁵⁰ <https://bankenverband.de/>

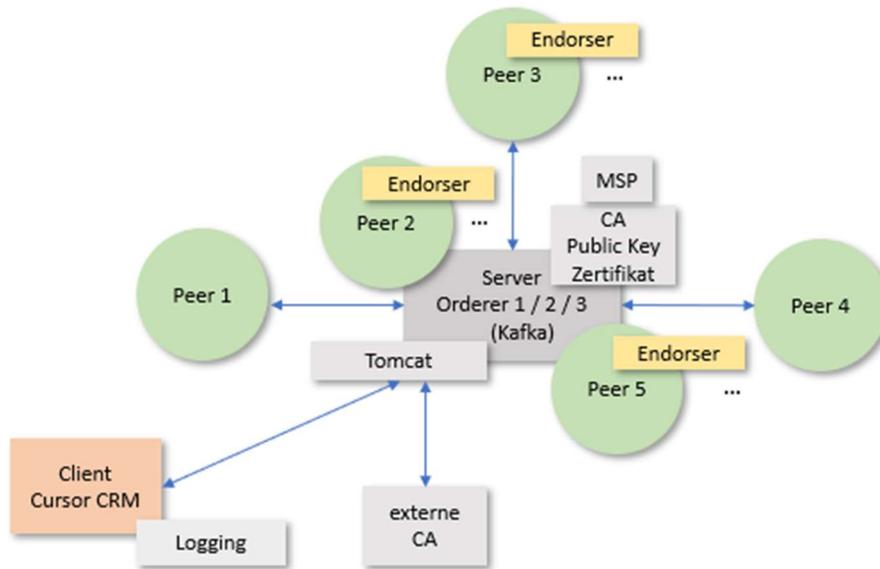


Abbildung 67: Teststellung Bundesverband deutscher Banken (BdB)

5.8.3 Test Prototyp im Testunternehmen

Der Testlauf beim Bundesverband deutscher Banken wurde gemeinsam mit Herrn Kraatz, dem Leiter des IT-Managements des Bankenverbands, über MS-Teams durchgeführt. Da für den potentiellen Anwender die dahinterstehende Technik des vorliegenden Ansatzes weitgehend verborgen bleibt, wurden die verschiedenen technischen Abläufe und Zusammenhänge des Gesamt-Modells zuvor in einer kurzen Präsentation aufgezeigt.

Der Test basierte auf verschiedenen Szenarien aus der BdB-Praxis.

- a) Speichern/Lesen eines Feldwertes einer Eingabemaske aus dem BdB CRM-System
- b) Speichern/Lesen eines Feldwert-Schlüssels einer Eingabemaske
- c) Speichern/Lesen eines Dokuments

Zu a)

Aus einer Eingabemaske für Firmierung von Kontaktdaten wurde ein Standard-Feldwert (ca. 1KB) auf die Blockchain geschrieben und gelesen.

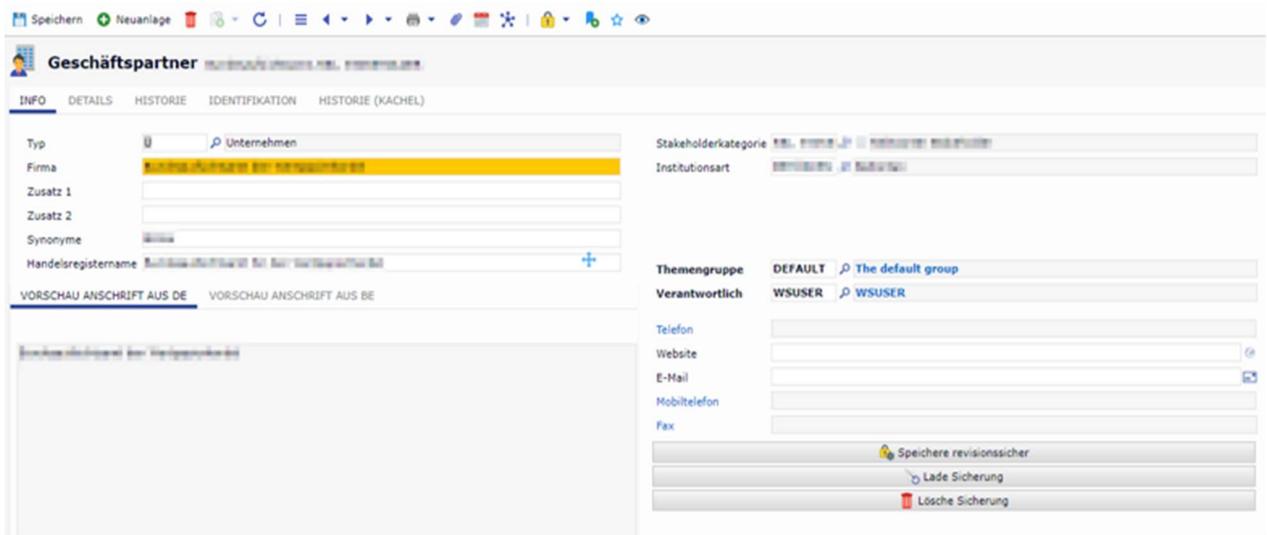


Abbildung 68: CRM-Eingabemaske Firmierung von Kontaktdaten

Zu b)

Aus einer Eingabemaske für die Gremienverwaltung wurde ein Feldwert, der die Rolle eines Schüssels/Verweis auf einen anderen Datensatz übernahm, auf die Blockchain geschrieben und gelesen. Die Feldwert-Anzeige wurde in der Eingabemaske durch „XXXXXXX“ vor unberechtigtem Zugriff geschützt. Bei der Verlinkung handelte es sich um den Bezug zu einem sensiblen Datensatz. Es wurde nur der Feldwert als Schlüssel/Link zu dem verbundenen Datensatz auf der Blockchain gespeichert.

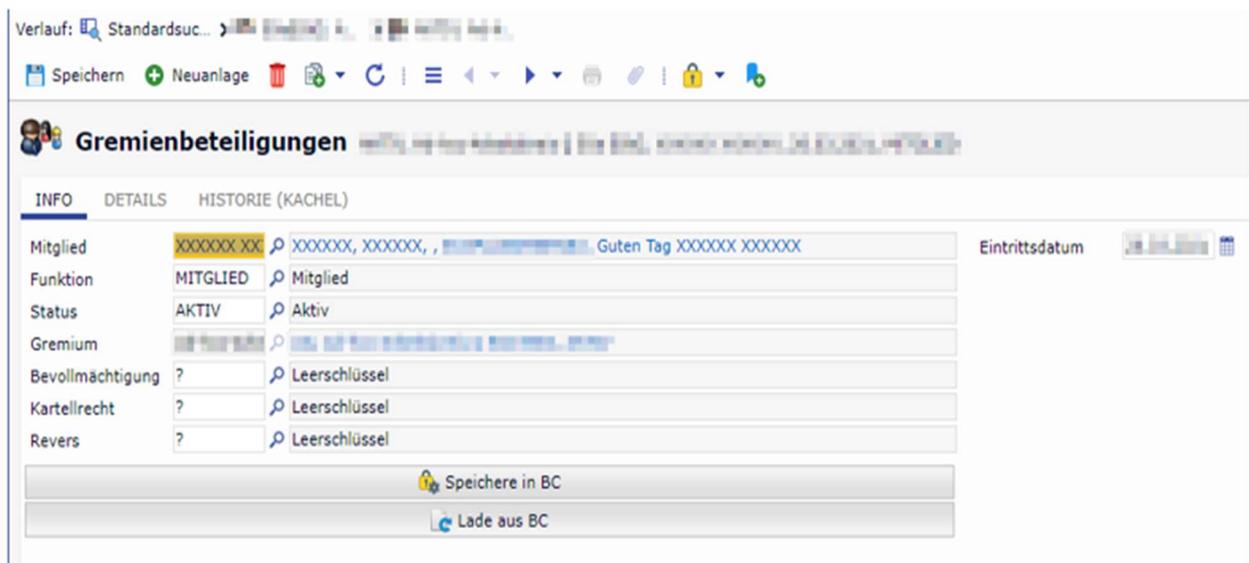


Abbildung 69: CRM-Eingabemaske „Gremienbeteiligung“

Zu c)

Aus dem Dokumenten-Pool des BdB wurde das Dokument „Bilanzierung“ zur Speicherung auf die Blockchain ausgewählt. Das Dokument umfasste eine Größe von ca. 67 KB. Bei den Testläufen zur Speicherung des Dokuments kam es zu Systemzuständen, bei denen die Verarbeitung der Transaktion abgelehnt wurde. Die Ablehnung der Transaktion könnte z.B. damit zusammenhängen, dass ein validierender Peer temporär keine Netzwerkverbindung hatte. Nach unmittelbar erneutem Versuch wurde die zuvor abgelehnte Transaktion erfolgreich verarbeitet.

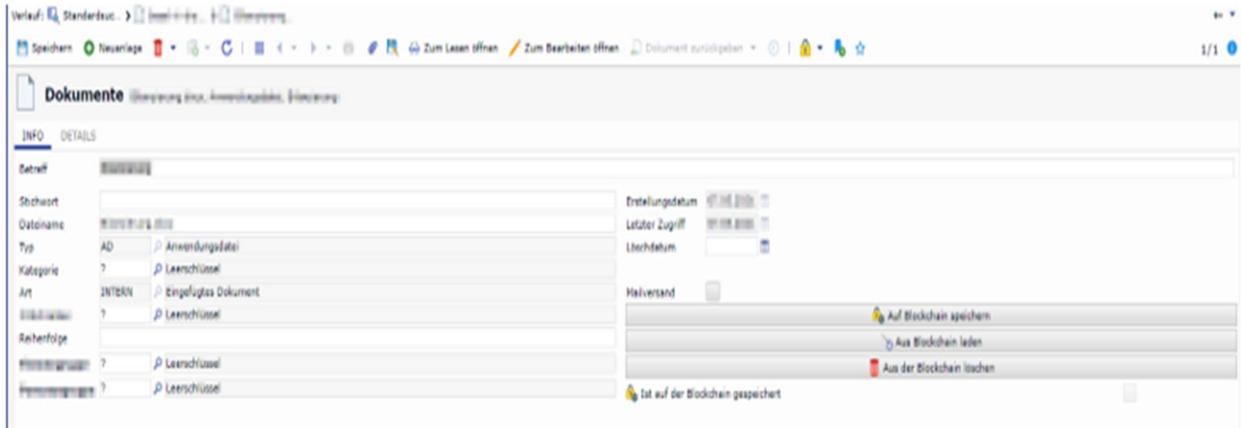


Abbildung 70. Dokumentenspeicherung aus CRM-System

Die durchschnittlichen Latenzen des Testlaufs betragen:

Transaktion	Latenz(sec)
Feldwert schreiben ins Ledger (1KB)	5,80
Feldwert lesen aus Ledger (1KB)	5,10
Feldwert schreiben in Private Data (1KB)	6,72
Feldwert lesen aus Private Data (1KB)	5,90
Dokument schreiben ins Ledger (67KB)	6,22
Dokument lesen aus Ledger (67KB)	4,83
Dokument schreiben in Private Data (67KB)	6,92
Dokument lesen aus Private Data (67KB)	6,62

Abbildung 71: Durchschnittliche Latenzen des BdB-Testlaufs

Der Latenzen lagen im Bereich der Werte, die bei der Evaluierung/Leistungsbewertung des Modellansatzes gemessen wurden.

5.9 Gruppendiskussion Testteilnehmer

Nach Durchführung des Testlaufs wurden die Ergebnisse und die Eindrücke in Form einer offenen Diskussion beurteilt. Die Diskussion wurde über MS-Teams geführt, und dauerte ca. 30 Minuten. Für die Diskussion gab es keinen festen Leitfaden, sondern nur richtungsweisende Themenpunkte.

- Allgemeine Eindrücke Usability, Latenzen etc.?
- Userverhalten und Datenströme beim BdB?
- Varianten für die Auslösung der Speicherung?
- Speicherung der sensiblen Informationsobjekte nur auf der Blockchain?
- Speicherung von verknüpften Informationen (Link) auf der Blockchain?
- Wie sollen die Peers verteilt werden?
- Fazit / Ausblick

Das Ziel der Diskussion war, neben den ersten Eindrücken zur allgemeinen Anwender Usability, vor allem auf die Implementierung praxistauglicher Workflows für die unterschiedlichen Speicherszenarien der spezifischen Informationsobjekte ausgerichtet. Nach der Diskussion der verschiedenen Umsetzungsvarianten sollte zum Abschluss ein Fazit und Ausblick erfolgen.

5.9.1 Zusammenfassung der Ergebnisse / Reflexion

Im Folgenden werden die wichtigsten Ergebnisse aus der Gruppendiskussion zum Testlauf beim Bundesverband deutscher Banken zusammenfassend dargestellt.

In Bezug auf die Latenz-Zeiten (Performance) für das Schreiben und Lesen der Informations-Objekte merkte der Leiter des IT-Managements des BdB an, dass die Verteilung für die Schreib- und Leseprozesse in der BdB-Praxis als hochgradig „lesend“ einzustufen sei.

Zum Ablauf und Design der verschiedenen durchgespielten Speicherszenarien gab es einige Anmerkungen seitens des Bankenverbands. Da die Informationen auf der Blockchain unveränderlich gespeichert werden, ist die Speicherung von z.B. „Tippfehlern“ nicht direkt zu löschen, sondern nur durch erneutes Speichern zu korrigieren. Hierzu stellte sich die Frage, ob für sensitive Felder, die auf der Blockchain gespeichert werden, die Speicherung über einen speziellen „Button“ vom Anwender separat ausgelöst werden soll, um die Aufmerksamkeit zu erhöhen oder, ob die Daten zusammen mit den anderen Feldwerten automatisch gespeichert werden sollen. Der BdB merkte hierzu an, dass das Auslösen über einen zusätzlichen Button in der Praxis keinen Vorteil bringen würde. Die Speicherung der sensitiven Daten auf die Blockchain sollte automatisch mit allen anderen Daten erfolgen. Versehentlich gespeicherte, falsche Eingaben auf die Blockchain wurden als nicht kritisch eingestuft.

Ein weiterer Punkt war die Frage, ob z.B. sensitive Dokumente nach dem Speichern auf der Blockchain parallel noch in der CRM-Datenbank vorgehalten werden sollten oder nicht. Herr Kraatz vom BdB stufte die Vorgehensweise unterschiedlich ein. Die Speicherung der sensitiven Dokumente oder Feldwert-Informationen müsste von einem Gutachter als „Revisionssichere Speicherung“ bestätigt werden, dann könnten z.B. Rechnungen etc. nur auf der Blockchain vorgehalten werden. Es sei denn die abgelegten Informationen würden für „Massenabfragen“ benötigt. Dann sei es aus Gründen der Performance besser sie zusätzlich in der CRM-Datenbank vorzuhalten. Eine weitere Möglichkeit wurde darin gesehen, die wichtigsten Metadaten für Abfragen (z.B. Kategorien, Personengruppen) in der Datenbank zu halten und die eigentlichen Nutzdaten sicher auf der Blockchain zu speichern. Das Verbergen von Feldinhalten, z.B. durch Anzeige von XXXX-Werten in der CRM-Eingabemaske wäre besonders für Adressdaten von z.B. hochrangigen Gremienmitgliedern interessant. Der Feldwert würde dann nur mit Berechtigung zum Lesen bzw. Ändern von der Blockchain abgefragt.

Einigkeit herrschte bei der Beurteilung der Blockchain in Bezug auf die Ausfall- und Änderungssicherheit und dass die Technik diesen Mechanismus ohne Zusatz-

Implementierungen mitbringt. Um die Sicherheit zu erhöhen müsste die Anzahl der Client-Peers entsprechend erhöht werden, was sich jedoch nachteilig auf die Performance durch höhere Latenzen auswirken könnte. Bei der Verteilung der Peers auf die Client-Rechner wies der Leiter des IT-Managements des BdB daraufhin, dass ein ausgewogener Mix zwischen Client-Rechner und Server für die Peers sicherzustellen sei, da außerhalb der Kernarbeitszeiten und z.B. am Wochenende unter Umständen nicht genügend Peer-Clients zum Betreiben der Blockchain beim BdB zur Verfügung stehen könnten.

Zusammenfassend wurde konstatiert, dass für einen reibungslosen Ablauf und für die Bedienung durch den (End-)Anwender in der Praxis noch weitere technische Feinabstimmung und Workflow-Szenarien zu erarbeiten sind. In Bezug auf das Aufsetzen der Hyperledger Blockchain-Umgebung wurde festgehalten, dass das Aufsetzen viel Sachverstand verlangt und seitens der Entwickler-Community von Hyperledger Tools zur Vereinfachung der Installation bereitgestellt werden sollten. Der Leiter des IT-Managements des BdB schlug vor, in einem Follow-Up Termin die weitere Vorgehensweise zu der eingesetzten Technik zu besprechen.

5.10 Ableitung Ressourcenbedarf, Aufwand, Kosten, Nutzen

Bezogen auf das entwickelte Gesamtmodell soll im Nachfolgenden der damit verbundene Ressourcenbedarf sowie Aufwand und Kosten skizziert werden. Die nachfolgende Aufstellung bezieht sich auf die Lösungsvariante: Blockchain auf einem Verbund eigener Rechner/Server, Schnittstellen-Logik über Tomcat und Hyperledger Fabric Blockchain.

Beim Betreiben der Hyperledger Fabric Blockchain auf einer Cloud (z.B. bei IBM Public Cloud) fallen hauptsächlich Kosten für den Betrieb der virtuellen Maschinen an. Bei der klassischen Public Cloud wird der Server als sog. "shared instance" betrieben, d.h. auf einer Hardware-Plattform, die sich mehrere Kunden teilen. Berechnet wird die Server-Instanz selbst und dann noch die verbrauchten CPU-Zeiten. Die Kosten für die kleinste Installation, die für Testzwecke bei der IBM Public Cloud für Hyperledger Fabric angemietet wurde, belief sich auf ca. 700,- EUR/Monat.

Ressourcen

Hardware

- Hardware muss für Docker geeignet sein
- Server für Hyperledger Fabric-Instanz
- nach Möglichkeit mehrere Server für die Verteilung der Endorser-Peers
- Client-Rechner für die Peers

Software

- Tomcat-Webserver (kostenfrei)
- Hyperledger Fabric Blockchain (Open-Source, kostenfrei)
- Betriebssystem (Windows/Linux)
- Docker
- CRM-Software
- Schnittstellen-Logik (CRM/Hyperledger Fabric)

Personal

- IT-Administrator (intern)
- Externer IT-Dienstleister

Die Installation der Software-Komponenten sowie die Durchführung von Updates und der Support von Clients und Server kann von einem internen IT-Administrator oder einem externen IT-

Dienstleister übernommen werden. Der interne IT-Administrator müsste sich das benötigte Know-how aneignen. Ein Teil des erforderlichen Know-hows kann aus der Studie entnommen werden.

Aufwand

Der Aufwand umfasst den Initialaufwand für das Aufsetzen der software-technischen Komponenten sowie den Aufwand für die Sicherstellung des störungsfreien Betriebs.

Die nachfolgende Aufstellung zeigt die benötigten Arbeitspakete für die initiale Implementierung des Gesamtmodells.

	<u>ca. Tag(e)/Aufwand</u>
1. Kick-off / Konzeption Analyse bestehende Hardwareumgebung (Clients, Server), Analyse Netzwerk, CA Verteilung der Arbeitspakete Planung Hyperledger Fabric Topologie (wo und wie werden die unterschiedlichen Peers aufgesetzt?)	4
2. Installation Hyperledger Fabric Server (Docker-Container)	0,5
3. Implementierung Hyperledger Fabric Blockchain auf Server (Erstellung Zertifikate, Dimensionierung Anzahl Peers etc.)	3-7
4. Konfiguration Netzwerk zum Betrieb (Ports, VPN, etc.)	1
5. Erstellung Schnittstellen-Logik (CRM/Hyperledger Fabric) (Erstellung Chaincode und Scripting, Anbindung an das CRM etc.)	7
6. Inbetriebnahme Schnittstellen-Logik (CRM-System/Fabric)	1-5
7. Roll-out / Verteilung Peers	1
8. Testbetrieb	7
9. Freigabe zum Produktivbetrieb	

Der Aufwand für Wartung und Support umfasst die Einspielung von Updates, den Support der Clients, das Ausrollen von Paketen, die Aktualisierung der Docker-Containern etc.

Kosten

Für die benötigten Softwarekomponenten (Hyperledger Fabric, Tomcat) fallen keine Lizenzkosten an. Für die skizzierten Aufwandstage sind die internen Tagesätze bzw. die der externen Dienstleister anzusetzen.

Um den Aufwand für die Installation der Hyperledger Fabric Blockchain zu reduzieren wäre es sinnvoll ein spezielles „Softwareimplementierungstool“ zu entwickeln. Mit diesem Tool könnten ähnlich wie bei der IBM-Cloud, über eine Eingabemaske diverse Parameter, z.B. Anzahl der Peers, Peer-Funktionen, Endorsement-Policy, CA, Felder der CRM-Software, Dokumentengröße etc. abgefragt werden und anschließend ein fertiges Hyperledger Fabric Framework automatisch erstellt werden. Die erstellten Docker-Container für Clients und Server könnten als Images bereitgestellt werden. Ziel wäre es den Ablauf noch einfacher und komfortabler zu gestalten.

Der Einsatz eines solchen Tools könnte den Aufwand bei der Erstinstallation und der laufenden Wartung wesentlich reduzieren. Die Aufwände der Schritte 3,5,6 und 7 würden sich signifikant reduzieren. Die Entwicklung dieses Tools wäre in einem weiteren Projekt zu realisieren. Diese Erweiterung war eine wichtige Erkenntnis aus der vorliegenden Studie.

Nutzen

Das vorliegende Gesamtmodell ermöglicht die sichere Speicherung sensibler Informationen aus einem CRM-System auf die Hyperledger Fabric Blockchain.

- Erhöhung der Ausfallsicherheit und Schutz von Datenverlust durch verteilte Datenhaltung
- Verstärkter Schutz vor Cyberangriffen durch verteilte Strukturen
- Änderungssicherheit / Revisionssicherheit durch Blockchain-Mechanismus
- Kostengünstige Variante (keine Lizenzkosten)
- Ressourcen-Schonung durch Nutzung bestehender Infrastruktur (Hardware)
- Transparenz über Datenstatus und ausgeführte Aktionen
- Konsolidierungsregeln beim Schreiben von Daten sind beim Blockchain-Ansatz individuell regelbar
- Einhaltung der DSGVO-Anforderungen

Die Latenzen des vorliegenden Ansatzes sind aufgrund der umfassenden Sicherheits- und internen Systemabstimmungsprozesse sowie der dezentralen Datenverteilung nicht mit denen einer herkömmlicher Datenbanken zu vergleichen. Auch die Skalierbarkeit ist eingeschränkt und es kann zu temporären Einschränkungen bei der Verfügbarkeit kommen. Dafür bietet der vorliegende Ansatz aber eine kostengünstige und sichere Speicherung sensibler Informationen. Die Latenzen für das Lesen liegen deutlich unter den Latenzen für das Schreiben. Der Ansatz eignet sich unter dem Aspekt der Performance besonders für Konstellationen, bei denen die Lesezugriffe den Schreibprozessen überwiegen.

6 Fazit und Ausblick

Das Ziel der Machbarkeitsstudie war die Entwicklung und Überprüfung technischer Ansätze zur Speicherung sensibler Informationen aus CRM-Systemen auf einem Blockchain-Netzwerk. Der Einsatzzweck ist auf kleine und mittlere Unternehmen und sonstige Organisationen ausgerichtet und soll sich bei überschaubarem Kostenrahmen einfach und wartungsarm implementieren lassen und den Anforderungen der Datenschutzgrundverordnung (DSGVO) entsprechen. Es sollte geprüft werden, welches Blockchain-Modell und welche herstellerunabhängige Schnittstellen-Logik dafür in Frage kommt. Die technischen Ansätze hierfür wurden nur auf einem sehr rudimentären Niveau mit dem Ziel der Überprüfung der Funktionstüchtigkeit entwickelt. Um zunächst einen ersten Eindruck über die Kenntnisse der potentiellen Zielgruppe zur „Blockchain-Technologie“ zu erhalten, wurden mehrere Experteninterviews geführt und ausgewertet. Aus den Ergebnissen konnten weitere Impulse für die Konzeption und Entwicklung der Ansätze abgeleitet werden. Als CRM-Systeme für die Teststellungen wurde das CRM-System von Cursor und das Microsoft Dynamics 365 CRM-System ausgewählt. Da anzunehmen ist, dass ein Teil der Daten aus den CRM-Systemen einen Personenbezug aufweist, waren besonders die Anforderungen der Datenschutzgrundverordnung (DSGVO) zu beachten, was die Auswahl bzw. das Design der Blockchain maßgeblich mitbestimmte. Nach einer umfangreichen Recherche und Überprüfung potentieller Blockchain-Modelle fiel die finale Auswahl auf das private zulassungsbeschränkte Blockchain-Modell Hyperledger Fabric der Linux Foundation. Dieses Blockchain-Framework erfüllte zum Zeitpunkt der Studie die Anforderungen des Datenschutzes am weitesten im Vergleich zu allen anderen betrachteten Modellen. Darüber hinaus überzeugte der hochmodulare Aufbau, die Programmierung der „Chaincodes“ in gängigen Programmiersprachen sowie die große und weltweite Community. Über den eingesetzten Konsensmechanismus von Hyperledger Fabric konnten die Transaktionskosten (Energieverbrauch) niedrig gehalten werden.

Dass sich das Aufsetzen der Blockchain-Umgebung auf einem Verbund eigener Rechner/Server einfacher gestaltete als zuvor angenommen, zeigte sich als eine der zentralen Erkenntnisse aus dieser Studie. Auch das Debugging der Fabric Blockchain auf eigenen Systemen gestaltete sich effizienter als bei der überprüften IBM Public Cloud Lösung für Fabric. Bei den Tests zum möglichen Einsatz der IBM-Cloud für das Hyperledger Fabric Netzwerk zeichnete sich ab, dass noch einiges an Experten Know-how für ein erfolgreiches Implementieren der Blockchain in der Cloud mit weiteren Peers in Clientnetzen benötigt wird. Die Kosten für die kleinste Einheit der Public IBM-Cloud für Hyperledger Fabric bezifferten sich zum Zeitpunkt der Studie auf ca. 700 EUR/Monat. Das Aufsetzen der Blockchain auf eine Private Cloud bei IBM bedarf einer getrennten Anfrage bzw. eines individuellen Angebots. Die Kosten dürften dementsprechend höher ausfallen als bei der Public-Variante. Aus dem veröffentlichten Bericht von Bitkom Research und KMPG „Cloud-Monitor (2020)“⁵¹ ist zu entnehmen, dass der Cloud-Einsatz bei Unternehmen in den letzten Jahren zwar zugenommen hat, aber weiterhin gerade bei der Public Cloud-Variante Sicherheitsbedenken bestehen. Die Mehrzahl der Unternehmen fürchten einen unberechtigten Zugriff auf sensible Unternehmensinformationen und die Rechtslage wird als unklar bewertet. Aus diesen Erkenntnissen und im Hinblick auf die Zielsetzung der vorliegenden Studie wurde beschlossen den Schwerpunkt der Untersuchung auf den Einsatz der Blockchain auf einem Verbund eigener Rechner und Server zu setzen.

Während der Entwicklung der Schnittstellen-Logik stellte sich heraus, dass die Variante „Schnittstellen-Logik über Tomcat-Dienst“ in Bezug auf die Anforderung einer einfachen und kostengünstigen Implementierung sowie der angestrebten Herstellerunabhängigkeit am besten eignete. Bei der Anbindung der CRM-Systeme über die Schnittstellen-Logik zeigte sich, dass die Datenstrukturen eine besondere Rolle spielen.

Die anschließende Evaluierung des Gesamtsystems in einem umfassenden Testlauf ergab Latenzen, die nicht mit denen anderer Studien zu vergleichen sind. Neben der generellen Varianz aufgrund von unterschiedlicher Hardwareausstattung und Netzwerktopologie ist hier vor allem auf die erweiterte Mess-Schleife, die eine „End-to-End“ Messung bei der vorliegenden Teststellung vollzog, zu verweisen. Es wurde vom CRM-System über die Schnittstelle (Tomcat)

⁵¹ <https://www.bitkom.org/Presse/Presseinformation/Drei-von-vier-Unternehmen-nutzen-Cloud-Computing#:~:text=Berlin%2C%202023,.,Jahr%202017%20erst%2066%20Prozent.> (Zugriff: 07.04.2021)

zur Blockchain und wieder zurück bis zum Ausgangspunkt des CRM-Systems gemessen. Die gemessenen Werte (Latenzen) sind für die erste Ausprägung des Modell-Ansatzes akzeptabel. Sie sollten jedoch weiter durch Anpassung der Konfiguration unter Berücksichtigung des Spannungsfeldes Performance und Sicherheit und Kosten optimiert werden. Über die Sicherheitsanalyse des Gesamtmodells wurden mögliche Schwachstellen und Angriffsvektoren der Blockchain aufgezeigt und Mindeststandards für die Sicherheit beschrieben. Es ist erforderlich, diesen Bereich iterativ auf den Prüfstand zu stellen und in Abhängigkeit der Konfiguration zu bewerten. Für die „CIA Schutzziele“ konnte unter dem Gesichtspunkt der Informationssicherheit folgende Erkenntnis gewonnen werden. Die Ziele Vertraulichkeit (Confidentiality) und Integrität (Integrity) können über die Mechanismen der Blockchain erfolgreich sichergestellt werden. Für das Ziel der Verfügbarkeit (Availability) muss angemerkt werden, dass es im Bereich der Transaktionsverarbeitung der Blockchain-Lösung zu kurzfristigen Störungen der Verfügbarkeit der Daten kommen kann, z.B. aufgrund von Störungen im Blockchain-Netzwerk, der vielschichtigen Sicherheitsarchitektur und der verteilten Datenstruktur. Bezogen auf die Langzeitsicherheit ist die hoch modulare Architektur des ausgewählten Blockchain-Frameworks Hyperledger Fabric mit seiner großen und weltweiten Community als positiv zu bewerten.

Der erste Praxistest beim Bundesverband deutscher Banken in Berlin bestätigte die Funktionstüchtigkeit des Gesamtmodells und zeigte, neben den offenen Punkten für eine vereinfachte Implementierung vor allem die fachlichen Anforderungen, die bei der Anbindung eines CRM-Systems entstehen und zu lösen sind. Speichert ein CRM-System z.B. Informationen auch über Referenzen, dann müssen je nach fachlichen Anwendungsfall sowohl die Referenz als auch die dort hinterlegten Datenfelder verschlüsselt gespeichert werden. Ein weiteres Beispiel stellt die Anforderung für „Massenprozesse“ in CRM-Systemen dar. Werden z.B. Serienmails oder Serienbriefe erzeugt, die Informationen verarbeiten, die auf der Blockchain gespeichert wurden, so sind die Schnittstellenaufrufe anders als in den getesteten Szenarios im System zu implementieren.

Die Ableitung der Faktoren Ressourcen, Aufwand und Kosten soll als erster Orientierungsmaßstab dienen. Der in der Studie entwickelte Gesamtansatz kann auch als generische Referenzarchitektur für weitere Entwicklungen in diesem Bereich gesehen werden. Aus ökologischen Gesichtspunkten macht es durchaus Sinn die intern vorhandene technische Infrastruktur für das Blockchain-Netzwerk von Unternehmen und anderen Organisationen zu nutzen, da externe Rechenzentren in der Regel einem enormen Ressourcen- und Energieverbrauch unterliegen. Hinsichtlich einer validen ökologischen Betrachtung ist es allerdings derzeit noch schwierig, die Vorteile der Digitalisierung den Nachteilen durch die Bereitstellung von IKT und Netzinfrastruktur aussagekräftig zu bewerten. Eine vollumfängliche Überprüfung ökologischer, nachhaltiger Aspekte müsste in einer weiteren Studie untersucht werden.

Die Machbarkeitsstudie weist somit nach, dass die Speicherung sensibler Daten aus CRM-Systemen auf ein Blockchain-Netzwerk mit den aufgezeigten technischen Ansätzen unter Berücksichtigung der Zielformulierungen umsetzbar ist.

Wie für jede junge Technologie gilt auch für die Blockchain, dass die Akzeptanz gegenüber der Technologie bei Unternehmen und Gesellschaft erst wachsen muss und Ängste sowie Vorurteile durch Aufklärung abgebaut werden müssen. Die vorliegende Machbarkeitsstudie soll auch hierzu einen Beitrag leisten.

Die Entwicklung der technischen Ansätze und die durchgeführten Testläufe ermutigen die Entwicklung fortzuführen. Aufbauend auf den Erkenntnissen dieser Studie soll in einem folgenden Projekt ein „Softwareimplementierungstool“ entwickelt werden, um die Implementierung und Wartung des Hyperledger Fabric Netzwerks auf einem Verbund eigener Rechner und Server weiter zu vereinfachen bzw. zu automatisieren. Über eine Eingabemaske könnten mit diesem Tool diverse Parameter (z.B. Anzahl Peers, Endorser, Endorsement-Policy, Blockgröße etc.) abgefragt werden und im Anschluss die Umgebung (Docker-Container) für das Hyperledger Fabric Netz automatisch erstellt werden. Zusätzlich könnte mit diesem Tool die laufende Wartung der Lösung vereinfacht werden. Beispiele wären hier das Hinzufügen von Feldern in der CRM-Anwendung und deren Speicherung auf der Blockchain oder das Hinzufügen von Peers im Blockchain-Netz. Zur Bestandsaufnahme der IT-Infrastruktur von Unternehmen und sonstigen Organisationen sollte ein Leitfaden entwickelt werden, der die systematische Erfassung der vorliegenden Hardware-Komponenten und Netzwerkstrukturen unterstützt. Zur fachlichen Integration der Schnittstellen-Logik in das CRM-System sind weitere Standard-Workflows für die unterschiedlichen Speicherszenarien zu erarbeiten. Das Error-Handling ist in Bezug auf die Aussagefähigkeit der Fehlermeldung zu verbessern, um die Fehleranalyse und Beseitigung zu beschleunigen. Um Dokumente speichern zu können, die die Blockgröße der Blockchain übersteigen, sollte ein Programmcode entwickelt werden, der das Dokument zur Laufzeit beim Speichern auf mehrere Blöcke verteilt und beim Lesen wieder zusammenführt.

Zur Verbesserung der Latenzen sind weitere Anordnungen der Hyperledger Fabric Komponenten und der Schnittstellen-Logik zu prüfen. So könnte der Tomcat-Dienst durch ein Node.js Programm ersetzt werden, um Auswirkungen auf die Latenzen in Erfahrung zu bringen.

Die Entwicklung der Cloud-Technologie für das Aufsetzen einer Blockchain-Umgebung ist im Hinblick auf die Sicherheit sowie rechtliche Aspekte und bezüglich der Kostenentwicklung weiter zu beobachten. Das gesamte Potential der Blockchain-Technologie kann sich jedoch nur richtig entfalten, wenn der Gesetzgeber entsprechende Spielräume schafft. So hat Hyperledger Fabric neben dem eigentlichen Hauptbuch (Ledger) zusätzlich den „Private Data Bereich“ geschaffen, um den wichtigsten Anforderungen der DSGVO, dem Bearbeiten und Löschen von Daten, zu entsprechen. Derzeit sind viele Anwendungsszenarien für eine Blockchain vorstellbar, kollidieren aber oft mit den bestehenden Datenschutzerfordernissen. Über den Gesetzgeber müssen hier Anpassungen vorgenommen werden, damit mögliche Entwicklungen der Blockchain-Technologie nicht schon im Keim erstickt werden. Die rechtlichen Rahmenbedingungen spielen hierbei eine entscheidende Rolle, um Deutschland als attraktiven Standort für Blockchain-Anwendungen auszubauen.

7 Literaturverzeichnis

Andola, N.; Manas Gorgoi, R.; Venkatesan, S.; Verma, S. (2019). Vulnerabilities on Hyperledger Fabric In: <https://www.sciencedirect.com/science/article/abs/pii/S157411921830720X> (Zugriff: 08.01.2021)

Androulaki, E.; Cocco, S.; Ferris, C. (2018). Private and confidential transactions with Hyperledger Fabric. <https://developer.ibm.com/technologies/blockchain/tutorials/cl-blockchain-private-confidential-transactions-hyperledger-fabric-zero-knowledge-proof/> (Zugriff: 14.12.2020)

Berghoff, C.; Gebhardt, U.; Lochter, M.; Maßberg, S. (2019). Blockchain sicher gestalten, Bundesamt für Sicherheit in der Informationstechnik (BSI). In: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Krypto/Blockchain_Analyse.pdf?__blob=publicationFile&v=5 (Zugriff: 16.08.2020)

Bitkom - Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V.. Drei von vier Unternehmen nutzen Cloud-Computing. In: <https://www.bitkom.org/Presse/Presseinformation/Drei-von-vier-Unternehmen-nutzen-Cloud-Computing#:~:text=Berlin%2C%2023.,Jahr%202017%20erst%2066%20Prozent.> (Zugriff: 07.04.2021)

Bundesamt für Sicherheit in der Informationstechnik (BSI), Technische Richtlinie - IT Sicherheit. In: https://www.bsi.bund.de/DE/Publikationen/TechnischeRichtlinien/technischerichtlinien_node.html / (Zugriff: 16.08.2020)

Bundesministerium für Bildung und Forschung (BMBF). Digitalisierung und Nachhaltigkeit. In: <https://www.bmbf.de/de/digitalisierung-und-nachhaltigkeit-10466.html> (Zugriff: 18.11.2020)

Dabholkar, A.; Sarawat, V. (2019). Ripping the Fabric: Attacks and Mitigations on Hyperledger Fabric. In: https://www.researchgate.net/publication/337276742_Ripping_the_Fabric_Attacks_and_Mitigations_on_Hyperledger_Fabric (Zugriff: 08.01.2021)

Davenport, A.; Shetty, S.; Liang, X. (2018). Attack Surface Analysis of Permissioned Blockchain Platforms for Smart Cities. In: <https://par.nsf.gov/servlets/purl/10083311> (Zugriff: 08.01.2021)

Gentemann L. (2019). Studienbericht: Blockchain in Deutschland – Einsatz, Potentiale, Herausforderungen. Studie der Bitkom e.V. In: https://www.bitkom.org/sites/default/files/2019-06/190613_bitkom_studie_blockchain_2019_0.pdf (Zugriff: 16.08.2020)

Goldhammer, K.; Weigand, A.; Lehr, S. (2018). Kompass IT-Verschlüsselung. Orientierungs- u. Entscheidungshilfen für KMU zum Einsatz von Verschlüsselungslösungen. Studie im Auftrag des BMWi. In: https://www.bmwi.de/Redaktion/DE/Publikationen/Studien/kompass-it-verschluesselung.pdf?__blob=publicationFile (Zugriff: 16.08.2020)

Kerkmann, J. (2020). Was steckt hinter der Business Blockchain Quorum?. In: <https://blockchainwelt.de/was-steckt-hinter-der-business-blockchain-quorum/> (Zugriff: 18.11.2020)

Kunde, E.; Kaulartz, M.; Naceur, M.R.B.; Liban, S.; Kunz, M.; Skwarek, V.; Adam, K.; Weiß, R., Liesenjohann, M. (2017). Blockchain und Datenschutz, Faktenpapier h im Auftrag vom KPMG. In: <https://www.bitkom.org/sites/default/files/file/import/180502-Faktenpapier-Blockchain-und-Datenschutz.pdf> (Zugriff: 08.10.2020)

Kuzlu, M.; Pipattanasomporn, M.; Gurses, L.; Rahman, S. (2019). Performance Analysis of a Hyperledger Fabric Blockchain Framework: Throughput, Latency and Scalability. In: <https://ieeexplore.ieee.org/abstract/document/8946222/authors#authors> (Zugriff: 20.11.2020)

Le Hors, A. (2019). Demystifying Hyperledger Fabric Ordering and decentralization. In: <https://developer.ibm.com/articles/blockchain-hyperledger-fabric-ordering-decentralization/> (Zugriff: 20.11.2020)

Lincoln, N.K.. IBM Blockchain Developer Tools. Hyperledger Fabric 1.4.0 Performance Information Report. In: https://hyperledger.github.io/caliper-benchmarks/fabric/resources/pdf/Fabric_1.4.0_javascript_node.pdf (Zugriff: 04.12.2020)

Martini, M.; Weinzierl, Q. (2017). Die Blockchain-Technologie und das Recht auf Vergessenwerden. In: <https://www.uni-speyer.de/fileadmin/Lehrstuehle/Martini/BlockchainundRechtAufVergessenwerdenTyposkriptversion20-03-19NZ.pdf> (Zugriff: 12.10.2020)

Melnik, I. (2020). Comparison: Ethereum vs Hyperledger Fabric vs Corda. In: <https://merehead.com/blog/comparison-ethereum-hyperledger-fabric-r3-corda/> (Zugriff: 19.10.2020)

Meuser, M.; Nagel, U. (2010). Experteninterviews - wissenssoziologische Voraussetzungen und methodische Durchführung. In: Barbara Friebertshäuser, Heike Boller und Sophia Richter (Hg.): Handbuch qualitative Forschungsmethoden in der Erziehungswissenschaft. 3., vollst. überarb. Aufl., (Neuausg.). Weinheim: Juventa-Verlag

Polge, J.; Robert J.; Le Traon, Y. (2020). Permissioned blockchain frameworks in the industry: A comparison. In: <https://www.sciencedirect.com/science/article/pii/S2405959520301909> (Zugriff: 18.11.2020)

Pols, A.; Vogel, M. (2017). Cloud Monitor 2017, Eine Studie vom Bitkom Research im Auftrag vom KPMG. In: <https://www.bitkom.org/sites/default/files/pdf/Presse/Anhaenge-an-PIs/2017/03-Maerz/Bitkom-KPMG-Charts-PK-Cloud-Monitor-14032017.pdf> (Zugriff: 16.08.2020)

Prinz, W.; Schulte, A. (2017). Blockchain – Technologien, Forschungsfragen und Anwendungen. Positionspapier Fraunhofer Gesellschaft. In: https://www.aisec.fraunhofer.de/content/dam/aisec/Dokumente/Publicationen/Studien_TechReports/deutsch/FhG-Positionspapier-Blockchain.pdf (Zugriff: 17.10.2020)

Putz, B.; Pernul, G. (2020). Detecting Blockchain Security Threats In: <https://epub.uni-regensburg.de/43942/1/AcceptedManuscript.pdf> (Zugriff: 08.01.2021)

Schmitz, P. (2019). Definition Permissioned / Private Blockchain. In: <https://www.blockchain-insider.de/was-ist-eine-permissioned-blockchain-a-871313/#:~:text=Eine%20Permissioned%20Blockchain%20ist%20eine,Personen%20angesehen%20und%20%C3%BCberpr%C3%BCft%20werden.> (Zugriff: 17.10.2020)

Schiller, K. (2019). Die Blockchain-Typen im Überblick. In: <https://blockchainwelt.de/blockchain-typen-ueberblick/> (Zugriff: 23.10.2020)

Schmitz, P. (2019). Was ist eine Public Blockchain? In: <https://www.blockchain-insider.de/was-ist-eine-public-blockchain-a-871294/> (Zugriff: 23.10.2020)

Steinfeld, G. (2020). Hyperledger Fabric for the enterprise – privacy and security strategies. In: <https://www.linkedin.com/pulse/hyperledger-fabric-enterprise-privacy-security-grant-steinfeld> (Zugriff: 11.01.2021)

Xiaogiong, X.; Long, L.; Huilong, C.; Yu, H. (2021). Modellierung und Analyse der Latenzleistung für das Hyperledger Fabric Blockchain-Netzwerk. In: https://www.researchgate.net/publication/347833428_Latency_performance_modeling_and_analysis_for_hyperledger_fabric_blockchain_network (Zugriff: 08.01.2021)

Xiaogiong, X.; Gang S.; Long, L.; Huilong, C.; Honfang, Y.; Athanasios, V. Vasilakos (2021). Latency performance modeling and analysis for hyperledger fabric blockchain network. In: <https://www.sciencedirect.com/science/article/pii/S0306457320309298#bib0015> (Zugriff: 20.04.2021)

Zeiselmaier, A.; Bogensperger, A.. Technologie, Aufbau und Begrifflichkeiten im Kontext der Blockchain. In: <https://www.ffe.de/themen-und-methoden/digitalisierung/929-technologie-aufbau-und-begrifflichkeiten-im-kontext-der-blockchain> (Zugriff: 27.10.2020)

8 Anlage

Interviewleitfaden

Digitalisierung – Speicherung sensibler Informationen mit Hilfe der Blockchain-Technologie am Beispiel von CRM-Systemen

1. Begrüßung
2. Vorstellung
3. Erläuterung des Forschungsvorhabens
4. Organisatorisches

5. Offene Leitfragen

5.1 Was wissen Sie über die Blockchain-Technik zur Datenspeicherung?

5.2 Lässt sich Ihrer Meinung nach, das geplante technische Vorhaben in Ihrer Systemumgebung betreiben?

5.3 Welche Vor-/Nachteile sehen Sie bei dem geplanten Vorhaben?

5.3 Wie beurteilen Sie die mögliche Akzeptanz für den beschriebenen technischen Ansatz in Ihrem Unternehmen/Organisation?

5.6 Wie speichert Ihr Unternehmen/Organisation sensible Daten und sind Sie mit der Lösung zufrieden?

5.7 Wo sehen sie weitere Möglichkeiten für den Einsatz der Blockchain- Technologie?

5.8 Wie schätzen Sie die Entwicklung der Blockchain-Technik in der Zukunft ein?

6. Verabschiedung