Sachbericht Teil I - Kurzbericht

TRUSTDBLE – THE TRUSTED DATABASE

Zuwendungsempfänger:	Förderkennzeichen:
Data Management Lab der TU Darmstadt	16KIS1267

Laufzeit des Vorhabens:

01.01.2021 – 28.02.2023

Autoren:

Simon Karrer, Benedikt Völker, Prof. Dr. Carsten Binnig

Das Projekt „TrustDBle – The Trusted Database" wurde als eine StartUpSecure-Förderung (Phase I) durch das Bundesministerium für Bildung und Forschung (BMBF) gefördert und durch wissenschaftliche Mitarbeiter am Fachgebiet Data Management des Fachbereichs Informatik an der Technischen Universität Darmstadt durchgeführt.

Ziel des Projektvorhabens TrustDBle war es, eine neue Plattform für ein vertrauenswürdiges Datenmanagement bereitzustellen, um Anwendungsfälle für eine gemeinsame Nutzung von Daten besser und schneller umzusetzen. Neben einer standardisierten Schnittstelle basierend auf der weitverbreiteten Abfragesprache SQL soll TrustDBle die Möglichkeit bieten, sogenannte verifizierbare Datenbankprozeduren (Data Contracts) zu definieren, welche Anwendungen die verbindliche Umsetzung von individuellen Nutzungsvereinbarungen oder gesetzlichen Vorgaben beim Zugriff auf gemeinsame Daten garantieren.

Das Vorhaben knüpfte an wissenschaftliche Vorarbeiten am Fachgebiet Data Management an. Zu nennen sind hier die Publikationen *BlockchainDB - A Shared Database on Blockchains*[1] und *BlockchainDB - Towards a Shared Database on Blockchains*[2] von Muhammad El-Hindi, die auf den führenden Datenbankkonferenzen SIGMOD und VLDB veröffentlicht wurden. Kern der Publikationen war die Kombination einer Speicherschicht, basierend auf Blockchain, mit einer Datenbankschicht, welche Datenbank-Techniken wie Sharding nutzt und eine Put-Get-Schnittstelle bereitstellt. Ein weiterer wissenschaftlicher Baustein war die Masterarbeit von Simon Karrer, in der bereits die Konzepte für die verifizierbare Transaktionsverarbeitung erarbeitet wurden. Die Ergebnisse wurden auf dem Symposium on Foundations and Applications of Blockchain als Publikation *TrustDBle: Towards Trustable Shared Databases*[3] veröffentlicht.

Basierend auf den wissenschaftlichen Vorarbeiten, wurde mit TrustDBle ein System entwickelt, welches Techniken aus den Bereichen Datenbank-Systeme und Blockchains vereint, um ein vertrauenswürdiges Management von gemeinsam genutzten Daten zu ermöglichen. Klassische Datenbank-Systeme sind dafür ausgelegt, Daten schnell und effizient zu verwalten. Allerdings sind sie nicht für die gemeinsame Nutzung von mehreren Organisationen entwickelt und schaffen keinen vertrauenswürdigen Umgang von Daten. Blockchains hingegen bieten durch ihren dezentralen Aufbau einen sehr hohen Grad an Sicherheit und Vertrauen, da die Daten auditierbar und manipulationssicher gespeichert werden. Eine schnelle und hohe Datenverarbeitung ist aber nicht gegeben und daher sind Blockchains oft nicht ausreichend für aktuelle Geschäftsprozesse. Um auf die gemeinsam genutzten Daten sicher und vertrauenswürdig aber auch schnell und effizient zuzugreifen, nutzt TrustDBle hardwarebasierte vertrauenswürdige Ausführungsumgebungen (Trust Execution Environment - TEE) und Konzepte wie

[1] El-Hindi, M., Binnig, C., Arasu, A., Kossmann, D., & Ramamurthy, R. (2019). BlockchainDB: A Shared Database on Blockchains. Proc. VLDB Endow., 12(11), 1597–1609. doi:10.14778/3342263.3342636

[2] El-Hindi, M., Heyden, M., Binnig, C., Ramamurthy, R., Arasu, A., & Kossmann, D. (2019). BlockchainDB - Towards a Shared Database on Blockchains. Proceedings of the 2019 International Conference on Management of Data, 1905–1908. Presented at the Amsterdam, Netherlands. doi:10.1145/3299869.3320237

[3] El-Hindi, M., Karrer, S., Doci, G., & Binnig, C. (2020). TrustDBle: Towards trustable shared databases. Third International Symposium on Foundations and Applications of Blockchain.

Partitionierung. TEEs ermöglichen eine manipulationssichere und überprüfbare Ausführung von Programmcode. TrustDBle bietet durch einen zwei-schichtigen Aufbau zum einen die Möglichkeit eine standardisierte SQL-Schnittstelle bereitzustellen, über diese der Nutzer wie mit einer traditionellen Datenbank interagieren kann. Zum anderen wird die Komplexität von Blockchains reduziert und eine einfache Verwendung von unterschiedlichen Blockchains zur vertrauenswürdigen Datenspeicherung möglich.

Die Daten-Plattform wurde auf Basis von dem OpenSource Datenbankmanagementsystem MySQL umgesetzt und erweiterte dieses durch ein Storage-Engine-Plugin, welches es ermöglicht, Blockchains als Speicher für Daten in einer Datenbank zu nutzen. Dadurch können mehrere Parteien geteilte Datenbanken und Tabellen erstellen, ihnen beitreten oder weitere Parteien einladen. Eine Verbindung zwischen den einzelnen TrustDBle Instanzen wird durch die darunterliegende Blockchain Netzwerke hergestellt. Die Änderungshistorie geteilter Daten ist für die beteiligten Parteien einsehbar und wird fälschungssicher hinterlegt. Durch Partitionierung und Replikation mehrerer Blockchain Netzwerke kann die Leistungsfähigkeit erhöht werden. Die Plattform unterstützt Ethereum und Hyperledger Fabric als Blockchain Systeme, die im Storage-Engine-Plugin als Speicher genutzt werden können. Neben geteilten Datenbanken können in TrustDBle Data Contracts erstellt werden, welche Logik definieren, die auf geteilten als auch lokalen Daten einer Partei vertrauenswürdig ausgeführt werden kann. Als Laufzeitumgebung kommt eine TEE zum Einsatz, um die korrekte Ausführung der Logik für den Datenhalter und den Nutzer des Data Contracts zu garantieren. Eine standardisierte SQL-Schnittstelle sowie eine einfach zu bedienende Benutzeroberfläche stehen dem Nutzer für die Bedienung von TrustDBle zur Verfügung. Die technische Implementierung von TrustDBle wurde unter der MIT Lizenz als Open Source Projekt auf GitHub veröffentlicht: https://github.com/DataManagementLab/trustdble

Für das TrustDBle System gibt es zahlreiche Anwendungsmöglichkeiten. Folgend werden zwei Anwendungsbeispiele erläutert, welche die Vorteile verschiedener Aspekte von TrustDBle hervorheben.
Um die CO2-Emissionen von Batterien zu senken und wertvolle Rohstoffe recyceln zu können, fordert die Batterieverordnung ab spätestens 2027 einen digitalen Produktpass für Batterien. Für die Umsetzung ist es erforderlich, dass Daten von unterschiedlichen Unternehmen gemeinsam und nachvollziehbar abgelegt werden und abrufbar sind. Hierfür können die geteilten Datenstrukturen von TrustDBle genutzt werden, welche die technischen Anforderungen bereits erfüllen. Im Rahmen der Qualitätssicherung im Automobilsektor sind Hersteller verpflichtet zu überprüfen, ob verwendete Materialien und Bauteile gefährliche und gesundheitsschädliche Stoffe beinhalten. Das führt häufig zu einem Konflikt, denn die stoffliche Zusammensetzung von Bauteilen sind häufig intellektuelles Eigentum von Zulieferern, welches diese nicht teilen. Mittels Data Contracts können Zulieferer den Automobilherstellern die Prüfung der Inhaltsstoffe ermöglichen, ohne ihnen Zugriff auf eine Liste der Inhaltsstoffe geben zu müssen. Damit können Automobilhersteller ihre gesetzlich erforderlichen Prüfungen durchführen und Zulieferer müssen ihr intellektuelles Eigentum nicht offenlegen.

Im Zusammenhang mit dem Vorhaben wurden vier Publikationen auf internationalen Konferenzen und Workshops veröffentlicht und vorgestellt:

ACID-V: Towards a New Class of DBMSs for Data Sharing, El-Hindi, M., Zhao, Z., & Binnig, C. (2021)

Benchmarking the Second Generation of Intel SGX Hardware, El-Hindi, M., Ziegler, T., Heinrich, M., Lutsch, A., Zhao, Z., & Binnig, C. (2022)

Towards a Benchmark for Shared Databases [Vision Paper], El-Hindi, M., Arora, A., Karrer, S., & Binnig, C. (2022)

TRUSTDBLE: Towards a New Class of DBMSs for Data Sharing, El-Hindi, M., Karrer, S., Völker, B., Binnig, C. (2022)

Außerdem sind fünf Bachelor- und Masterarbeiten in Verbindung mit dem Projektvorhaben entstanden und erfolgreich abgeschlossen worden.

Die Ergebnisse des Vorhabens TrustDBle fließen somit in die Forschung und Lehre am Fachgebiet Data Management ein. Die Weiterentwicklung des TrustDBle Systems wird in einer StartUpSecure (Phase 2) Förderung weitergeführt.

Sachbericht Teil II – Eingehende Darstellung

TRUSTDBLE – THE TRUSTED DATABASE

Zuwendungsempfänger:	Förderkennzeichen:
Data Management Lab der TU Darmstadt	16KIS1267
Laufzeit des Vorhabens:	
01.01.2021 – 28.02.2023	
Autoren:	
Simon Karrer, Benedikt Völker, Prof. Dr. Carsten Binnig	

Das Projekt „TrustDBle – The Trusted Database" wurde als eine StartUpSecure-Förderung (Phase I) durch das Bundesministerium für Bildung und Forschung (BMBF) gefördert und durch wissenschaftliche Mitarbeiter am Fachgebiet Data Management des Fachbereichs Informatik an der Technischen Universität Darmstadt durchgeführt.

Ziel des Projektvorhabens TrustDBle war es, eine neue Plattform für ein vertrauenswürdiges Datenmanagement bereitzustellen, um Anwendungsfälle für eine gemeinsame Nutzung von Daten besser und schneller umzusetzen. Neben einer standardisierten Schnittstelle basierend auf der weitverbreiteten Abfragesprache SQL soll TrustDBle die Möglichkeit bieten, sogenannte verifizierbare Datenbankprozeduren (Data Contracts) zu definieren, welche Anwendungen die verbindliche Umsetzung von individuellen Nutzungsvereinbarungen oder gesetzlichen Vorgaben beim Zugriff auf gemeinsame Daten garantieren.

Projektarbeiten

Arbeitsbereich: Speicherebene

AP01

Wie geplant wurde zunächst die Blockchain-Abstraktionsebene konzipiert und eine einheitliche Schnittstelle definiert, die es ermöglicht, eine beliebige Blockchain für die Speicherung von Daten zu verwenden. Diese dient dazu Lese- und Schreiboperation auf einer beliebigen Blockchain auszuführen. Um die Leistungsfähigkeit zu steigern und effizienter Daten zu lesen bzw. zu schreiben wurde eine Bündelung von Anfrage mit in die Schnittstelle integriert.
Die Implementierung dieser Schnittstelle und Abstraktion wurde in C++ umgesetzt und für Ethereum und Hyperledger Fabric durchgeführt und passende Adapter erstellt. Die geplante Erstellung eines Adapters für die Blockchain Hyperledger Sawtooth wurde nicht durchgeführt, da dieses System nicht mehr so stark genutzt wird und weitestgehend von Hyperledger Fabric abgelöst wurde. Dahingegend wurde ein Adapter implementiert, der eine Blockchain auf dem lokalen System durch die einfache Speicherung der Daten in einer Datei simuliert. Zum Debugging und Testen neuer Funktionen kann dieser so sehr schnell und einfach genutzt werden, ohne dabei ein ganzes Blockchain-Netzwerk aufzubauen.

AP02

Die Partitionierung der Daten auf mehrere Blockchain-Netzwerke wurde durch einen Hashing-Ansatz umgesetzt. Es wurde das statische als auch das dynamische Partitionieren in TrustDBle implementiert. Dadurch kann eine gemeinsam genutzte Datenbank mit einer festen Anzahl an Blockchain-Netzwerken erstellt werden, wobei die Daten anhand des Hashes des primären Schlüssels auf unterschiedliche Partitionen gleichmäßig verteilt werden. Im Betrieb einer bestehenden gemeinsam genutzten Datenbank kann durch die dynamische Partitionierung die Anzahl der Partitionen geändert werden. Die Daten werden dabei auf die neue Anzahl an Blockchain-Netzwerken erneut partitioniert. Die Replizierung der Daten auf mehrere Partitionen wurde nicht umgesetzt, da durch die Arbeit an den Partitionierungsstrategien ersichtlich wurde, dass dies keine Vorteile bietet und die Blockchain-

Netzwerke in sich die Daten schon vollständig replizieren, was vollkommend ausreichend ist. Zusätzlich wurde in diesem Arbeitspaket eine weitere Komponente, der Blockchain Manager, entwickelt, um beliebige Blockchain-Netzwerke sehr einfach und schnell automatisiert zu Erstellen und zu Verwalten. Der Blockchain Manager wird bei der statischen als auch bei der dynamischen Partitionierung verwendet, wenn Netzwerke hinzu oder entfernt werden. Da es hier große Probleme und es noch keine passende Lösung gab, ist die Arbeit am Blockchain Manager in einer Masterarbeit gemündet, in welcher ein generelles System zum automatisierten Orchestrieren von verschiedenen Blockchain-Netzwerken entwickelt wurde.

AP03

Die anfangs vorgesehenen Verifikationsprotokolle für eine Online- oder Offline-Verifikation wurden nicht wie geplant umgesetzt, da die transaktionale Verarbeitung von Anfragen schon durch die Separierung in Blockchain-Transaktionen die Verifikation durch die Blockchain abbildet. Somit ist eine Online-Verifikation schon bei der Ausführung von Transaktionen integriert und eine Offline-Verifikation wie sie in der Vorarbeit erarbeitet wurde nicht notwendig. Deswegen wurde in diesem Arbeitspaket für die Auditierung der Änderungen an der gemeinsam genutzten Datenbank eine zusätzliche Monitor-Komponente implementiert, die die Blockchain und deren manipulationssicheren verknüpften Blöcke zur Auditierung nutzt. Diese Komponente bietet einen vollständigen Nachweis über alle Änderungen, die an der gemeinsam genutzten Datenbank durchgeführt wurden. Dieser Nachweis beinhaltet wer welche Änderung wann durchgeführt hat. Außerdem ist es möglich den Stand der Daten zu jeder Änderung separat einzusehen und nachzuvollziehen welche Daten geändert wurden.

Arbeitsbereich: Datenbankebene

AP04

Datenbanktransaktionen sollen auf den gemeinsam genutzten Daten überprüfbar ausgeführt werden können. Dafür wurde die Korrektheit der Ausführung mit Bezug auf die ACID-Eigenschaften als die bestätigte Speicherung aller Abfragen einer Datenbanktransaktion in der Blockchain definiert. Daraus folgt, dass eine Datenbanktransaktion in mehrere Blockchain-Transaktionen separiert wird und diese einzeln in der Blockchain ausgeführt werden. Schlägt eine dieser Transaktionen fehl, schlägt auch die Datenbanktransaktion fehl. Um die transaktionalen SQL-Anfragen zum Lesen oder Schreiben von Daten verarbeiten zu können, wurden im Storage-Engine Plugin weitere Funktionalitäten hinzugefügt, die einen lokalen Cache und ein Rollback oder Commit ermöglichen. Die Verifikation einer Datenbanktransaktion wird, wie zuvor definiert, dabei durch die darunterliegende Blockchain abgebildet. Dadurch ist es möglich verifizierbare Datenbanktransaktion auf den gemeinsamen Daten auszuführen. Die Funktionsfähigkeit und Leistungsfähigkeit wurden mit Hilfe der Anwendung des OLTP-Benchmarks TPC-C auf einer gemeinsamen genutzten Datenbank getestet. Dabei ist die Problematik aufgekommen, dass klassische Datenbankbenchmarks nur für zentrale Datenbanksysteme ausgelegt sind und ausschließlich Leistungseigenschaften messen. Doch die wesentlichen Eigenschaften einer gemeinsam genutzten Datenbank wie Verifikation, Vertrauenswürdigkeit sowie Sicherheit werden nicht gemessen und evaluierbar gemacht. Die Masterarbeit A New Benchmark for Shared Databases thematisiert diese Problematik und wurde auch als Publikation veröffentlicht.

AP05

Die Verifikation von analytischen Anfragen wird durch die Verwendung der bestätigten Änderungen in der Blockchain und dem neusten Datenzustand ermöglicht. Eine aktive Verifikation ist nicht notwendig, da bei analytischen Anfragen direkt von der Blockchain gelesen wird und festgeschriebene Änderungen auf allen Knoten im Netzwerk synchronisiert sind und keine partielle Replizierung unterstützt wird. Zusätzlich wurde in diesem Arbeitspaket eine weitere Möglichkeit des Datenaustauschs in TrustDBle hinzugefügt, welche zuvor nicht im Projektvorhaben geplant war. Daten können dadurch nicht nur in gemeinsam genutzten Datenbanken basieren auf Blockchains geteilt werden, sondern auch direkt aus lokalen Datenbanken und Tabellen. Hierbei wird allerdings nur ein Lesezugriff gewährt, der mit Hilfe des Federated Storage-Engine-Plugins von MySQL umgesetzt wurde. Der Zugang zu lokalen Tabellen kann über denselben Mechanismus gegebenen werden, wie er für Tabellen in gemeinsam genutzten Datenbanken gegeben wird.

AP06
In diesem Arbeitspaket wurden Data Contracts konzipiert und anschließend durch den Einsatz von Trusted Execution Environments (TEEs) manipulationssicher und nachweisbar implementiert. Ein Data Contract beschreibt zum einen die Logik, die auf bestimmten Daten ausgeführt werden darf, zum anderen wer darauf zugreifen darf und welche Daten genutzt werden. Die Erstellung kann über einfache SQL-Befehle erfolgen und der Nutzer gibt an welche Logik auf welchen Daten wer ausführen darf.
In TrustDBle wurden Data Contracts durch eine zusätzliche Komponente in der Datenbankschicht realisiert. Die SQL-Befehle zur Erstellung wurden durch ein MySQL-Plugin implementiert, das in Arbeitspaket 7 hinzugefügt wurde, um den SQL-Standard von MySQL zu erweitern. Die Ausführung von Data Contracts in einer TEE wurde durch das Framework Gramine umgesetzt. Dieses Framework erlaubt die sichere und nachweisbare Ausführung von Python Code in SGX. Bei der Erstellung eines Data Contract gibt der Nutzer die Logik in Form von Python Skripten an, die dann durch Gramine in einer SGX Enclave ausgeführt werden. Zur Freigabe von Data Contracts für andere Nutzer wurde derselbe Mechanismus verwendet, wie für das Beitreten einer geteilten Datenbank. Durch eine individuelle Einladung erhält der Nutzer Zugriff auf den Data Contract und kann diesen dann aufrufen und gegebenenfalls Parameter mitgeben. Das Aufrufen eines Data Contracts wurde durch eine effiziente und leichtgewichtige gRPC-Kommunikation umgesetzt, welche eine Verbindung zum Eigentümer des Data Contracts aufbaut und dort die Ausführung mit optionalen Eingabeparameter in SGX startet. Das Ergebnis wird dann mit zusätzlichen Meta-Informationen zur Überprüfung der Korrektheit zurück an den Aufrufer geschickt. Dieser Ansatz ermöglicht es nicht nur auf gemeinsam geteilten Daten nachweisbare Data Contracts auszuführen, sondern auch auf lokalen Daten.

AP07
Erweiterung der SQL-Schnittstelle durch weitere SQL-Befehle zum Erstellen und Löschen von geteilten Datenbanken und Tabellen, Freigeben und Beitreten einer Datenbank und dessen Tabellen, Repartitionierung der Datenbank und Erstellen und Teilen von Data Contracts. Die Implementierung erfolgte durch die Verwendung eines Plugins für MySQL, welches eine SQL-Abfrage vor der Verarbeitung verändern bzw. umschreiben und zusätzliche Logik ausführen kann. Mit dem Plugin bietet sich der Vorteil eine existierende SQL-Schnittstelle, die viele standardisierte Protokolle wie ODBC, JDBC unterstützt, einfach zu erweitern und notwendige neue Befehle zu implementieren.
Die graphische Benutzeroberfläche wurde durch eine Web-Applikation umgesetzt, die es ermöglicht alle zuvor genannten Befehle per Klick auszuführen. Dem Nutzer werden alle geteilten Datenbanken und Tabellen angezeigt und er kann andere Nutzer einladen, um beizutreten. Die Monitoring Komponente für die Auditierung ist ebenfalls in der grafischen Benutzeroberfläche integriert und visualisiert die Änderungen der geteilten Datenbank. Durch die Implementierung eines Editors in der Benutzeroberfläche ist die Erstellung und Verwaltung von Data Contracts ohne großes technisches Wissen möglich.

Arbeitsbereich: Verschlüsselung & Zugriffskontrolle

AP08
Die Datenverschlüsselung wurde mit dem Framework OpenSSL implementiert und in einer Komponente gekapselt, welche relevante Methoden von OpenSSL integriert und kryptografische Funktionen bereitstellt. Dadurch stehen in allen Komponenten Funktion zur Datenverschlüsselung und notwenige kryptografische Verfahren auf die gleiche Weise zur Verfügung und lassen sich sehr einfach nutzen. In der Speicherebene können so die Daten ver- oder entschlüsselt werden, bevor sie in der Blockchain gespeichert oder ausgelesen werden.
Die Zugriffskontrolle wurde über einen hybriden schlüsselbasierten Austausch umgesetzt, welcher symmetrische als auch asymmetrische Verfahren zur Verschlüsselung nutzt. Teilnehmer können zu einer gemeinsam genutzten Datenbank eingeladen werden, indem sie notwendige individuell verschlüsselte Zugangsschlüssel erhalten. In der Datenbankebene wird hierfür die Verschlüsselungskomponente verwendet, um ein asymmetrisches Schlüsselpaar zu generieren und Zugangsschlüssel zu verschlüsseln und zu signieren. So kann nur der Besitzer des passenden privaten Schlüssels den Zugangsschlüssel entschlüsseln und verifizieren, wer der Sender ist.

AP09
Für die manipulationssichere Verwaltung und Speicherung von Metadaten über gemeinsam genutzte Daten ist eine Metachain, ein zusätzliches Blockchain-Netzwerk pro gemeinsam genutzte Datenbank, eingeführt worden.

Darin werden Daten wie Tabellennamen, geschützte Schlüssel und Information über Teilnehmer und Data Contracts gespeichert. Die Metadaten über gemeinsam genutzte Daten werden so unter allen Teilnehmer sicher und dezentral verwaltet. Jeder Nutzer ist im Besitz eines Schlüsselpaares, welches eine eindeutige Identifizierung sicherstellt. Mit Hilfe dieses Schlüsselpaares kann ein Nutzer Zugangsschlüssel zu einer gemeinsam genutzten Datenbank erhalten und dieser beitreten. Die Zugangsschlüssel werden in einer separaten verschlüsselten Tabelle in TrustDBle gespeichert und verwaltet und werden zur symmetrischen Verschlüsselung der Metadaten einer gemeinsam genutzten Datenbank verwendet. Innerhalb einer Datenbank wird jede geteilte Tabelle ebenfalls durch einen weiteren Schlüssel geschützt.

Arbeitsbereich: Planung & Validierung

AP10
Im AP10 wurden die Zielarchitektur und Schnittstellen zwischen den Komponenten definiert. Hierbei wurde das Open Source Datenbankmanagementsystem MySQL als Ausgangsarchitektur genutzt und durch Plugins erweitert. Die zwei-schichtige Architektur ergibt sich somit aus einen Storage-Engine-Plugin, welches die Speicherung von Daten in einer Blockchain ermöglicht und dem Datenbankmanagementsystem, welches zusätzlich durch ein Rewrite-Plugin neue Funktionen bereitstellt. Die passende Integration in das bestehende Datenbankmanagementsystem wurde über den Plugin-Mechanismus des MySQL Systems realisiert. Das neue Storage-Engine-Plugin nutzt intern die Blockchain-Abstraktion, die in AP1 definiert wurde und deren Adapter Implementierungen, um Daten, die in MySQL in einer Datenbank abgelegt werden, in einer beliebigen Blockchain zu speichern. Das Rewrite-Plugin wird zum Erweitern der SQL-Schnittstelle genutzt, womit neue Befehle dem SQL-Standard hinzugefügt werden können. Zur Einbindung von SGX und der Kommunikation mit Prozessen in SGX wurde eine User Defined Function in MySQL erstellt, welche die Ausführung von Logik in der SGX ermöglicht. Über eine gRPC-Schnittstelle werden die Daten zur Ausführung in der SGX übertragen. Die Implementierung dieser Schnittstelle wurde in AP6 durchgeführt. In Abbildung 1 werden beide Schichten mit den jeweiligen Komponenten dargestellt.

AP11
Zur Versionierung des Programmcodes während der Entwicklung wurde der Dienst Bitbucket genutzt. Ebenfalls wurde darüber die DevOps Infrastruktur für Entwickler verwaltet und CI/CD Pipelines zur automatischen Ausführung von Tests und automatisierten Bereitstellung der Anwendung umgesetzt. Die skalierbare Ausführung des Systems für Integrationstests und Demonstrationszwecke wurde durch mehrere virtuellen Maschinen von Microsoft Azure realisiert. Hierfür wurden sogenannte Runner in Bitbucket erstellt und in die Pipelines integriert. Bei der Überführung von Änderung am Programmcode in die finale Anwendung werden die CI/CD-Pipelines gestartet und testen die gesamte Anwendung und stellen diese dann als Container Images bereit. Einfach zu bedienende Skripte erleichtern die Kompilierung und das Starten des Systems. Entwickler sind so in der Lage Änderung am Programmcode effizient und schnell lokal zu testen. Für die lokale Bereitstellung von TRUSTDBLE werden Container und das Container-Orchestrierungstool Docker Compose verwendet. Anwender können so das System schnell und einfach ausführen und nutzen.

AP12
Die Entwicklung von System- und Integrationstests wurde mit Hilfe von verschiedenen Frameworks durchgeführt. Für die Überprüfung von Modulen und Komponenten wurde das Testing Framwork google test verwendet, welches ein einfaches und weniger komplexes Testen ermöglicht. In den jeweiligen Arbeitspaketen wurden damit die Systemtests implementiert. Die Integrationstests zur Überprüfung des Zusammenspiels aller Systemkomponenten wie Backend, Rewrite-Plugin und Storage-Engine-Plugin wurde mit dem MySQL Test Framework und Jest+Supertest umgesetzt und größtenteils in diesem Arbeitspaket entwickelt. Mit dem MySQL Test Framework wurden alle neu hinzugefügten Befehle zum SQL-Standard überprüft. Eine Testinstanz wurde mit Testdaten aufgesetzt und mehrere Testfälle für ein neuen Befehl ausgeführt. Ein Testfall besteht aus mehreren SQL-Befehlen und einem zu erwartenden Ergebnis. Nach der Ausführung der Befehle wird das erhaltene Ergebnis mit dem zu erwartenden Ergebnis verglichen. Mit Jest+Supertest wurde das Backend des Systems getestet, auf welchem die Benutzerschnittstelle aufbaut. In diesem Fall wurde eine Testinstanz mit Backend aufgesetzt und einzelne API-Endpunkte pro Testfall mit den jeweiligen Parametern aufgerufen und das Ergebnis mit einem zu erwartenden Ergebnis verglichen. Die Ausführung der Tests wurde sowohl lokal durch Skripte ermöglicht als auch zur automatischen Überprüfung des Programmcodes in unsere CI-Pipeline integriert.

AP13
Bei der Umsetzung eines Anwendungsfall wurde das System durch eine einfache und schnelle containerbasierte Installation bereitgestellt. Da während der Projektlaufzeit unter anderem mit Unternehmen im Gesundheitsbereich gesprochen wurde und hier auch ein großer potentieller Markt zur Anwendung der Lösung besteht, ist der Anwendungsfall nicht am Beispiel von IoT implementiert, sondern nutzt beispielhafte Gesundheitsdaten, die unter mehreren Krankenhäusern sicher und nachvollziehbar gemeinsam genutzt werden. Dadurch kann demonstriert werden, wie Krankenhäuser Patientendaten untereinander vertrauenswürdig teilen können.

AP14
Zur Projektkoordination und Abstimmung mit Projektpartnern nutzte das Projektteam überwiegend zwei Softwareprogramme. Zum einen Microsoft 365, welches MS Teams beinhaltet, zur internen und externen Kommunikation. Zum anderen Jira eine Projektmanagement-Software, womit die verschiedenen Aufgaben im Team organisiert, koordiniert und dokumentiert wurden. Hierbei wurde die Scrum-Methode angewendet, die eine agile und effiziente Softwareentwicklung ermöglicht.

AP15
Das Projekt wurde erfolgreich auf mehren Konferenzen vorgestellt und es wurden mehrere Veröffentlichungen publiziert. Zum Austausch mit Experten und zur Projektkommunikation wurden ebenfalls verschiedene Industrie- und Fachmessen besucht. Der Programmcode wurde am Ende der Projektlaufzeit auf GitHub unter der MIT Lizenz veröffentlicht. (https://github.com/DataManagementLab/trustdble)

Die wichtigsten Positionen des zahlenmäßigen Nachweises
Die wichtigste und größte Position im zahlenmäßigen Nachweis sind die Personalkosten für insgesamt 5 wissenschaftliche Mitarbeiter, welche während der Projektlaufzeit an dem Projekt gearbeitet haben. Des Weiteren wurden zur Unterstützung in der Entwicklung und bei Forschungsarbeiten studentische Hilfskräfte angestellt, deren Lohn die zweit größte Ausgabenposition bildet und unter der Position 822 im zahlenmäßigen Nachweis gebucht ist. Gegenstände über 410€ darunter fällt die Hardware für die wissenschaftlichen Mitarbeiter zur Durchführung der Projektarbeiten sowie einen leistungsstarken Server mit neuster Hardware und Unterstützung von SGX zur Ausführung von Leistungstests machen die dritte große Position aus. Weitere Positionen sind Kosten für Gegenstände bis 410€ wie Monitore und Peripherie, Miete und Rechnerkosten für virtuelle Maschinen in Azure zur Überprüfung der Skalierbarkeit, Ausgaben für Aufträge von Weiterbildungen und Workshops, Reise- und Veranstaltungskosten von Besuchen der Messen und Konferenzen sowie allgemeine Verwaltungskosten für Softwarelizenzen und Webhosting.

Notwendigkeit und Angemessenheit der geleisteten Arbeit
Die Projektmittel ermöglichten die Durchführung des Projekts und die angestellten wissenschaftlichen Mitarbeiter konnten die geplanten Arbeiten durchführen und die Projektziele unter der Berücksichtigung der kostenneutralen Verlängerung erfüllen.

Projektveröffentlichungen und -kommunikation
Während der Projektlaufzeit wurden nachfolgende Publikationen veröffentlicht und folgende Messen, Konferenzen sowie Netzwerktreffen zur notwendigen Projektkommunikation und zum Austausch mit Experten besucht. Die Besuche waren sehr hilfreich und die Gespräche und Rückmeldungen zum Projektvorhaben nutze das Team zum Nachjustieren und Anpassen des Vorgehens der Projektarbeiten.

Publikationen
Folgende vier Publikationen wurden auf Konferenzen und Workshops veröffentlicht und vorgestellt:

- “ACID-V: Towards a New Class of DBMSs for Data Sharing“ wurde im Januar 2022 auf Springer Link veröffentlicht. In der Publikation wird beschrieben, wie die Konsistenz-Garantien von klassischen Datenbanken um Verfizierbarkeitsgarantien erweitert werden können, um die Anforderungen von Data Sharing in Datenbankmanagementsystemen besser zu adressieren.
- “Benchmarking the Second Generation of Intel SGX Hardware” wurde im Juni 2022 auf ACM Open Access veröffentlicht und beim DAMON Workshop während der SIGMOD 2022 Konferenz in Philadelphia

(USA) vorgestellt. Die Publikation untersucht die neue Implementierung der Intel Software Guard Extension (SGXv2) und vergleicht diese zu der Vorgängerversion SGXv1, welche gewisse technische Limitationen aufweist.
- "Towards a Benchmark for Shared Databases" wurde im Dezember 2022 als Vision Paper auf Springer Link veröffentlicht. Diese Publikation stellt eine Vision und erste Ideen für einen neuen Benchmark vor, um geteilte Datenbanken zu bewerten und ihre einzigartigen Merkmale zu erfassen.
- "TRUSTDBLE: Towards a New Class of DBMSs for Data Sharing" wurde als Research Report in der Ausgabe Januar 2022 von efl insights veröffentlicht. In dieser Publikation wird vorgestellt, wie TRUSTDBLE die ACID-Eigenschaften von klassischen Datenbanken durch eine neue Eigenschaft der Verifizierbarkeit erweitert, welche für geteilte Datenbank Systeme notwendig ist.

Konferenzen
Für die Projektkommunikation und den Austausch mit Experten wurden folgende fachliche Konferenzen besucht:
- Auf der VLDB in Kopenhagen wurde sich mit Experten zu Themen des skalierbaren Datenmanagement sowie zu Big Data ausgetauscht.
- Die SIGMOD 2022 in Philadelphia bot die Möglichkeit mit führenden Wissenschaftlern und Experten im Bereich Data Management zu sprechen und Rückmeldungen zum wissenschaftlichen Stand des Vorhabens zu erhalten.

Messen
Nachfolgende Industire- und Fachmessen sowie Veranstaltungen und ein Netztreffen wurden besucht:
- Die SecIt in Hannover brachte Einblicke in die aktuellen Sicherheitstrends und es wurde mit Experten der IT- Sicherheit gesprochen und Einschätzung zum Projekt und dessen Ziel eingeholt.
- Auf der BlockchainExpo in Amsterdam wurden im Bereich Blockchain und dezentralen Systemen neuste Entwicklungen angeschaut und es wurde sich mit anderen Unternehmen und Gründern vernetzt.
- Auf der Hannover Messe wurde überwiegend der IT- und Softwarebereich besucht, um sich dort mit Industrieunternehmen auszutauschen und einen Einblick auf mögliche Anwendungsfälle und existierende Lösung zu erhalten.
- Die it-sa in Nürnberg ist eine der größten IT-Security Messen und bot die Möglichkeit sich mit Unternehmen und Experten aus der Branche auszutauschen und potentielle Anwendungsfälle zu diskutieren. Zudem wurde wertvolles Feedback zum Ziel des Projektvorhabens und dessen technischer Umsetzung gesammelt.
- In Berlin fand ein Netzwerktreffen von Amplify Partners statt, welches zum Netzwerken und Austauschen im Bereich Softwareentwicklung und Ausgründung genutzt wurde.
- Beim Startup & Innovation Day und FoundersXchange Capital Day in Darmstadt wurde das Projekt vorgestellt und auf einem eigenen Stand dessen Lösung mittels eines Demonstrators präsentiert.
- Der Besuch des DigitalFutureCongress in Frankfurt wurde genutzt, um zu Netzwerken und sich mit anderen auszutauschen.
- An den online Veranstaltungen HubNight Cybersecurity und Cybersecurity in Finance wurde teilgenommen und das Projekt vorgestellt.

Verwertung
Zur weiten Verwendung der Lösung wurde der Programmcode des Projekts auf GitHub veröffentlich und jedem zugänglich gemacht. Das Repository ist unter https://github.com/DataManagementLab/trustdble verfügbar. Die Ergebnisse des Projekts werden zum dem in einem Folgeprojekt, welches durch die 2. Phase von StartUpSecure gefördert wird, genutzt und die entstandene Lösung weiterentwickelt, um eine sichere und einfach zu verwendende Datenplattform für Data Sharing aufzubauen. Zum anderen sollen darauf Anwendungsfälle in unterschiedlichen Branchen umgesetzt werden. Die Projektverantwortlichen sind in Gesprächen mit potentiellen Partnern, die Interesse an der Implementierung eines spezifischen Anwendungsfall mit der Lösung haben.