

Pflichtenheft

SECURE DATA OUTSOURCING

 MICHAEL FECHNER, MAIK WILD, FABIAN SCHEYTT, JULIAN HINRICHS

Inhalt

Einleitung	2
Zielbestimmung.....	3
Musskriterien	3
Wunschkriterien.....	4
Abgrenzungskriterien	4
Produkteinsatz	6
Anwendungsbereiche	6
Zielgruppen	6
Betriebsbedingungen	6
Produktumgebung	7
Softwareanforderungen.....	7
Hardwareanforderungen	7
Anwendungsfälle	8
Musskriterien	8
Wunschkriterien.....	9
Funktionale Anforderungen	12
Clientseitige Musskriterien	12
Serverseitige Musskriterien	13
Clientseitige Wunschkriterien	14
Serverseitige Wunschkriterien	15
Nichtfunktionale Anforderungen	16
Systemmodell.....	17
Produktdaten	18
Clientseitig.....	18
Serverseitig.....	18
Qualitätsziele	19
Testfälle	20
Testfälle für Musskriterien	20
Testfälle für Wunschkriterien	20
Grafische Benutzeroberfläche.....	21
Entwicklungsumgebung.....	23
Glossar	24

Einleitung

Das Ziel des Secure Data Outsourcing Projektes, ist die sichere Ablage von **wichtigen** Daten in einer von einem externen Anbieter bereitgestellten Datenbank. Dazu sollen zwei Anwendungen entwickelt werden, die nur im internen Privat-, Firmen- oder Forschungsnetzwerk betrieben werden. Die Server Anwendung ist ein SQL-fähiger Dienst, der in Interaktion mit einem schon bestehenden **Encryption** Service als sicherer Mediator zu Cloudanbietern wie Amazon, Google, etc. agiert und so die Rolle eines sicheren SQL-Servers übernimmt. Zur Verwaltung bietet der Mediator Dienst für Administratoren eine Schnittstelle an, mit der Einstellungen **bearbeitet** und der Dienst überwacht werden kann.

Außerdem soll clientseitig eine einfach zu bedienende Anwendung mit grafischer Benutzeroberfläche erstellt werden, die über diesen Mediator auf verschlüsselte Datenbanken im Cloudspeicher zugreifen, diese lesen und verändern kann. Das Ziel hierbei ist es, dass der Cloudanbieter weder Zugriff auf diese Daten hat, noch auf sie zurückschließen kann. Damit können auch sicherheitsrelevante Daten auf einer externen Datenbank abgelegt werden, ohne Sicherheitsrisiken bei der Übertragung oder beim Anbieter zu erlauben.

Bei jedem Zugriff auf die Datenbank sollen sowohl die clientseitige als auch die serverseitige Anwendung Zeitmessungen vornehmen, um Daten bezüglich der Effizienz des verwendeten Verschlüsselungsmechanismus zu gewinnen.



Abbildung 1: Verschlüsselte Datenbank¹

¹ <https://crmbusiness.wordpress.com/2015/10/27/crm-20152013-all-you-need-to-know-about-database-encryption/>

Zielbestimmung

In diesem Kapitel wird definiert, welche Funktionen das Produkt bieten soll.

Dafür werden Funktionen, die das Produkt auf jeden Fall implementiert sowie Wunschfeatures, die zusätzlich implementiert werden können, allerdings optional sind, aufgelistet.

Wunschfeatures sind Features, die zwar sinnvoll sind, aber aufgrund von Zeit oder Geldmangel nicht implementiert werden müssen.

Musskriterien

Serverseitig

Grundsätzliche SQL Capabilities

INSERT verstehen und ausführen

Durch Eingabe des SQL-Befehls „INSERT“  en in die Datenbank einspeisen

SELECT verstehen und ausführen

Durch Eingabe des SQL-Befehls „SELECT“ Daten aus der Datenbank anzeigen

UPDATE verstehen und ausführen

Durch Eingabe des SQL-Befehls „UPDATE“ Daten in der Datenbank manipulieren

DELETE verstehen und ausführen

Durch Eingabe des SQL-Befehls „DELETE“ Daten aus der Datenbank löschen

JOIN verstehen und ausführen

Durch Eingabe des SQL-Befehls „JOIN“ zwei Tabellen der Datenbank zusammenfügen und anzeigen

Zeitmessungsfunktionalität

Alle Zugriffe werden aufgezeichnet und deren Ausführzeit in Logs protokolliert

Die Logs können von einem Administrator ausgelesen werden

Capabilities anzeigen

Durch eine direkte Verbindung zum Server können alle SQL-Befehle aufgelistet werden, die der Server unterstützt

Administrator Schnittstelle

Alle anfallenden Logs auslesen

Standard Credentials der Datenbank ändern

Clientseitig

Benutzeroberfläche

Tooltips anzeigen, wenn der Mauszeiger über ein Element gehalten wird

Eine Benutzerhilfe bildet eine Ansicht, in der die Befehle und deren Vorbedingungen erklärt werden

Zeitmessungsfunktionalität

Alle Zugriffe werden aufgezeichnet und deren Ausführzeit in Logs protokolliert

Anzeigen der Zugriffszeiten während Programmlaufzeit

Die Logs können vom Benutzer ausgelesen werden

SQL Statement Generierung

Option, beliebige SQL Statements, die der Server unterstützt, mit der GUI zu generieren

Syntaxkontrolle

Warnen, wenn eingegebener SQL Befehl ungültig ist

Warnen, wenn eingegebener SQL Befehl nicht unterstützt wird

Wunschkriterien

Serverseitig

Eine Auswahlmöglichkeit an Verschlüsselungsalgorithmen bereitstellen

Client – Mediator Kommunikation verschlüsseln

Jeglichen Datenverkehr im sicheren lokalen Netz verschlüsseln

Erweiterung der SQL Capabilities

Implementierung zusätzlicher SQL Befehle

User Credentials

Implementierung von User Credentials. Es können unterschiedliche Datenbankuser mit unterschiedlicher Rechte auf die Datenbank zugreifen

Clientseitig

Auswahl der Verschlüsselungsalgorithmen

Cloud Tab

Die Dateien in einer Virtuelle Ordnerstruktur anzeigen

Die Ordnerstruktur navigieren können

Einfaches Abspeichern von Dateien per Drag-And-Drop oder Dateiauswähler

Einfaches Löschen von Dateien per „entf“-Taste oder Kontextmenü

Einfaches Herunterladen von Dateien per Kontextmenü

Import-/Exportfunktionalität

Einlesen von bestehenden Datenbanken und sukzessives Aufbauen dieser auf dem Cloudserver

Auslesen der verschlüsselten Clouddatenbank und abspeichern dieser in einer SQL Datei

Credentials

Auswahlfunktion und Speichermöglichkeit von


Cloudserver Hostname

Cloudserver Benutzername

Cloudserver Benutzerpasswort

Abgrenzungskriterien

Jegliche Programmability

Aufgrund der Verschlüsselung der Datensätze ist eine Ausführung jeglicher  **serverseitiger** Skripte nicht möglich

Direkter Zugriff auf die Clouddatenbank des Clients

Die Verschlüsselung ist auf dem Mediator implementiert, daher ist jeglicher Zugriff des Clients auf die Clouddatenbank untersagt

Jeglicher unverschlüsselte Datenverkehr zwischen Server und Clouddatenbank

Dies würde die Prämisse des Produktes verletzen, da Daten nicht nur vom Anbieter, sondern allen, die bei der Datenübertragung beteiligt sind, ausgelesen werden können.

Lokale Ablage der Datenbank

Die Datenbank wird auf einem externen Cloud Server geführt

Produkteinsatz

Hier werden der Zweck des Produktes sowie seine Einsatzmöglichkeiten erläutert.

Das Produkt dient zusammen mit dem schon bestehenden Encrypt Service als Mediator zwischen Cloudspeicher und Benutzer und bietet clientseitig eine grafische Benutzeroberfläche. Damit stellt es eine einfache Möglichkeit zur Verfügung, Datenbanken - und eventuell auch Dateien - verschlüsselt bei Cloud-Diensten auszulagern.

Anwendungsbereiche

Privater Anwendungsbereich

Für den privaten Gebrauch ist hauptsächlich das Wunschfeature „Cloud Tab“ relevant. Jedoch kann auch hier die verschlüsselte Auslagerung von Datenbanken mit persönlichen Daten von Vorteil sein.

Kommerzieller/Administrativer Anwendungsbereich

Im kommerziellen und/oder administrativen Anwendungsbereich kann die Software für alles von Patienten- über Mitarbeiter- bis hin zu Produktdaten Verwendung finden.

Forschung

Auch in der Forschung können mit dem Produkt Mitarbeiterdaten sowie Forschungsergebnisse einfach und sicher ausgelagert werden.

Zielgruppen

Zielgruppen sind hier sowohl Privatpersonen als auch Unternehmen in praktisch allen Branchen sowie staatliche Einrichtungen. Vorrangig richtet sich das Produkt an Nutzer, die Database-as-a-Service gegenüber skeptisch sind und/oder Cloudanbietern nicht **vollständig vertrauen**, ihre Daten aber trotzdem gerne auslagern würden.

Betriebsbedingungen

Das Produkt kann sowohl in Büroumgebung als auch in Heimanwendung benutzt werden, unter der Voraussetzung, dass sich sowohl das Clientsystem als auch der Server, auf dem die Connect und Encrypt Services installiert sind, in dem selben gesicherten privaten oder Firmennetzwerk befinden.

Produktumgebung

Mit der Produktumgebung werden sowohl Software als auch Hardware Komponenten spezifiziert, die benötigt werden um das Produkt zu betreiben. Die hier aufgeführten Anforderungen sind die minimalen Voraussetzungen für eine korrekte Funktion des Produktes. Auf performanteren und aktuellen Systemen kann in der Regel auch eine Leistungssteigerung erkannt werden.

Softwareanforderungen

Serverseitig

Der Server auf dem der Mediator Dienst ausgeführt wird sollte mindestens ein Windows **Server 2003 Betriebssystem** mit installiertem .NET Framework 4 oder höher besitzen. Dadurch wird die korrekte Ausführung der C# Anwendung gewährleistet.

Clientseitig

Auf dem Client System sollte mindestens ein Windows 7 Betriebssystem oder eine aktuellere Version mit 32 oder 64 Bit installiert sein.

Außerdem wird für die Ausführung der Client Anwendung eine Installation des .NET Framework 4 oder eine aktuellere Version vorausgesetzt.

Hardwareanforderungen

Serverseitig

Für die ausgelegte Benutzung wird mindestens ein System mit Zweikern-Prozessor und 8 GB RAM benötigt, sowie Zugriff auf das lokale Netzwerk und die Cloud-Storage-Provider auf denen **Datenbanken** abgelegt sind.

Außerdem sollte mindestens 30 Megabyte Festplattenspeicher verfügbar sein um ausreichend Platz für die Server Anwendung und anfallende Log Dateien anzubieten.

Clientseitig

Für das Clientsystem genügen die minimalen Anforderungen des jeweiligen Betriebssystems, ebenfalls für die Client Anwendung.

Also sind für einen Windows 7 32-Bit Clientsystem ein 1 GHz Prozessor mit 1 GB RAM sowie **16 GB Festplattenspeicher** ausreichend um die Client Anwendung auszuführen.

Zusätzlich zu diesen Anforderungen benötigt der Client eine Verbindung ins lokale Netzwerk um mit dem Mediator kommunizieren zu können. Eine Internetverbindung ist nicht zwingend erforderlich.

Anwendungsfälle

Hier werden alle möglichen Szenarien gebündelt, welche auftreten können, wenn ein Akteur versucht ein Ziel (vergleiche Funktionale Anforderungen) zu erreichen. Die Anwendungsfälle unterteilen sich wieder zwischen Muss- und Wunsch Anwendungsfälle.

Es gibt nur zwei agieren Akteure. Benutzer und Administratoren. Ein Administrator ist Benutzer mit Zugriffsrechten auf den Server, ein Administrator *erweitert* also einen Benutzer. Das heißt jeder Administrator ist ein Benutzer, allerdings nicht jeder Benutzer ein Administrator. Alle Dinge, die ein Benutzer darf, darf ein Administrator auch.

Musskriterien

Benutzer führt Insert-Befehl aus

Ein Benutzer kann durch Eingabe eines syntaktisch korrekten SQL „INSERT“ Statements in der GUI Datensätze in die Datenbank einspeisen

Benutzer führt Select-Befehl aus

Ein Benutzer kann durch Eingabe eines syntaktisch korrekten SQL „SELECT“ Statements in der GUI Datensätze aus Datenbank auslesen

Benutzer führt Update-Befehl aus

Ein Benutzer kann durch Eingabe eines syntaktisch korrekten SQL „UPDATE“ Statements in der GUI Datensätze in die Datenbank manipulieren

Benutzer führt Delete-Befehl aus

Ein Benutzer kann durch Eingabe eines syntaktisch korrekten SQL „DELETE“ Statements in der GUI Datensätze aus der Datenbank löschen

Benutzer führt Join-Befehl aus

Ein Benutzer kann durch Eingabe eines syntaktisch korrekten SQL „JOIN“ Statements in der GUI Tabellen der Datenbank zusammenfügen und die resultierende Tabelle einsehen

Benutzer generiert SQL Statement

Ein Benutzer hat die Option alle möglichen SQL Statements, die sich aus den genannten Befehlen erzeugen lassen, automatisch über die GUI zu generieren

Benutzer ändert die Tabellenansicht

Ein Benutzer kann die durch „SELECT“ angezeigten Tabellen nach beliebigen Spalten sortieren können

Benutzer sieht Ergebnisse

Ein Benutzer sieht die Ergebnisse einer Ausführung eines SQL-Befehls auf dem Bildschirm

Benutzer gibt syntaktisch inkorrekten SQL Befehl ein

Da ein Benutzer SQL-Befehle frei eingeben kann, können auch syntaktisch inkorrekte SQL-Befehle eingegeben werden. In diesem Fall tritt der Anwendungsfall *Benutzer sieht Syntaxerror* ein

Benutzer gibt nicht unterstützten SQL-Befehl ein

Da ein Benutzer SQL-Befehle frei eingeben kann, können nicht unterstützte SQL-Befehle eingegeben werden. Da der Server diese per Definition nicht versteht tritt in diesem Fall der Anwendungsfall *Benutzer sieht Syntaxerror* ein

Benutzer sieht Syntaxerror

Bei ungültigen SQL-Befehlen wird auf dem Bildschirm eine Fehlermeldung angezeigt und benachrichtigt den Benutzer über den Fehler

Benutzer liest Ausführzeiten aus (Client)

Ein Benutzer kann die vom Client gespeicherte Ausführzeiten auslesen

Administrator liest Ausführzeiten aus (Server)

Ein Administrator kann die vom Server gespeicherten Ausführzeiten auslesen

Administrator liest Error Logs aus

Ein Administrator kann Fehler, die bei vergangenen Anfragen aufgetreten sind auslesen

Administrator ändert Default Credentials

Ein Administrator kann die Standard Credentials für die Clouddatenbank ändern

Administrator liest SQL Capabilities aus

Ein Administrator kann die SQL Capabilities des Servers auslesen

Wunschkriterien

Benutzer wählt Verschlüsselungsalgorithmus

Ein Benutzer soll auswählen können, welche Verschlüsselungsalgorithmen zur Verschlüsselung der Datensätze verwendet wird.

Benutzer ändert User Credentials

Ein Benutzer soll ändern können, auf welchem Cloudserver er arbeiten möchte, mit welchem Cloudserver Benutzernamen er sich anmeldet und mit welchem Cloudserver Benutzerpasswort er sich authentifiziert. Die Änderung kann durch Auswahl gespeicherter Einträge passieren oder manuelle Eingabe

Benutzer speichert User Credentials

Ein Benutzer soll den Hostnamen des Cloudservers, sein Cloudserver Benutzernamen und sein Benutzerpasswort speichern können

Cloud Tab Anwendungsfälle

Benutzer speichert eine Datei

Ein Benutzer soll eine Datei in einem separaten Cloud Tab hochladen können

Benutzer sieht vorhandene Dateien

Ein Benutzer soll eine zuvor durch das Cloud Tab hochgeladenen Dateien in einer Ordnerstruktur einsehen können

Benutzer lädt Datei herunter

Ein Benutzer soll eine zuvor durch das Cloud Tab hochgeladenen Dateien in einer Ordnerstruktur herunterladen können

Benutzer navigiert durch Ordnerstruktur

Ein Benutzer soll die virtuelle Ordnerstruktur navigieren können

Benutzer ändert die Ordneransicht

Ein Benutzer soll die angezeigten Dateien nach bestimmten Spalten sortieren können (Dateityp, Änderungsdatum etc.)

Benutzer löscht Datei

Ein Benutzer soll Dateien aus der Ordnerstruktur löschen können

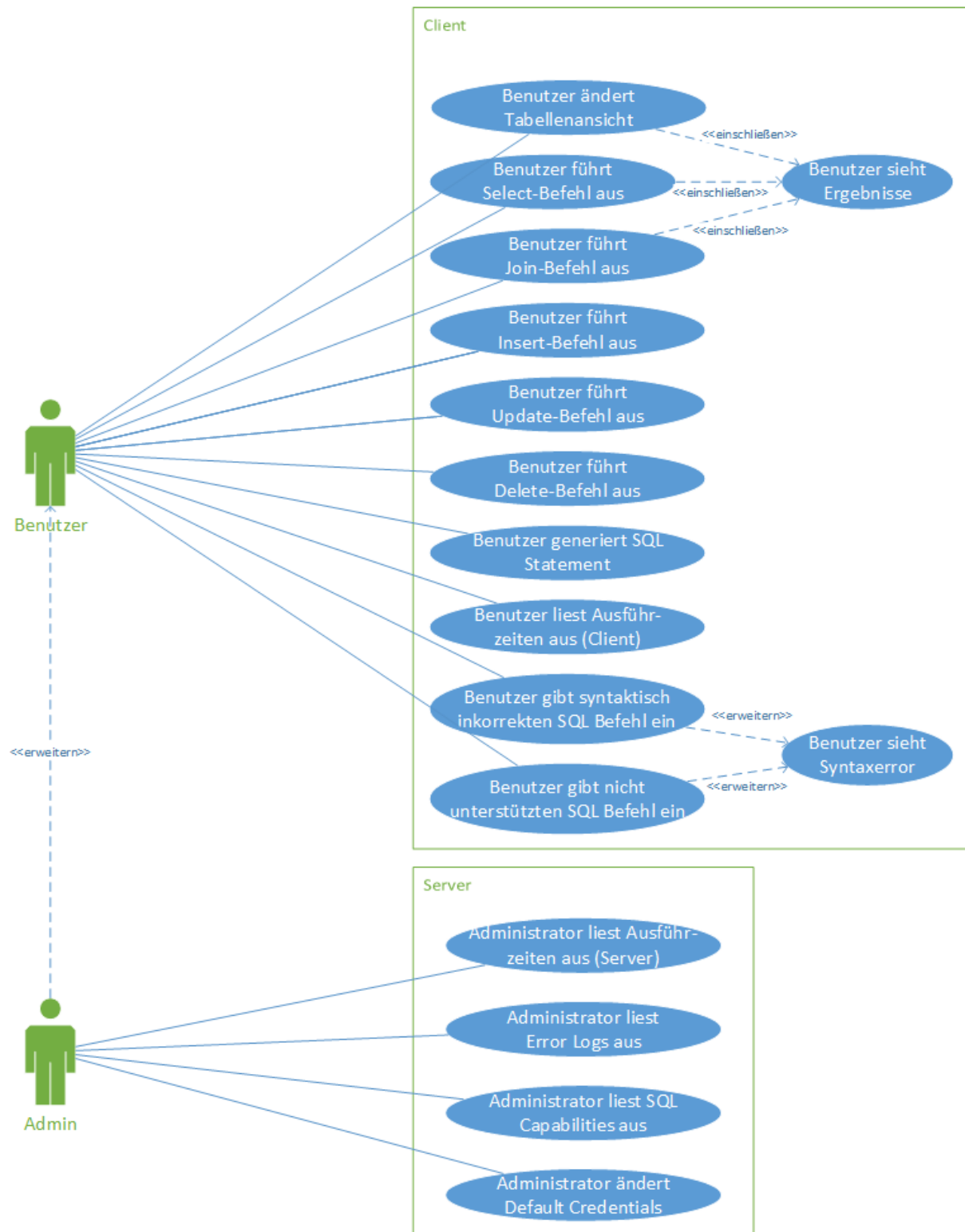


Abbildung 2: Anwendungsfälle der Musskriterien

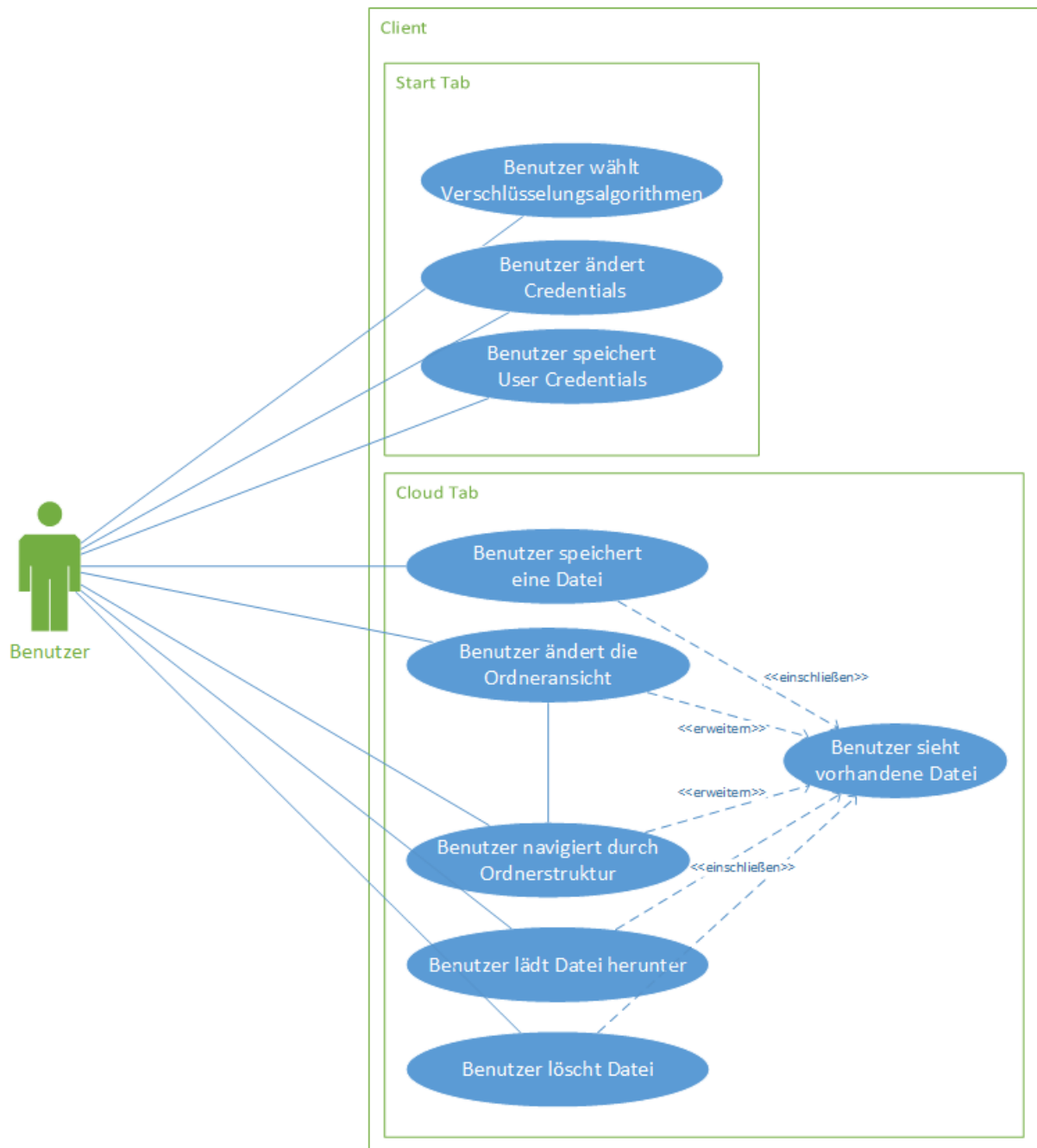


Abbildung 3: Anwendungsfälle der Wunschkriterien

Funktionale Anforderungen

Clientseitige Musskriterien

- /F005/ Client Anwendung öffnet GUI bei Startvorgang
Startet der Benutzer die Client Anwendung, so öffnet sich die Grafische Benutzerschnittstelle der Anwendung auf dem Computer des Benutzers.
- /F010/ Client kann eine TCP Verbindung zum Mediator herstellen
Die Client Anwendung errichtet nach dem Starten eine TCP Verbindung zum Connect Service, der Teil des Mediator Dienstes ist.
- /F015/ Client Anwendung besitzt Eingabefeld für SQL Befehle
Die Client Anwendung besitzt auf der GUI Oberfläche ein Eingabefeld für die Eingabe von SQL Befehlen. Automatisch generierte SQL Befehle werden ebenfalls in diesem Feld angezeigt.
Unterstützt werden INSERT, SELECT, UPDATE, DELETE und JOIN Befehle.
- /F020/ Client Anwendung speichert gesendete SQL Anfragen in Logdatei
Die Client Anwendung vermerkt jede vom Benutzer abgeschickte SQL Anfrage, sowie die zugehörige gemessene Latenzzeit in einer Logdatei auf dem lokalen System.
- /F025/ Client Anwendung zeigt Ergebnisse in Tabelle an
Ergebnisse, einer SQL Anfrage, die der Mediator zurückliefert werden in einer Tabelle auf der GUI Oberfläche der Client Anwendung visualisiert und die jeweiligen Spalten mit den entsprechenden Spaltennamen in der Kopfzeile betitelt.
- /F030/ SQL Anfragen können aus Tabellenzelle über das Kontextmenü generiert werden
Werden in der Ergebnistabelle der Client Anwendung die Resultate einer erfolgreichen Datenabfrage angezeigt, so kann über einen rechtsklick auf eine entsprechende Zelle der Tabelle das Kontextmenü für die entsprechende Zelle aufgerufen werden.
Nun kann zwischen verschiedenen SQL Anfragen gewählt werden, die sich auf diesen speziellen Datensatz beziehen. Möglichkeiten wären zum Beispiel das Bearbeiten einer Zelle, das Filtern nach bestimmten Wertebereichen einer Spalte oder das Einfügen eines ähnlichen Datensatzes.
Die generierten SQL Anfragen werden dann im Eingabefeld angezeigt und können vom Benutzer ausgeführt werden.
- /F035/ Client sendet SQL Befehle an Connect Service
Von der Client Anwendung aus können, nach erfolgreicher TCP Verbindung zum Server, eingegebene SQL Befehle an den Connect Service des Mediators gesendet werden
- /F040/ IP-Adresse des Servers in der Client Anwendung konfigurierbar
Die IP-Adresse, zu der der Client beim Starten eine TCP Verbindung aufbaut, kann in der UI über ein Eingabefeld konfiguriert werden.
- /F045/ SQL Befehle über UI generierbar
Mit einer UI Maske können in der Client Anwendung SQL Befehle erstellt werden, um zum Beispiel auch unerfahrenen Benutzern die Datenabfrage zu erleichtern und die SQL Befehlseingabe zu vereinfachen.
- /F050/ Zeitmessung am Client für Datenbankabfragen
Die Client Anwendung misst die verstrichene Zeit vom Versenden einer SQL Anfrage bis zum Erhalt einer Antwort vom Mediator Dienst und zeigt diese mit den Ergebnissen auf der GUI an.

Die Latenzzeit wird dann zusätzlich im Aktivitätslog mit der entsprechenden SQL Anfrage vermerkt.

/F055/ Benutzerhilfe in der Client Anwendung als Pop-Up enthalten

Die Benutzerhilfe kann in der GUI der Client Anwendung über den entsprechend gekennzeichneten Button im Menü gestartet werden und öffnet ein Pop-Up.

/F060/ Benutzerhilfe erläutert SQL Abfragen und Secure Data Outsourcing

In der Benutzerhilfe finden sich nützliche Informationen für die Anwendung der Client Software. Es wird vor allem das Prinzip des Secure Data Outsourcing erklärt und wie die Verbindung über den Mediator zur Cloud Database aufgebaut werden kann.

Außerdem werden in der Benutzerhilfe alle Funktionen des Produktes ausführlich und verständlich erklärt und in Zusammenhang gebracht, sodass das Produkt auch ohne Vorkenntnisse zu bedienen ist.

/F065/ Client Anwendung zeigt Tooltips für alle Funktionen an

Wird in der Oberfläche der Client Anwendung der Mauszeiger für kurze Zeit über eine Funktion, also einen Button oder eine Einstellungsmöglichkeit, gehalten so wird ein Tooltip für die jeweilige Funktion eingeblendet. Dieses beschreibt für den Benutzer die Auswirkung, die diese Funktion oder Einstellung hat und liefert Informationen und Tipps zur Anwendung.

/F070/ Client Anwendung erkennt falsche SQL Anfragen

Bevor eine SQL Eingabe an den Server gesendet wird, wird geprüft ob die Eingabe einen korrekten SQL Befehl darstellt. Ist dies nicht der Fall, so wird dem Benutzer eine Fehlermeldung angezeigt und die fehlerbehaftete Eingabe nicht an den Server gesendet.

Serverseitige Musskriterien

/F080/ Connect Service sendet Capabilities bei Verbindungsaufbau

Möchte ein Client eine Verbindung mit dem Connect Service herstellen, so antwortet der dieser zunächst mit einer Liste von Befehlen die der Mediator ausführen kann (Capabilities). Dies geschieht um dem Client eine Übersicht aller verfügbaren Befehle zu ermöglichen und Fehler bei der Kommunikation zu verhindern.

/F085/ Mediator stellt über den  encryption Service eine Verbindung mit Cloud Datenbank her

Beim Starten des Mediator Dienstes auf dem Server, versucht der Connect Service zuerst über die Schnittstelle des Encryption Service eine Verbindung zur Cloud Datenbank herzustellen. Die Verbindungsdaten der Datenbank sind in einer Konfigurationsdatei auf dem Server gespeichert.

/F090/ Connect Service kann SQL Befehle interpretieren

Der Connect Service des Mediators kann von einem Client eingehende SQL Anfragen interpretieren. Die enthaltenen Befehle sind:

- INSERT
- SELECT
- UPDATE
- DELETE
- JOIN

/F095/ Connect Service führt SQL Befehle über eine Schnittstelle des Encryption Service aus

Der Connect Service des Mediators führt nach dem korrekten interpretieren eines SQL Befehls, diesen über die Schnittstelle des Encryption Service aus.

/F100/ Zeitmessung am Server während einer Datenbankabfrage im Aktivitätslog
Der Mediator Dienst misst die verstrichene Zeit vom Empfang eines SQL Statements bis zum Senden der Ergebnisse an den Client. Dieser Wert wird gemeinsam mit der SQL Anfrage im Aktivitätslog vermerkt.

/F105/ Server vermerkt Fehlermeldungen im Error Log
Tritt bei der Ausführung eines Befehls ein Fehler auf, so vermerkt der Server die Fehlermeldung, sowie den zugehörigen Befehl in der Error Logdatei auf dem lokalen System. Sollten Fehler zur Laufzeit der Server Anwendung auftreten, so werden diese ebenfalls im Error Log gespeichert.

/F110/ Server besitzt eine Schnittstelle für Administratorzugriffe
Der Administrator kann über eine Konsolen Schnittstelle auf den Mediator Dienst zugreifen um zum Beispiel Einstellungen zu ändern oder Log Dateien abzurufen.

/F115/ Cloud Datenbank Login kann über Administratorschnittstelle geändert werden
Ein Administrator kann über die Konsolen Schnittstelle für den Mediator Dienst die Authentifikationsdaten (also Benutzername und -passwort), sowie die Cloud Server Adresse, einsehen und bearbeiten.

/F120/ Mediator speichert Login Daten für Cloud Datenbank
Der Mediator Dienst vermerkt die Benutzerdaten für die Authentifizierung an der Cloud Datenbank und legt diese in einer Konfigurationsdatei ab.

Clientseitige Wunschkriterien

/F130/ Client Anwendung bietet Export für Daten an
Die Client Anwendung bietet eine Exportfunktion für die in der Ergebnistabelle angezeigten Daten an. Exportiert werden kann entweder zur Präsentation in PNG oder PDF Formate oder zur Weiterverarbeitung als SQL oder XML Datei.

/F135/ Client Anwendung bietet Import für SQL Dateien an
Mit der Client Anwendung können bereits bestehende Datenbanken über eine SQL Datei eingelesen werden. Diese wird dann sukzessiv über SQL Befehle an den Mediator gesendet und somit auf der Cloud Datenbank aufgebaut.

/F140/ Client Anwendung kommuniziert verschlüsselt mit dem Mediator
Jegliche Kommunikation zwischen Client und Mediator Dienst im lokalen Netzwerk erfolgt verschlüsselt. Dies kann besonders bei sehr großen internen Netzen die Sicherheit kritisch erhöhen.

/F145/ Verschlüsselungsalgorithmen in der Client Anwendung wählbar
In der Client Anwendung ist über eine Dropdown-Liste der Verschlüsselungsalgorithmus wählbar, den der **Encryption** Service für die verbundene Datenbank verwendet soll.

/F150/ Speicherung der User Credentials und Server Adresse
Die Client Anwendung bietet dem Benutzer an die eingegebenen Benutzernamen und Benutzerkennwörter sowie die Server Adresse zu speichern, um ein zukünftiges Verbinden mit dem Mediator zu vereinfachen.

/F155/ Client Anwendung besitzt einen Cloud Tab
In der Client Anwendung existiert ein Fenster, in dem eine Ordnerstruktur dargestellt wird, die sich verschlüsselt auf dem Datenbankserver befindet. Durch diese Ordnerstruktur kann wie gewohnt, durch Mausklicks auf Ordner, navigiert werden.

/F160/ Dateien können in den Cloud Tab geladen, heruntergeladen und gelöscht werden

Durch hineinziehen in den Cloud Tab oder durch ein Dateiauswahlfenster können Dateien, die sich auf dem lokalen System befinden auf den Cloud Server geladen werden.

Mit einem Linksklick auf ein Objekt im Cloud Tab öffnet sich das zugehörige Kontextmenü, in dem die Datei entweder gelöscht oder heruntergeladen werden kann.

Das Löschen einer Datei, ist auch durch auswählen und betätigen der Entfernen Taste möglich.

Serverseitige Wunschkriterien

/F180/ Server Anwendung kommuniziert verschlüsselt mit dem Client

Jegliche Kommunikation zwischen Client und Mediator Dienst im lokalen Netzwerk erfolgt verschlüsselt. Dies kann besonders bei sehr großen internen Netzen die Sicherheit kritisch erhöhen.

/F185/ Server bietet verschiedene Verschlüsselungsalgorithmen an

Der Mediator Service bietet dem Client beim Verbindungsaufbau verschiedene Verschlüsselungsalgorithmen an, zwischen denen der Client wählen kann. Die Bekanntmachung der verfügbaren Algorithmen geschieht mit dem senden der Capabilities an den Client während des Verbindungsaufbaus.

/F190/ User Credentials für verschiedene Berechtigungen auf der Cloud Datenbank

Bei der Authentifizierung fordert der Mediator eine Benutzerkennung und ein Benutzerpasswort vom Client bevor eine Verbindung hergestellt wird. Mit diesen Credentials authentifiziert der Mediator sich dann an der Cloud Datenbank.

Nichtfunktionale Anforderungen

Nichtfunktionale Anforderungen beschreiben die Rahmenbedingungen des zu entwickelnden Systems. Sie beschreiben, was das System zu leisten hat, neben den funktionalen Anforderungen. Nichtfunktionale Anforderungen werden verwendet, um Systemgrößen, wie der verwendete Speicherplatz, Einarbeitungszeit sowie maximale Antwortzeiten zu beschreiben. Man erhält durch sie eine bessere Einschätzung, wie performant das System ist.

/NF10/ Erwartete Client - Anzahl, die das System verwenden

50 bis 100 Clients müssen gleichzeitig auf das System zugreifen und ohne Probleme benutzen können.

/NF20/ Intuitivität der GUI

Die GUI des Systems muss intuitiv zu bedienen sein.

/NF30/ Übersichtlichkeit der GUI

Die GUI muss übersichtlich und klar strukturiert sein. Es darf nicht zu viel Zeit in Anspruch nehmen, Kernfeatures des Systems zu finden und zu bedienen.

/NF40/ Verhalten des Systems in der Ausführung

Das System sollte flüssig sowie ohne Verzögerungen laufen.

/NF50/ Größe der Datenbank

Datenbanken bis zu einer Mindestgröße von 50 GB sollten unterstützt werden.

/NF60/ Größe einer Datenbankantwort

Die Tabellengrößen bis zu einer Mindestgröße von 1.000.000 sollten unterstützt werden.

/NF70/ Antwortzeiten des Systems

Die Antwortzeit sollte eine Sekunde nicht übersteigen.

Systemmodell

Systemmodelle beschreiben das zu entwickelnde System auf einfache Weise. Sie helfen dabei, die Komponenten und den allgemeinen Aufbau simpel, ohne technische Details, und klar verständlich darzustellen. Im Folgenden sehen sind einige Diagramme aufgelistet, die das System beschreiben.

Das erste Diagramm beschreibt den Aktivitätsfluss bei einer Anfrage an eine Datenbank, sowie den Aktivitätsfluss der Antwort.

Das zweite Diagramm beschreibt von einer höheren Ebene die Verbindung zwischen Connection Service und Client.

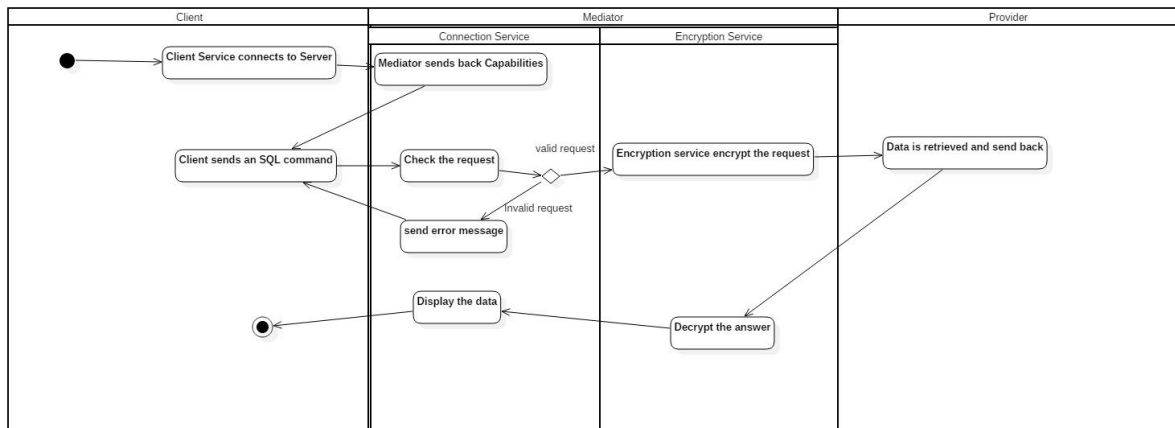


Abbildung 4: Aktivitätsdiagramm einer Anfrage

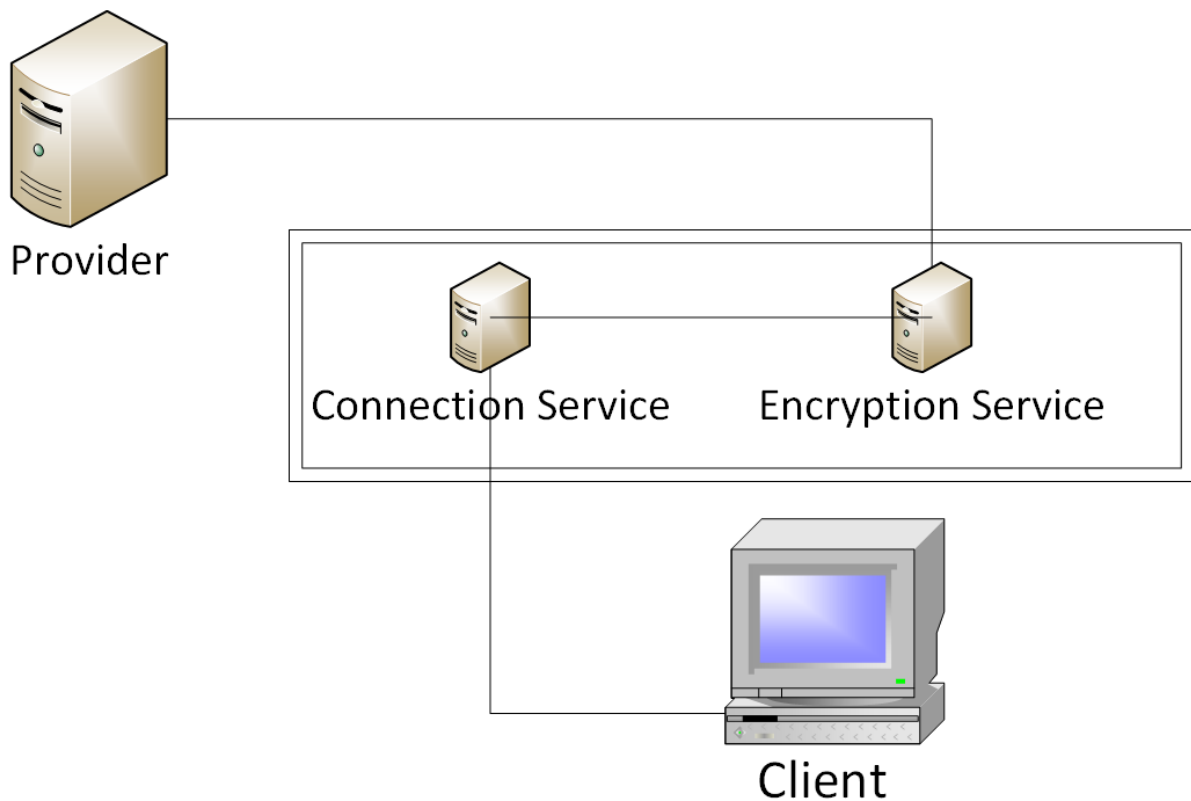


Abbildung 5: Komponentendiagramm des Produkts

Produktdaten

In diesem Kapitel sind alle Daten aufgeführt, die während dem Betrieb des Produktes anfallen und langfristig gespeichert werden sollen. Dies umfasst insbesondere Konfigurationsdaten und Aktivitätsprotokolle.

Clientseitig

/D10/ Konfigurationsdatei mit gewählten Einstellungen des Benutzers

Die Konfigurationsdatei enthält die vom Benutzer getroffenen Einstellungen bezüglich IP und Port des Mediators im neutralen Netz.

Auch wird hier vermerkt ob die Einstellung für die Automatische Verbindung der Client Anwendung mit dem Server vom Benutzer aktiviert wurde.

Die Konfigurationsdatei wird als Textdatei auf dem lokalen Festplattenspeicher im selben Verzeichnis wie die Client Anwendung gespeichert.

/D20/ Aktivitätslog der letzten Datenbankabfragen

Im Aktivitätslog wird nach jeder Datenbankabfrage vermerkt, welche Datenbank mit welchem SQL Statement abgefragt wurde und welche Latenz die Antwort des Mediators besaß. Zusätzlich markiert ein Zeitstempel den Log Eintrag.

Das Aktivitätslog wird als Textdatei auf dem lokalen Festplattenspeicher im selben Verzeichnis wie die Client Anwendung gespeichert.

Serverseitig

/D40/ Konfigurationsdatei mit Servereinstellungen

Der Server besitzt eine Konfigurationsdatei, deren Eigenschaften über die Konsolen Schnittstelle des Mediators bearbeitet werden kann.

Die Datei enthält Daten, die für die Verbindung zur Cloud Datenbank relevant sind. Also speziell die Adresse der Datenbank, sowie einen Benutzernamen und ein Benutzerpasswort um die Verbindung herzustellen.

Die Konfigurationsdatei wird als Textdatei auf dem lokalen Festplattenspeicher im selben Verzeichnis wie die Server Anwendung gespeichert.

/D50/ Aktivitätslog mit Datenbankzugriffen und Latenz Zeiten aller Anfragen

Der Mediator Service führt ein Aktivitätslog über alle Datenbank Anfragen, die über den Connect Service getätigt werden. Vermerkt wird für jeden Eintrag des Logs, von welchem Client (welcher IP-Adresse) die Anfrage gesendet wurde, sowie den gesamten SQL Befehl und die Zeit, welche benötigt wird um eine Anfrage durchzuführen. Komplettiert wird der Eintrag mit einem Zeitstempel, der angibt zu welchem Zeitpunkt die Anfrage empfangen wurde.

Das Aktivitätslog wird als Textdatei auf dem lokalen Festplattenspeicher im selben Verzeichnis wie die Server Anwendung gespeichert.

/D60/ Error Logdatei

Der Server erstellt eine Error Logdatei für eventuell auftretende Fehler, welche auf dem lokalen Speicher im selben Verzeichnis wie die Server Anwendung gespeichert wird. Dort wird jeder zur Laufzeit auftretende Fehler mit einem Zeitstempel vermerkt. Entspringt der Fehler einer SQL Anfrage auf dem Cloud Server, so wird zusätzlich das SQL Statement welches den Fehler erzeugt, sowie der Client, von dem die Anfrage ausging, in die Datei geschrieben.

Qualitätsziele

Der Dienst hat folgende Qualitätsziele, nach Priorität absteigend sortiert:

Zuverlässigkeit

Wie bei allen SQL Anwendungen ist Zuverlässigkeit das wichtigste Qualitätsziel, um die Integrität der Daten zu sicherzustellen.

Effizienz

Um einen schnellen Datenzugriff zu gewährleisten, ist Effizienz sehr wichtig, weshalb auch Funktionalitäten zur Geschwindigkeitsmessung implementiert werden.

Erweiterbarkeit

Mit der Intention unter anderem nachträglich bessere Verschlüsselungsalgorithmen hinzuzufügen, ist Erweiterbarkeit ein wichtiges Qualitätsziel. In diesem Zusammenhang gibt es ein Zusammenspiel mit Effizienz, da verbesserte Verschlüsselungsalgorithmen sich auch positiv auf die Ausführzeit auswirken können

Robustheit

Benutzerfreundlichkeit

Testfälle

In diesem Abschnitt sollen die wichtigsten Testfälle festgehalten werden, die den größten Teil der Produkt-Funktionen abdecken. Hinter den Testfällen werden die jeweils abgedeckten funktionalen Anforderungen genannt.

Testfälle für Musskriterien

/T010/	Client Anwendung starten (/F005/)
/T020/	Verbindung zum Server aufbauen (/F010/, /F040/, /F080/, /F085/)
/T030/	SQL Befehl eingeben (/F015/)
/T040/	SQL Befehl per GUI generieren (/F045/)
/T050/	Gültigen SQL Befehl senden (/F020/, /F025/, /F035/, /F050/, /F090/, /F095/, /F100/)
/T060/	Ungültigen SQL Befehl eingeben/sendern (/F070/)
/T070/	SQL Befehl per Kontextmenü generieren (/F030/)
/T080/	Server wechseln (/F040/)
/T090/	Benutzerhilfe öffnen (/F055/)
/T100/	Benutzerhilfe lesen (/F055/, /F060/)
/T110/	Tooltips anzeigen (/F065/)
/T120/	Client Log Dateien lesen (/F020/)
/T130/	Server Log Dateien lesen (/F105/, /F110/)
/T140/	Server Konfigurationsdatei lesen (/F110/, /F120/)
/T150/	Clouddatenbank Login ändern (/F115/)

Testfälle für Wunschkriterien

/T160/	Daten exportieren (/F130/)
/T170/	SQL Datenbank importieren (/F135/)
/T180/	Testdaten verschlüsseln, senden, empfangen, entschlüsseln und abgleichen (/F140/, /F180/)
/T190/	Verschlüsselungsalgorithmus wechseln (/F145/, /F185/)
/T200/	Credentials und Serveradresse nach Neustart überprüfen (/F150/)
/T210/	Cloud Tab öffnen (/F155/)
/T220/	Datei in Cloud Tab speichern (/F160/)
/T230/	Datei von Cloud Tab herunterladen (/F160/)
/T240/	Datei aus Cloud Tab löschen (/F160/)
/T250/	User wechseln (/F190/)

Grafische Benutzeroberfläche

In Abbildung 6 ist eine erste Version der grafischen Benutzeroberfläche zu sehen. Auf ihr wird der Benutzer die Features des Programms bedienen.

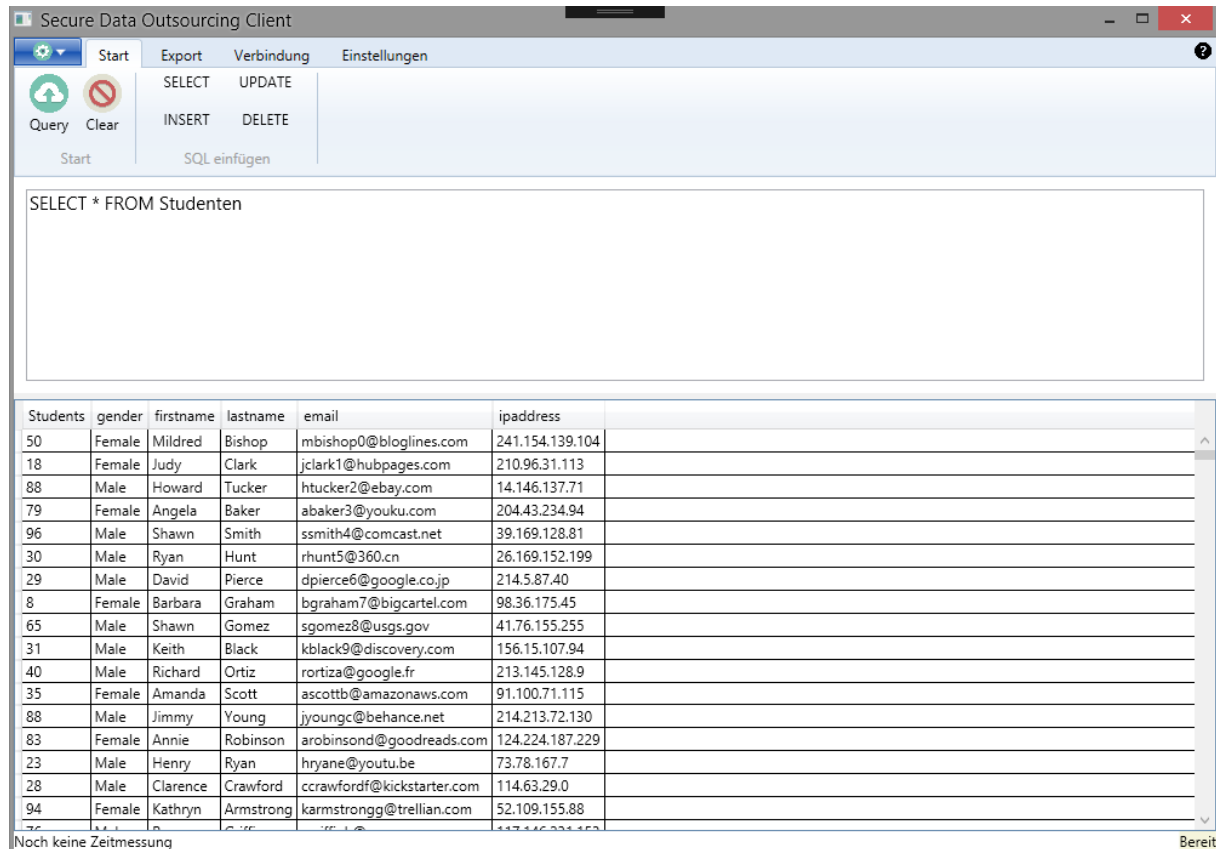


Abbildung 6: Grafische Benutzeroberfläche der Clientanwendung

Das Eingabefeld

Das Textfeld im zweiten Drittel der Anwendung dient der Eingabe von SQL-Befehlen. Es verhält sich wie ein einfacher Texteditor. Zwischen dem Eingabefeld und dem Anzeigefeld existiert eine Komponente, die das Vergrößern und Verkleinern des Eingabe- und des Anzeigefeldes ermöglicht.

Das Anzeigefeld

Das Anzeigefeld dient zur Darstellung der Ergebnisse aus einem SQL-Befehl. Die Ergebnisse sind wie in Abb. 6 zu sehen, in tabellarischer Form dargestellt. Zudem wird in der linken Seite der Statusleiste, die sich am unteren GUI Rand befindet, die gemessene Latenzzeit angezeigt.

Die Menüleiste

Die Menüleiste füllt das erste Drittel der grafischen Oberfläche und wird in verschiedene Menüpunkte eingeteilt. Jeder Menüpunkt definiert einen Eigenschaftsbereich. Diese werden im Folgenden erläutert.

Start

Dieser Menüpunkt enthält vor allem Komfortfeatures um den Benutzer bei der Erstellung der SQL-Abfragen zu unterstützen, aber auch Grundfunktionen, wie zum Beispiel den „Query“ Button.

Export

Dieser Menüpunkt dient dazu die Ergebnisse der SQL-Anfrage zu exportieren. Es wird die Möglichkeit geben verschiedene Dateiformate zu wählen.

Verbindung

Im Verbindungsmenü befinden sich Steuerelemente um die Verbindung zum Mediator zu kontrollieren. So kann zum Beispiel die Verbindung gestoppt und gestartet werden oder die Adresse des Mediators geändert werden.

Einstellungen

Dieser Menüpunkt dient dazu, allgemeine Einstellungen vorzunehmen.

Entwicklungsumgebung

Die Entwicklung des Produkts findet in der Programmiersprache C# auf der integrierten Entwicklungsumgebung Microsoft Visual Studio 2015 statt.

Es werden PCs mit dem Betriebssystem Microsoft Windows 10 genutzt.

Für die grafische Benutzeroberfläche kommt WPF (Windows Presentation Foundation) aus dem .NET Framework 4 oder höher zum Einsatz.

Zur Versionskontrolle werden hierbei Git Repositories verwendet, die auf den Team Foundation Servern von Microsoft gehostet sind.

Alle UML Diagramme in der Planungs- und Entwurfsphase werden mit Microsoft Visio oder StarUML erstellt.

Zum Testen wird das Unit Test Framework aus .NET Framework 4 oder höher verwendet.

Glossar

Aktivitätslog

Eine Log Datei, die alle Ereignisse protokolliert

Benutzer

Nutzer der virtuellen Datenbank (Firmenmitarbeiter, Privatperson etc.)

Capabilities

Eine Auflistung von Funktionen, die ein bestimmter Dienst für Benutzer anbietet.

Client(-anwendung)

Die (grafische) Benutzerschnittstelle für die Datenbank auf dem System des Benutzers

Clientsystem

Der Computer, auf dem die Client Anwendung installiert ist

Cloudanbieter/Cloud-Storage-Provider

Anbieter von Speicherplatz auf von ihm betriebenen Servern

Clouddatenbank

Die tatsächliche Datenbank, die zur Speicherung der verschlüsselten Daten dient. Sie befindet sich auf den Servern eines nicht vertrauenswürdigen

Clouddienst

Von Cloudanbietern zur Verfügung gestellte Dienstleistung

Cloudspeicher

Für den Benutzer zugänglicher Speicherplatz auf den Servern eines Cloudanbieters

Computernetzwerk

Ein System mehrerer miteinander verbundener Computer

Connect Service

Serverseitiger Teil des Produktes, der die Clientanwendung mit dem **Encryption Service** verbindet

Credentials

Daten, mit denen sich ein Benutzer authentifizieren kann. Also zum Beispiel Benutzername und ein zugehöriges Passwort.

Database-as-a-Service(DBaaS)

Cloudbasierter Ansatz zur Speicherung und Verwaltung von strukturierten Daten. DBaaS bietet ähnlich wie relationale Datenbank-Management-Systeme (RDBMS, zum Beispiel Microsoft SQL Server, MySQL oder andere Oracle Datenbanken) Datenbank-Funktionen.

Dateiauswahlfenster

Ein Pop-Up, welches dem Benutzer die Auswahl einer Datei ermöglicht, die von der Anwendung verwendet werden soll.

Daten

Eine interpretierbare Darstellung von Information in formalisierter Art, geeignet zur Kommunikation, Interpretation oder Verarbeitung

Datenbank

Eine große Menge von Daten, die in einem Computer nach bestimmten Kriterien organisiert sind und komplexe Abfragen zulassen

Datensatz

Eine Gruppierung von Daten, die durch bestimmte Faktoren zusammengehörig sind.

DELETE

SQL Befehl um Daten aus einer Tabelle zu löschen.

Dropdown-Liste

Steuerelement einer GUI mit dem ein Wert aus einer vordefinierten Liste gewählt werden kann.

Encrypt Service

Serverseitige Software, die SQL Datenbanken und Befehle verschlüsselt und anschließend auf den Cloudspeicher überträgt und vom Cloudspeicher Datenbanken lädt, entschlüsselt und an den Connection Service weitergibt

Feature

Funktion eines Programms

Git

Git ist eine freie Software zur verteilten Versionsverwaltung von Dateien, die ursprünglich für die Quellcode-Verwaltung des Linux-Kernels entwickelt wurde.

Git Repository

Speicherplatz, der für die Versionsverwaltung Git verwendet wird

Implementieren

Umsetzen eines Features

INSERT

SQL Befehl um Daten in eine Tabelle einzufügen.

Integrierte Entwicklungsumgebung (IDE)

Eine in ein interaktives Anwendungsprogramm gepackte Programmierungsumgebung, das den Softwareentwickler in seinen Entwicklungs- und Routinearbeiten unterstützt

Internes Netzwerk/Intranet

Ein lokales, abgeschlossenes Computernetzwerk

IP-Adresse

Jeder Computer in einem Netzwerk bekommt eine eindeutige IP-Adresse zugewiesen, sodass Identifikation und Kommunikation im Netzwerk ermöglicht wird.

Kommunikation

Jeglicher Datenverkehr

Konfigurationsdatei

Eine Datei, in der Einstellungen gespeichert sind die zur Laufzeit der Anwendung getroffen werden können. Bei einem erneuten Start der Anwendung können somit die zuvor konfigurierten Einstellungen wieder geladen werden.

Kontextmenü

Ein Benutzermenü im User Interface, das durch einen Klick auf die rechte Maustaste aufgerufen werden kann und Funktionen für das jeweilige Objekt anbietet, auf das geklickt wurde.

Latenzzeit

Eine Zeitspanne, die angibt wie lange ein zugehöriger Vorgang für die Abfertigung benötigt.

Log

Datei, die Ereignisse wie Errors oder Zugriffe protokolliert

Mediator

Software im internen Netzwerk (LAN/Intranet etc.), der einen tatsächlichen SQL Server simuliert und die Datenbanken auf einen externen Cloudserver auslagert.

Pop-Up

Eine grafische Benutzerschnittstelle, die bei einem definierten Ereignis angezeigt wird.

Schnittstelle/API

Eine Programmierschnittstelle (application programming interface) ist ein Kommunikationsprotokoll zwischen den verschiedenen Teilen einer Software Anwendung.

SELECT

SQL-Befehl, um Daten aus einer Tabelle auszulesen.

Server

In diesem Zusammenhang wird der Mediator beziehungsweise der enthaltene Connect Service bezeichnet

SQL

“Server Query Language” Skriptsprache zur Manipulation von Datenbanken

SQL-Befehl/SQL-Anfrage

Eine Anweisung für die Datenbank, die in SQL Skriptsprache verfasst ist.

TCP

Transmission Control Protocol - Ein Netzwerkprotokoll, welches definiert, auf welche Art und Weise Daten zwischen Computern ausgetauscht werden sollen.

TCP Verbindung

Eine Netzwerkverbindung zwischen zwei Computern, welche mit dem TCP Protokoll kommunizieren.

Tooltip

Ein Pop-Up Fenster auf einer Benutzeroberfläche, welches Beschreibungen zu einem bestimmten Element anzeigt.

UPDATE

SQL-Befehl um Daten in einer Tabelle zu ändern.

Verschlüsselungsalgorithmus

Ein Verschlüsselungsalgorithmus, beschreibt wie Daten von einem lesbaren Zustand in einen scheinbar chaotischen und unleserlichen Zustand konvertiert werden können und wie die Zurückübersetzung in einen lesbaren Zustand möglich ist.

Wunschfeature

Feature, das zwar sinnvoll ist, aber aufgrund von Zeit oder Geldmangel nicht implementiert werden muss