Die Blockchain zur Verwaltung von Zugriffen auf Gesundheitsdaten durch den Patienten selbst

Oscar Balcells
oscar@redmedic.net
Herr Ihlau, Deutsche Schule Barcelona
Jugend Forscht, Mathematik und Informatik

2019-2020

Kurzfassung

Das Ziel dieses Projekts ist es, dass die Personen die Hoheit über ihre Gesundheitsdaten haben und diese verwalten können. Diese liegen aber meist an unterschiedlichen Orten und sehr verschiedene Institutionen wie Ärzte, Krankenhäuser oder Forschungsinstitute haben ein Interesse auf den Zugriff. Dazu habe ich ein Blockchain-basiertes System entwickelt, damit die Patienten mit einer App ihre Daten von den spanischen Krankenhäusern abrufen können und diese Daten mit Ärzten oder Forschern teilen können. In einem zusätzlichen Schritt werde ich meine Ergebnisse mit den aktuell bekannten Verfahren des deutschen Gesundheitssystem vergleichen.

Inhaltsverzeichnis

1 Einleitung

1.1 Ideenfindung

Die Idee kam vor den Winterferien im vergangenen Jahr. Meine Familie und ich planten, ein fremdes Land zu besuchen, also benötigte ich verschiedene Impfstoffe. Ich hatte jedoch die meisten von ihnen bereits in der Vergangenheit erhalten. Es stellte sich heraus, dass das Zentrum für Reiseimpfungen in Barcelona, wo ich die entsprechenden medizinischen Untersuchungen durchführen musste, nicht bestätigen konnte, dass ich diese Impfungen erhalten hatte, obwohl ich mir sicher war, dass ich sie erhalten hatte, weil diese Informationen in der Datenbank eines anderen öffentlichen Anbieters gespeichert waren und sie daher keinen Zugriff darauf hatten. Ich musste alle Impfstoffe erneut erhalten.

Es war eine Verschwendung von Zeit und Geld und es tat auch weh. Warum war es so schwierig, diese trivialen Daten zwischen zwei öffentlichen Anbietern, die demselben Netz in derselben Stadt angehören, zu teilen? Warum ist der Prozess des Sendens von Gesundheitsdaten nicht so einfach wie das Senden eines Bildes, einer SMS oder anderer Informationen, die ständig und reibungslos durch das Internet fließen?

Einige Wochen später unterhielt ich mich mit einem spanischen Arzt über dieses Problem und er sagte mir, dass das meiner kein Einzelfall sei. Das spanische öffentliche Gesundheitssystem war in verschiedene regionale Subsysteme unterteilt und innerhalb dieser Subsysteme gab es in einigen Provinzen wie Barcelona bis zu 6 verschiedene Datenbanken, die miteinander inkompatibel waren. Es fehlte ein gemeinsames Netz für den Informationsaustausch, eine Gemeinschaftsautobahn, über die Daten von verschiedenen Anbietern fließen konnten. Dieses Gespräch war der Anfang dessen, was schließlich mein Forschungsprojekt werden sollte.

1.2 Aktueller Stand der Technik

Es wurden eine Reihe von Initiativen zur Bekämpfung dieses Problems sowohl im privaten als auch im öffentlichen Sektor gestartet. In den letzten Jahren wurden neue Standards geschaffen, wie z. B. FHIR [6] von HL7, die das Datenformat standardisieren und an neue Technologien wie REST APIs anpassen. Von Seiten der Privatwirtschaft gibt es Versuche großer Technologieunternehmen wie Google oder Microsoft, patientenorientierte Anwendungen zu entwickeln, um die Organisation dieser Daten zu erleichtern. Diese Initiativen haben jedoch nur wenige Ergebnisse erbracht, da Google Health drei Jahre nach seiner Gründung geschlossen wurde [4] und Microsoft HealthVault über die Einstellung der Plattform ab dem 20. November berichtet [5]. Andere neuere Initiativen mit neuen Ansätzen, wie Ciitizen oder Apple Health Records, befinden sich noch in der Entwicklungsphase. Andererseits setzt der öffentliche Sektor seit mehr als 20 Jahren Vorschriften um, wie z. B. den Health Insurance Portability and Accountability Act [7] in den USA 1996 oder das Gesetz 44/2003 [8] in Spanien, die Gesundheitsdienstleister verpflichten, Patienten Zugang zu ihrer Daten im Format ihrer Wahl zu gewähren.
Deutschland schlug 2006 einen radikaleren Plan vor. Der Deutsche Staat versuchte im Januar 2006, die Telematikinfrastruktur (TI) und die elektronische Gesundheitskarte (eGK) einzuführen. Zwei Großprojekte mit dem Ziel, alle Leistungserbringer im Gesundheitswesen miteinander zu vernetzen und somit den Transfer von Gesundheitsdaten effizienter zu gestalten. Die Umsetzung scheiterte jedoch vor allem aufgrund datenschutzrechtlicher Bedenken aufseiten der Ärzte. Im Jahr 2015 wurde das E-Health-Gesetz [11] verabschiedet und trat am 1. Januar 2016, also 10 Jahre nach der ursprünglichen geplanten Einführung der TI, in Kraft. Diesem Gesetz zufolge sollten Ärzte bis zum 1. Juli 2018 an die TI angeschlossen sein, sonst würden sie mit Honorarabkürzungen von 1% sanktioniert. Die TI läuft gerade in Deutschland aber die gesamte Umsetzung mit der elektronischen Patientenakte (ePa) ist für das Jahr 2021 geplant. Ähnliche Systeme wie die TI sind auch in Österreich, Norwegen und Estland eingeführt. Für diese Arbeit habe ich logischerweise die Existenz dieser Art von Systemen berücksichtigt und werde in einem späteren Abschnitt eine detailliertere Beschreibung dieser Systeme geben und sowohl positive als auch negative Aspekte in Bezug auf mein Projekt vergleichen.

Was die Anwendung der Blockkettentechnologie zur konkreten Lösung dieses Problems betrifft, so hat es auch hier in letzter Zeit Entwicklungen gegeben. Die Idee der Anwendung der Blockchain für die dezentrale Verwaltung privater Informationen wurde 2015 mit der Arbeit von der MIT Media Lab [1] geboren. Später gab es Vorschläge, wie diese Idee modifiziert werden könnte, um sie ausschließlich im Bereich der medizinischen Information umzusetzen. Einige dieser Arbeiten [2, 20] haben als Grundlage für dieses Projekt gedient und viele andere sind aufgrund der Neuartigkeit im Laufe meiner Forschung veröffentlicht worden [9, 10].

1.3 Meine Lösung

Für dieses Projekt habe ich ein System geschaffen, das es den Menschen ermöglicht, die Hoheit über ihre Gesundheitsdaten zu haben. Es erlaubt dem Patienten, die folgenden Funktionen mit meiner Desktop-App auszuführen:

1. Automatisch seine Informationen aus mehr als 90 öffentlichen und privaten spanischen Krankenhäusern oder jedem anderen Gesundheitsdienstleister, der ein von mir erstelltes API-Programm installiert hat, zu extrahieren und zu aggregieren
2. Automatische Organisation und Zusammenfassung der Informationen, sodass Patienten oder Ärzten sie effizienter verwenden können
3. Ärzten, Familienangehörigen oder Forschern einen zeitlich begrenzten oder unbegrenzten API-ähnlichen Zugang zu bestimmten Teilen der Informationen zu gewähren, oder ihnen die neueste Version ihrer Daten als PDF zu schicken
4. Auf anonyme Weise für Forscher durchsuchbar machen, welche nach Patienten mit bestimmten Merkmalen und Pathologien suchen
5. Die Möglichkeit der Erstellung eines Zugriffs-beschränkten Repositoriums mit Daten für Notfälle

Um diese Technologie zu entwickeln, um pragmatisch den tatsächlichen Bedürfnissen der lokalen Patienten gerecht zu werden und um mehr über das Problem zu erfahren, habe ich mehr als 30 Interviews mit Ärzten, Branchen-verwandten Informatikern und Patienten mit verschiedenen Pathologien aus verschiedenen Krankenhäusern geführt.

1.4 Forschung bei der Blockchain-Anwendung

Wie bereits erwähnt, gab es bereits frühere wissenschaftliche Arbeiten, die sich auf die Untersuchung der Anwendbarkeit der Blockkette für den Austausch und die Verwaltung von Daten privater Form, wie z. B. medizinische Informationen, konzentrierten. Mein System unterscheidet sich in mehreren Aspekten von anderen bestehenden Arbeiten.

Der erste Aspekt ist, dass mein Ziel nicht darin bestand, ein grundlegendes Konzept oder einen Prototyp vorzuschlagen, sondern ins Detail zu gehen und weitergehende Funktionen zu entwickeln, die die Nützlichkeit eines dezentralen Netzwerks für den Austausch medizinischer Informationen demonstrieren. Ich habe zum Beispiel eine Reihe von Smart Contracts implementiert, so dass Patienten anonym von Forschern gesucht werden können, die Patienten mit bestimmten Pathologien benötigen.

Der zweite große Unterschied ist der pragmatische Ansatz, den ich bei der Entwicklung des Systems verfolgt habe. Um dieses System für die Patienten nutzbar zu machen, müssen sie in der Lage sein, ihre Daten über dieses System zu extrahieren. Abgesehen von der Entwicklung eines Programms, das als API für Gesundheitsdienstleister, die dem Netzwerk beitreten wollen, funktioniert, habe ich das System mit verschiedenen Patientenportale in Katalonien integriert. Aus diesem Grund kann ein Patient seine Informationen aus verschiedenen Zentren, insgesamt mehr als 90, hinzufügen, auch wenn diese Zentren sich noch nicht durch eigene Entscheidung dem System angeschlossen sind. Mit anderen Worten, ich habe das System so programmiert, sodass es mit der bestehenden Infrastruktur kompatibel ist.

Ein wissenschaftlicher Fortschritt, den ich mit diesem Projekt mache, hat mit der Frage der Privatsphäre zu tun. In bisherigen Systemen waren die Identitäten der in der Blockchain registrierten Nutzer pseudoanonym und konnten mit Deanonymisierungstechniken leichter abgeleitet werden. Der Grund dafür ist, dass die verschiedenen Teile der Krankengeschichte eines Benutzers miteinander verknüpft wurden. In diesem Projekt werden Technologien wie hierarchische, deterministische Schlüssel und neue Smart Contracts implementiert, um die verschiedenen Entitäten völlig unverbunden und gleichzeitig indexierter für Personen zu machen, die auf sie zugreifen wollen.

1.5 Was ist eine Blockchain?

Eine Blockchain [12] wie Bitcoin oder Ethereum [13] ist ein verteiltes Netzwerk, in dem alle Knoten eine Liste von Blöcken mit Transaktionen teilen. Diese Liste von Blöcken mit Transaktionen beschreibt die Interaktionen und Änderungen, die im Netzwerk stattgefunden haben, und wird auch als den Zustand der Blockchain bezeichnet. Es ist sehr wichtig, dass alle Knoten im Netzwerk den gleichen Zustand haben.

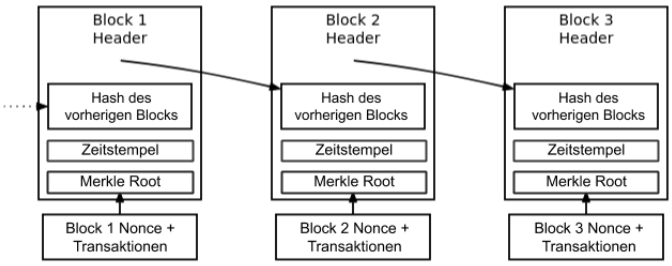

Abbildung 1: Die Struktur der Blockchain

Andernfalls entsteht das Problem der doppelten Ausgaben, nämlich, dass ein Konto mehrmals dasselbe Bitcoin an verschiedenen Konten ausgeben kann.

Um einen Konsens über diese Liste von Blöcken aufrechtzuhalten, werden so gennante Konsensmechanismen eingeführt. Die Konsensmechanismen legen Regeln, Anreize und Hemmnisse fest, damit sich die Knoten im Netzwerk auf diese Liste von Blöcken einigen sollen und wollen. Der Mechanismus, den ich kurz erläutern werde, heißt Proof of Work (PoW) [12, 14]. Es ist der einfachste und am häufigsten verwendeten und deren Anwendung war die große Innovation des anonymen Erfinders von Bitcoin. Es gibt aber auch andere und neue, effizientere und sicherere Mechanismen die in den letzten Jahren entwickelt worden sind.

In PoW ist die längste Liste von Blöcken immer die gültige. Jeder Block ist mit dem vorherigen Block verbunden, da die Vorschau der Hash-Vorschau des aktuellen Blockheaders den Hash des vorherigen Blocks enthält. In sehr vereinfachter Form lässt sich sagen, dass der Header jedes Blocks folgende Elemente enthält: die Wurzel eines Hash-Baums, der die Nonce sowie alle Transaktionen enthält, einen Zeitstempel und den Hash des vorherigen Blocks. Ein Hash des Hashes dieser Parameter wird gemacht, um den Hash des Headers des Blocks zu erhalten. Die Beziehung zwischen diesen Elementen kann durch das folgende Diagramm besser verstanden werden:

Blockchain-Miner sammeln unbestätigte Transaktionen, die von Netzwerkbenutzern gesendet werden, und nehmen diese Transaktionen in ihre Blöcke auf. In der Zwischenzeit arbeiten sie daran, eine gültige Nonce zu finden, die den Hash ihres Blockheaders niedriger macht als das Ziel, das durch die Schwierigkeit im Netzwerk vorgegeben ist. Heutzutage, damit ein Block gültig ist, muss sein Hash gleich oder niedriger als diese Hexadezimalzahl sein: 0x148edf0000000000000000000000000000000000000000000000.
Da die in Bitcoin verwendete Hash-Funktion (SHA256) vorerst resistent gegen so genannte Pre-Image-Angriffe ist, kann nicht bekannt sein, welche Nonce einen geeigneten Hash erzeugt. Aus diesem Grund testen Miner einfach mit verschiedenen Zahlen und erhöhen allmählich die Nonce, um einen zu finden, der die derzeitige Schwierigkeit erfüllt. Je höher die Schwierigkeit, desto niedriger das Ziel und desto schwieriger ist es, eine gültige Nonce zu finden. Was passiert, wenn der Miner, der nach dem nächsten Block sucht, erkennt, dass ein anderer Miner sie bereits gefunden hat? Er hat jetzt zwei Möglichkeiten:

1. Sie könnte die Arbeit am aktuellen Block beenden und zur Arbeit an der längsten Kette wechseln. Die Belohnung ging dann an den rivalisierende Miner.

2. Mit einer Rechenleistung von mehr als 50% des Netzwerks könnte er Blöcke schneller abbauen und damit den Rest des Netzes, die in einer längeren Kette arbeiten, überholen. Wenn es jedoch irgendwann gelingt, die Hauptkette in Anzahl der Blöcke zu überholen, wird die Hauptkette ungültig, da, wie oben erwähnt, die gültige Kette die längste Kette ist. Auf diese Weise sammelt er alle Belohnungen und der rivalisierende Miner, der vorher als erster einen neuen gültigen Block gefunden hatte, wird keine Belohnungen mehr haben.

Wenn ein Miner nicht mehr als die Hälfte der Rechenleistung hat, ist er nicht daran interessiert, weiter an seiner eigenen kürzeren Kette zu arbeiten, da er höchstwahrscheinlich nie an der Hauptkette vorbeikommen wird. Aus diesem Grund wählen die Miner die Option 1) und tragen so dazu bei, den Konsens im Netzwerk zu erhalten. Wenn ein Miner mit einem gültigen Block beschließt, die Regeln zu brechen und eine Transaktion zu ändern, um z. B. mehr Geld an ihm selbst zu senden, wird der Rest des Netzwerks den Block überprüfen und die Manipulation erkennen, weil die Transaktionen signiert sind, so dass der Block ungültig wird und der Miner ohne seine wertvolle Belohnung zurückgelassen wird. Dieser Aspekt fördert das gute Verhalten von Miners, die aus wirtschaftlicher Sicht nicht daran interessiert sind, den Rest des Netzes zu verärgern. Es lohnt sich mehr für ihnen, die Regeln zu befolgen.

1.6 Ethereum

Ethereum, die Plattform, auf der dieses Projektes basiert, verwendet die Blockchain-Technologie für die Verteilung und Ausführung eines globalen virtuellen Computers, auf den jeder und jede auf der Welt zugreifen kann. Die Programme in Ethereum werden Smart Contracts (SC) genannt. SCs können mit verschiedenen Programmiersprachen programmiert werden. Sie laufen in einer transparenten Ausführungsumgebung. Dieses Phänomen wird als "Open Execution" bezeichnet. Auf diese Weise können digitale Vereinbarungen zwischen verschiedenen Personen oder Organisationen mit Regeln und Ausführungsformen getroffen werden, die nicht [1] zu ändern sind.

2 Vorgehensweise, Materialien und Methode

2.1 Zentralisierter Ansatz

Die Frage, die ich mir gestellt habe, als ich zum ersten Mal versucht habe, mir eine Lösung für das Problem auszudenken, ist: Wie kann man ein medizinisches Netzwerk schaffen, das öffentlich und für jeden zugänglich ist, aber gleichzeitig die Privatsphäre und die Datensicherheit gewährleistet?

Die erste intuitive Lösung ist eine zentrale Infrastruktur, die die verschiedenen Knoten des Netzwerks miteinander verbindet. Wenn ein Krankenhaus oder ein Patient Daten an ein anderes Krankenhaus senden möchte, werden die Daten an einen zentralen Knotenpunkt gesendet, der sie an das andere Krankenhaus weiterleitet. Diese Architektur ist der Architektur sehr ähnlich, die normalerweise für soziale Netzwerke und viele andere Arten von Plattformen verwendet wird. Es gibt eine Firma im Zentrum, die den Netzwerkverkehr ermöglicht und regelt.

Nach einer kurzen Zeit der Forschung sah ich, dass diese Architektur diejenige war, die von den IT-Systemen im Gesundheitswesen in verschiedenen Ländern wie Estland, Norwegen oder Deutschland verwendet wird. Diese bieten ihren Bürgern über ein einziges Webportal Zugang zu ihren aggregierten Informationen aus allen oder fast allen verschiedenen Gesundheitszentren des Landes. Bei diesen Modellen gingen die Daten von den nationalen Krankenhäusern oder Kliniken in ein zentrales Repository, das die Daten an die Patienten oder Ärzte weitergab, die sie anforderten.

Die Hauptmängel dieser Architektur sind die folgenden:

1. Weniger Sicherheit. Es gibt einen einzigen Punkt, an dem alle Informationen gespeichert sind oder an dem der Zugang zu ihnen bearbeitet werden kann. Daher reicht nur ein erfolgreicher Angriff auf diesen zentralen Knotenpunkt aus, um alle Informationen zu erhalten.

2. Weniger Privatsphäre. Die Regierung, der zentrale Portaladministrator, hat uneingeschränkten Zugang zu all unseren medizinischen Informationen.

3. Weniger robuste Architektur. Wenn dieser zentrale Knoten, aus welchem Grund auch immer, ausfällt, fällt das gesamte Netzwerk aus und nicht nur ein Teil davon, mit fatalen Folgen für das nationale Gesundheitssystem.

Durch die Analogie mit dem Internet, das dezentral arbeitet, bin ich zu dem Schluss gekommen, dass der Inhalt selbst, in meinem Fall die Krankenakten (aber es kann jede Art von Information sein) nicht in einer einzigen zentralen Datenbank gespeichert werden muss, sondern jedes Mal, wenn ein Benutzer eine Zugangsanfrage stellt, aus den verschiedenen Einzelsystemen hinzugefügt werden kann. Auf diese Weise könnten die großen Risiken einer zentralisierten Architektur gemildert werden. Es wurden aber auch neue Komplikationen eingeführt, die zu lösen waren.

2.2 Dezentrale API und Zugriffsrechte

Die Hauptfunktionen, die der zentrale Knotenpunkt ausführt, sind jedoch für die Wartung und den Betrieb des Netzes von essenzieller Bedeutung. Sie sind die folgenden:

1. Indexierung der Daten. Der zentrale Knoten aggregiert die verschiedenen Teile einer Krankenakte, die aus unterschiedlichen Datenquellen stammen, und speichert sie in seiner Datenbank.

[1] Solange es keinen 51% erfolgreichen Angriff gibt.

2. Verwaltung der Zugriffsberechtigungen. Ein Patient oder ein Arzt kann nur auf die Informationen zugreifen, die durch seinen Benutzernamen und sein Passwort im System vorgegeben sind. Der zentrale Knoten speichert diese Berechtigungen und weiß, wer auf was zugreifen kann.

Damit das Netzwerk die oben genannten Aufgaben erfüllen kann und trotzdem seine Dezentralisierung beibehalten kann, ergibt sich die Anwendbarkeit von Blockchain. Diese Technologie bietet eine nachweislich faire, robuste und prüfbare Ausführungsumgebung, in der die Gesundheitsdaten der Patienten indiziert werden können und deren Zugriffsrechte vom Patienten verwaltet werden.
Durch den Einsatz von Smart Contracts [13, 15] kann der Patient entscheiden, wer was in seinem Datenprofil sehen darf. Die Gesundheitsdienstleister, die die Daten hosten, verwenden die Blockkette, um nach dem Empfang einer Datenanforderung über ihre API[2] zu prüfen, ob die digitale Signatur, die die Anforderung signiert hat, die Erlaubnis des Patienten hat, auf die angeforderten Daten zuzugreifen. Man kann sagen, dass die Blockchain die Erstellung von programmierbaren Zugriffsberechtigungen erlaubt. So kann der Patient z.B. den Zugriff auf seine Daten vorübergehend an einen Forscher vermieten und der Mietvertrag wäre dann komplett digital, manipulationssicher und in der Blockchain gespeichert.

Diese Architektur arbeitet mit einer kryptographischen Authentifizierung, die aus verschiedenen Gründen einen den Passwörtern überlegenen Authentifizierungsmechanismus darstellt. Der Hauptgrund dafür ist, dass der Verifizierer bei Passwörtern irgendwann im Verifizierungsprozess über das Klartext-Passwort verfügt. Sie sind daher anfälliger für Phishing-Angriffe und benötigen eine sichere Transportmethode wie HTTPS und Verschlüsselung für Transport und Speicherung. Berücksichtigt man auch die zusätzliche Komplikation, dass ein Patient nicht nur ein, sondern verschiedene Passwörter verwalten muss, um auf die Informationen seiner verschiedenen Gesundheitsdienstleister zuzugreifen, fällt die inhärente Überlegenheit eines kryptografischen Verifikationsschemas auf. Die Verwendung kryptografischer asymmetrischer Schlüssel eröffnet auch neue Möglichkeiten, wie z. B. Mehrparteiensignaturen, die eine von verschiedenen Signaturen kontrollierte Signatur ermöglichen, oder die Möglichkeit, von einem sicheren Offline-Gerät aus zu signieren und dann ein anderes, mit dem Internet verbundenes Gerät zu übertragen.

Das Blockchain-System funktioniert auch als Domain Name System (DNS), ähnlich dem, das das Internet verwendet. Ärzte können die Informationen ihrer Patienten nachschlagen, indem sie die verschiedenen APIs, auf denen diese Daten zu finden sind, einfach auffinden. In ähnlicher Weise wird die Blockchain für die Abfrage der aktualisierten URL verwendet, wo die API eines Providers gefunden werden kann.

2.3 Entwicklungsphase

Nachdem dieses System allgemein konzipiert war, galt es im nächsten Schritt, die kleinen Details zu verfeinern und es zu entwickeln. Nach zwei Treffen mit einer kleinen Gruppe von zwei Programmierern und einem Arzt, in denen sie mir halfen, die Details des Systems zu ergründen, begann ich mit der Entwicklung des Prototyps.

Dann begann ich mit Hyperledger Fabric [18] zu arbeiten, einem Open-Source-Framework, das von der Linux Foundation und IBM gemeinsam für die Entwicklung von zugelassenen und privaten Blockketten entwickelt wurde. Das Fehlen von Richtlinien und Lehrmitteln sowie die Komplexität der Architektur, die den Blockchain-Teil mit der lokalen Datenbank für die Netzwerkteilnehmer kombiniert, haben mich schnell veranlasst, eine andere Plattform zu verwenden, auf der ich mein System entwickeln kann. Die Umsetzung eines Projektes mit Hyperledger Fabric war zu komplex. Außerdem entsprach die erlaubte oder private Struktur des Netzwerks nicht den Bedürfnissen der offenen Plattform, die ich entwerfen wollte.

Ich habe Ethereum als nächste Option gewählt, da es eines der bekanntesten Projekte des Blockketten-Ökosystems ist und über eine große Anzahl von Entwicklern unterstützt wird. Es gibt auch viele Anleitungen, Videos und Bücher über Ethereum, die das Lernen enorm erleichtern. Nach weniger als 2 Wochen, in denen ich die Bücher "Mastering Ethereum" von Andreas Antonopoulous und Gavin Wood und "Building Blockchain Projects: Die Entwicklung dezentraler Blockchain-Anwendungen mit Ethereum und Solidity" von Narayan Prusty gelesen hatte, konnte ich Smart Contracts mit einem gewissen Maß an Komplexität entwickeln und auch den Betrieb des Ethereum-Netzwerks im Detail verstehen. Es dauerte 2 Wochen, bis ich alle notwendigen Smart Contracts im Netzwerk kodiert hatte. Seitdem habe ich sie modifiziert, den Code vereinfacht und neue Funktionen hinzugefügt.

[2]Diese API habe ich programmiert.

Mein nächster Schritt war der Aufbau der redmedic.org Website für das Projekt. Die Webseite war nicht nur ein relativ einfaches Werkzeug, um meiner Arbeit Sichtbarkeit und Verbreitung zu verleihen, sondern auch ein Lernprozess, der für die spätere Weiterentwicklung des Projektes sehr nützlich war. Die Website hat eine rein informative Funktion, sie hat keine Funktionalität innerhalb des Systems, und deshalb war sie einfach zu entwickeln.

Das Wissen, das ich bei der Erstellung der Webseite gewonnen habe, war notwendig, um den Kern des Projektes zu entwickeln: die Desktop-Anwendung. Für die Desktop-Anwendung habe ich die Ausführungsumgebung NodeJS [21] verwendet, zusätzlich zum ElectronJS [22] Desktop App Development Framework, das auf NodeJS aufsetzt. ElectronJS ist eine Bibliothek, die die Entwicklung von plattformübergreifenden Desktop-Anwendungen mit einfachen Werkzeugen wie HTML, CSS und Javascript ermöglicht, die auch für die Erstellung des Web verwendet werden. Die Tatsache, dass Electron auf der gleichen Technologie wie das Web basiert, hat es mir ermöglicht, Zeit zu sparen, die ich sonst in das Erlernen einer neuen Sprache oder eines neuen Frameworks hätte investieren müssen. Schließlich habe ich ReactJS [23] verwendet, um den visuellen Teil der App zu entwickeln, da ich sie schon einmal benutzt hatte. ReactJS ist ein modernes Framework, mit dem Sie Ihre eigenen programmierbaren HTML-Komponenten erstellen können, um Benutzeroberflächen zu entwickeln.

Das nächste Element, das entwickelt wurde, war die API, über die ein Anbieter verfügt und mit der er den Zugriff auf seine Daten öffnet. Für die Entwicklung der API habe ich Docker [24] und Python [25] verwendet. Python ist die am häufigsten verwendete Programmiersprache der Welt. Es ist einfach zu programmieren, einfach zu installieren und hat auch viele Bibliotheken, die die Entwicklung vereinfachen. Docker ist ein Werkzeug, das entwickelt wurde, um die Erstellung, Implementierung und Ausführung von Anwendungen durch den Einsatz von Containern zu erleichtern. Container ermöglichen es einem Entwickler, eine Anwendung mit allen benötigten Teilen, wie Bibliotheken und anderen Abhängigkeiten, zu verpacken und alles in einem Container zu versenden. Dadurch kann der Entwickler dank des Containers sicher sein, dass die Anwendung auf jedem anderen Linux-Rechner oder einem, auf dem das Docker-Programm installiert ist, ausgeführt wird, unabhängig von einer benutzerdefinierten Konfiguration, die dieser Rechner haben kann und die sich von der Maschine unterscheiden kann, die zum Schreiben und Testen des Codes verwendet wird. Das ultimative Ziel ist es, die API sehr einfach zu installieren und zu verwenden.

Danach machte ich mich daran, an der Integration mit den großen spanischen Gesundheitsportale zu arbeiten. Auf diese Weise hätten Patienten, die meine Anwendung nutzen, nicht nur die Möglichkeit, Daten herunterzuladen, die in Krankenhäusern gehostet werden, die in meinem System registriert sind, sondern auch in externen Gesundheitszentren, die nicht im System registriert sind und die nicht die von mir entwickelte API verwenden. Ich habe mich für das katalanische Gesundheitsportal La Meva Salut (LMS) [16] und das Portal des Quironsalud-Netzwerks [17] entschieden, weil sie zwei der größten Netzwerke in ganz Katalonien sind und weil ich selbst verschiedene Gesundheitszentren innerhalb dieser beiden Portale besucht habe. Aus diesem Grund konnte ich die externe Struktur der beiden Portale leichter erlernen, da ich mehr Genehmigungen und mehr Funktionen freigeschaltet hatte.

Als Erstes habe ich einen Proxy auf meinem Computer installiert und den Browser so konfiguriert, dass alle meine Anrufe über den Proxy umgeleitet werden. Auf diese Weise konnte ich die Anrufe sehen, die stattfanden, als ich mich im Portal angemeldet und alle meine Informationen heruntergeladen habe. Beide Portale hatten ernsthafte Mängel, die die User Experience verschlechterten und den Nutzen meiner Arbeit demonstrierten. So konnte man beispielsweise nicht schnell die gesamte Historie herunterladen, sondern durfte nur jedes Dokument einzeln betrachten oder manuell einen kompletten Download anfordern, der Tage dauerte, bis er akzeptiert wurde. Auf der anderen Seite bot es keine einfache Suchmaschine, die Dokumente nach Diagnose oder Datum finden konnte, sodass ich beschloss, diese Funktionen selbst in meine Anwendung zu implementieren. In einigen Treffen mit Patienten sagten sie mir, dass einer der größten Mängel dieser Portale die Tatsache sei, dass die Informationen in PDFs gespeichert seien und jedes eine andere Struktur habe, sodass sie, wenn sie ihre Krankengeschichte zum Arzt bringen wollten, ganze Ordner voller gedruckter PDFs mitbringen müssten. Informationen über ein Mobiltelefon zu tragen, war für sie auch keine Option, da PDFs nicht an diese Art von Gerät angepasst sind.

Nach einigen Tagen der Analyse, in denen ich Informationen über die Funktionsweise dieser Portale gespeichert habe, begann ich damit, die Skripte zu programmieren, um den Download von Informationen durch die App zu automatisieren. Ein Benutzer musste nur sein Passwort und seinen Benutzernamen,

der nur auf seinem Computer gespeichert war, von den verschiedenen Portale angeben, und meine Anwendung konnte in Millisekunden alle Berichte in einer einzigen Schnittstelle herunterladen, hinzufügen und anzeigen.

Die Dokumente lagen als PDF vor und waren sehr unterschiedlich aufgebaut. Um die Darstellung dieser Informationen effizienter zu gestalten, mussten sie in ein kompakteres Format umgewandelt werden. Außerdem wurde ein neues Format benötigt, um die Dokumente durchsuchbar zu machen und Information zu extrahieren.

Um den Text aus den PDFs zu extrahieren, wurde keine OCR benötigt, sondern ein normales Textextraaktionswerkzeug, weil sie durchsuchbar waren. Dafür gibt es verschiedene Bibliotheken wie z. B. Mozilla's pdf.js [26] oder pdf-to-text [27]. Nach mehreren Tests habe ich gesehen, dass die pdf-to-text-Bibliothek genauso effizient wie pdf.js ist und weniger Platz benötigt, also habe ich sie in der Anwendung installiert.

Ich habe dann ein Programm erstellt, um den Text in den PDFs zu analysieren. Dieses Programm extrahiert die wichtigsten Informationen aus dem Dokument. Zu den Informationen, die extrahiert werden können, gehören Rezepte, individuelle Testergebnisse, schriftliche Notizen des Arztes sowie der Titel des Dokuments, der Name des Arztes, der Name des Zentrums und das Datum, um die Daten in einer temporären und visuelleren Art und Weise zu organisieren. Da der Analyse-Algorithmus jedoch nicht perfekt ist und wichtige Informationen übersehen werden könnten, kann ein Benutzer auch das gesamte PDF betrachten. Eine der größten Schwierigkeiten, auf die ich bei der Programmierung dieses Algorithmus stieß, war die Knappheit der mir zur Verfügung stehenden Daten. Ich kontaktierte verschiedene Gesundheitszentren und suchte im Internet, um zu sehen, ob ich auf anonymisierten Krankenakten im Textformat zugreifen könnte. Es hat nicht funktioniert, also konnte ich nur meine eigene medizinische Krankengeschichte sowie die von Familie und Freunden nutzen, um diese Technologie zu testen.

Dieses Diagramm fasst alle für dieses Projekt entwickelten Elemente zusammen:

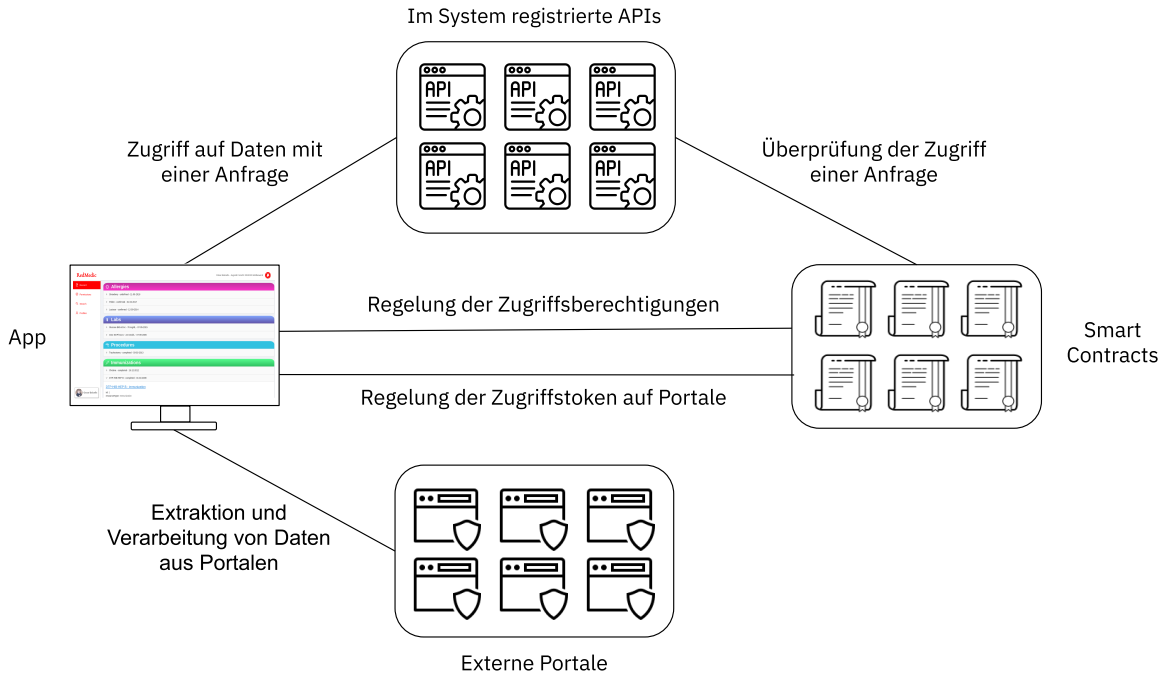

Abbildung 2: Die Elemente des Systems

3 Ergebnisse

3.1 Smart Contracts

Smart Contracts (SCs) sind das ideale Werkzeug, um Zugriffsrechte auf Patientendaten zu verwalten. Sie eliminieren vollständig das Risiko, einen Vermittler zu haben, von dem das gesamte Netzwerk abhängt.

Für diese Arbeit wurden 4 Smart Contracts entwickelt. Die Logik innerhalb der Blockchain, die ich entwickelt habe, ist jedoch vollständig formbar. Um neue Funktionalitäten zu etablieren, genügt es, einen neuen SC zu entwerfen, der mit den bestehenden interagieren kann. Der Patient kann auch seinen eigenen SC programmieren, um seine Berechtigungen zu verwalten, falls er der Sicherheit der von mir programmierten nicht vertraut. Solange die Art und Weise der Interaktion mit diesem SC die gleiche ist wie in der Norm, ist der SC absolut gültig. Diese Tatsache erhöht die Sicherheit des Systems weiter, die nicht nur von der Sicherheit meiner 4 Smart Contracts abhängt.

3.1.1 Akte

Dieser SC ist das zentrale Register der am Netzwerk beteiligten zertifizierten Institutionen. Um sich registrieren zu können, muss eine Institution von den anderen Mitgliedern manuell akzeptiert werden, daher wird geprüft, ob es sich tatsächlich um die Organisation handelt, für die sie sich ausgibt. Es ist ein demokratischer Prozess, da die Entscheidungen nicht vom Gründer des SC, sondern von allen registrierten Mitgliedern kontrolliert werden.

über diesen SC kann ein Patient die Identität einer im System registrierten Institution überprüfen. Darüber hinaus verknüpft dieser SC das Ethereum-Konto jeder Institution mit einem Zugang zu ihrer API, um auf die ihnen zur Verfügung stehenden Informationen zuzugreifen.

3.1.2 Profil

Jeder Patient besitzt im System verschiedene Profile. Bei der Erstellung diese verschiedene Contracts werden die Patientendaten so festgelegt, dass sie nur in der Blockchain (und nicht im wirklichen Leben) identifiziert werden können. Der Patient kann Ethereum Transaktionen an diesen verschiedenen Profil-Contracts schicken, um den Zugriff auf seine Daten zu regulieren.

Um jedoch die Anonymität im System zu fördern, die es erschwert, die Identitäten der Patienten in der Blockchain zu identifizieren, sollten die verschiedenen Profile eines Patienten nicht miteinander in Beziehung gesetzt werden.

Der Patient muss sich einen einzigen mnemonischen Schlüssel, der zwölf Wörter umfasst, die einen privaten kryptografischen Schlüssel darstellen, merken und kann mit diesem einzigen Wort so viele Konten generieren, wie er will. Aus diesem Grund kann er für jede Beziehung zu jedem Gesundheitsdienstleister eine andere Identität haben.

Ich habe zwei Arten von Profilverträgen entwickelt, die die zwei verschiedenen Arten von Profilen repräsentieren, die es im System geben kann.

Der erste Typ arbeitet mit den im Netzwerk registrierten Leistungserbringern im Gesundheitswesen, die den Zugriff auf Daten über eine API mit einer spezifischen Struktur ermöglichen, die die Berechtigung anhand dieser Verträge in der Blockchain überprüft und die Daten im FHIR-Format liefert. Auf die Funktionsweise dieser APIs werde ich später im Detail eingehen.

Die zweite Art von Verträgen arbeitet mit Gesundheitsdienstleistern, die nicht in meinem Blockchain-Netzwerk registriert sind, aber ein Patientenportal haben. Dieser Smart Contract versucht, die Funktionalität des ersten Typs nachzuahmen, indem er sich an die Einschränkungen externer Portale anpasst, die Datenanfragen nicht mit der Blockchain, sondern mit Passwörtern und Benutzernamen prüfen. Die von mir entwickelte Anwendung kann auf automatisierte Weise ein neues Zugangs-Token für ein bestimmtes Portal generieren. Dieser Zugangs-Token ist mehr als 3 Monate für die beiden von der Anwendung unterstützten Portale gültig und damit können Sie auf alles zugreifen, auf das der Patient bei manueller Nutzung des Portals zugreifen würde.

Wenn der Patient einem Arzt vorübergehend Zugang zur Überwachung seiner Gesundheit gewähren möchte, sendet er mit diesem Token, das mit dem öffentlichen Schlüssel seines Arztes verschlüsselt ist, eine Transaktion an diesen Profil-Vertrag in der Blockchain. Der Arzt kann es herunterladen, solange es verfügbar ist. Er entschlüsselt sie dann mit seinem privaten Schlüssel und kann damit auf die Informationen seines Patienten zugreifen. Der Patient kann dieses Token auch automatisch über die App deaktivieren, indem er sich aus dem Gesundheitsportal ausloggt und den Arzt über eine Transaktion mit der Blockchain automatisch benachrichtigt.

Obwohl die Nutzung von Informationen aus Patientenportale eingeschränkter ist, weil weder die Daten noch der Extraktionsprozess standardisiert sind, ist es für Patienten sehr nützlich, diese automatisiert hinzufügen zu können und macht das für dieses Projekt konzipierte Blockkettensystem ab der ersten Minute nutzbar, obwohl noch keine Krankenhäuser registriert sind.

3.1.3 Notfallsdaten

Dieser SC wird im Gegensatz zu den anderen Verträgen auf freiwilliger Basis und nicht anonym veröffentlicht. Sie dient dazu, dem betreuenden Gesundheitspersonal eines Patienten in Notfallsituationen, in denen es nicht in der Lage ist, seine Daten direkt durch eine Transaktion manuell zu autorisieren, zu extrahieren.

Ein konkretes Beispiel, bei dem dies nützlich sein könnte, wäre ein Verkehrsunfall. Bei diesen Unfällen gibt es die sogenannte "Goldene Stunde", die ersten 60 lebenswichtigen Minuten, damit sich der Zustand des Verletzten nicht verschlechtert und er so schnell wie möglich behandelt werden kann. In dem hypothetischen Fall, dass die Bevölkerung eines Landes in meinem Blockchain-System registriert ist, könnte das medizinische Notfallpersonal seine Daten extrahieren und ihre Allergien, Blutgruppe oder Krankheiten direkt und sofort sehen, um unnötige Tests und Risiken zu vermeiden. Dies könnte die Behandlung beschleunigen und Kosten sparen.

Mit diesem SC verknüpft der Patient seine Sozialversicherungsnummer oder seinen Personalausweis mit einer Adresse für den Zugriff auf die API eines Gesundheitsanbieters. Dieser prüft aus Sicherheitsgründen, ob diese Nummer mit der Identität des Patienten übereinstimmt. Wenn ein Mitarbeiter des Gesundheitswesens einen Patienten nach einem Unfall behandelt, kann er oder sie medizinische Informationen über den Patienten konsultieren, indem er oder sie die Dokumentation des Patienten einsieht. Der Patient kann wählen, ob er seine Notfallinformationen komplett öffentlich oder nur bestimmten Unterzeichnern wie den in meinem Blockchain-System registrierten Krankenhäusern, dem Rettungsdienst oder Familie und Freunden zugänglich machen möchte.

Der Code für diese Smart Contracts kann in meinem GitHub-Repository unter diesem Link gefunden werden: github.com/oscarbalcells/contracts-redmedic.

3.2 Desktop-App

Diese App habe ich entwickelt, damit der Patient visuell mit dem System interagieren kann. Die Funktionen der Anwendung sind auf die folgenden Tabs verteilt:

- **Daten**: Organisierte und kategorisierte Dokumente und Bildgebung einsehen
- **Zugriffsänderung**: Für jedes Profil-Contract kann der Patient Kategorien ändern, auf die eine bestimmte Account zugreifen darf
- **Portale**: Hinzufügen von externen Portale. Der Patient gibt sein Benutzername und Passwort ein (Diese Daten bleiben nur lokal verschlüsselt) und ein neuer PortalProfile SC wird erstellt
- **Notfallsdaten**: Der Patient entscheidet, welche Notfallsdaten Daten werden mit einer identifizierbaren ID, wie z. B. einer Sozialversicherungsnummer oder Ausweisnummer assoziiert und wer kann auf diese Notfallsdaten zugreifen kann
- **Search**: Der Patient kann sich jede Krankengeschichte im System ansehen, auf die er Zugriff hat
- **Accounts**: Verschiedene Profile mit unterschiedlichen digitalen Signaturen verwalten. Beispielsweise kann ein Vater nicht nur sein Konto, sondern auch das seines Kindes mit derselben Anwendung verwalten, indem er dieses Profil auf der Registerkarte Profile aktiviert.

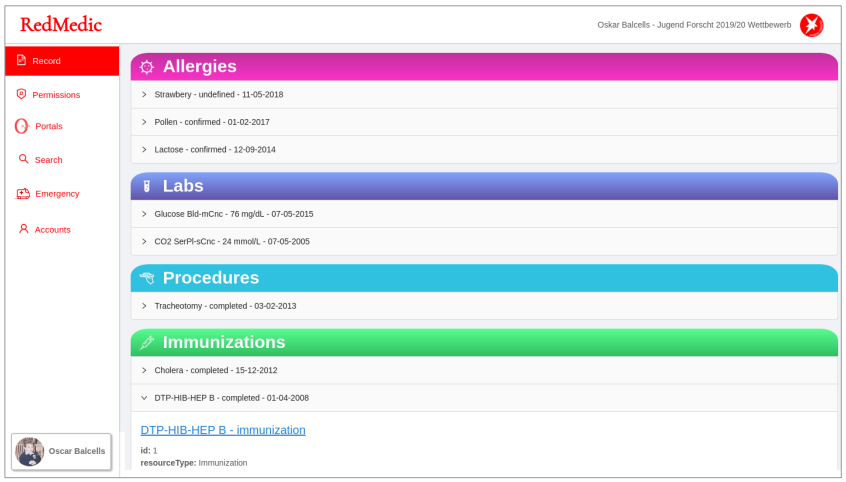

Abbildung 3: Tab zur Ansicht der Krankengeschichte, Screenshot der App

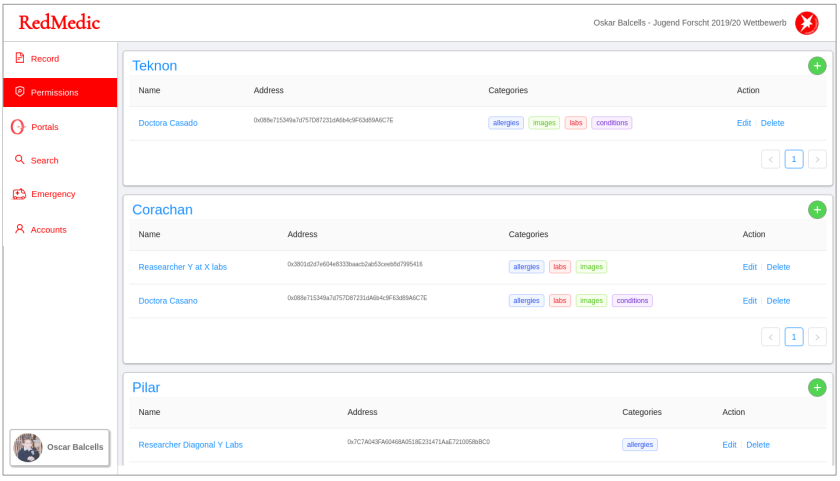

Abbildung 4: Tab zur Regulierung der Zugriffsberechtigungen

3.2.1 Warum eine Desktop-App?

Es ist eine sicherere und transparentere Art der Informationsübermittlung als über eine Web-App und gewährleistet auch die Dezentralisierung. In der App können Sie den gesamten ausgeführten Code anzeigen und bearbeiten. Nicht alle Anwender verfügen über die technologischen Möglichkeiten dazu, aber trotzdem ist es noch eine weitere Garantie geboten. Webanwendungen hingegen führen den größten Teil ihres Codes serverseitig aus, wenn es um intensive Aufgaben geht, für die es keine reinen clientseitigen JavaScript-Bibliotheken gibt. Und wenn die Informationen auf der Serverseite laufen, hat der Benutzer keine volle Kontrolle über sie, sodass er nicht weiß, was wirklich mit seinen Daten los ist. Andererseits, was nützt es, ein ganzes dezentrales System mit verschiedenen Datenbanken zu haben, um das Risiko des Netzwerks zu teilen, wenn der einzige Weg, auf das System zuzugreifen, über ein einziges Web ist, das von einer einzigen Organisation oder Person kontrolliert wird?
Die Tatsache, dass ich mich für eine Desktop-App entschieden haben, spiegelt die Philosophie des Projekts wider, ein dezentrales und transparentes System entwickeln zu wollen, dem die Nutzer nicht vertrauen müssen, weil sie den Code selbst sehen und modifizieren können.

3.3 API

Die API ist ein Computerprogramm, das das Tor zu den medizinischen Daten eines bestimmten Anbieters repräsentiert. Ziel war es, dieses Programm so einfach wie möglich zu gestalten, um es so einfach wie möglich in die verschiedenen IT-Systeme der Vielzahl von Gesundheitsdienstleistern integrieren zu können. Die API soll auch sicher sein, denn es ist völlig öffentlich. Dies macht das System zugänglicher,

ist aber gleichzeitig ein perfektes Angriffsziel für jeden Hacker. Aus diesem Grund ist der beste Weg, die API auszuführen, von einem Server oder einer Reihe von Servern, die vollständig vom Rest des Computersystems isoliert sind, wobei alle Ports geschlossen sind, außer demjenigen, der von der API verwendet wird. Diese Sicherheit kann und muss das Krankenhaus gewährleisten.

Alle Anfragen zum Zugriff auf die Daten werden signiert und der Informationsaustausch zwischen dem Client und der API wird mit Transport Layer Security (TLS) verschlüsselt. Aus der mit der Petition gesendeten digitalen Signatur kann die Ethereum-Adresse des Unterschrifteninhabers abgerufen werden. Die Blockchain wird dann gefragt, ob die Adresse das Recht hat, auf die von ihr angeforderten Daten zuzugreifen.

Ich werde nun kurz auf die verschiedenen Teilen in der API eingehen.

3.3.1 Nonce

Die Nonce ist eine einmalige Nummer, die in den Signaturen enthalten ist, um sogenannte "Replay-Attacks" zu verhindern. Andernfalls, wenn ein Angreifer eine unverschlüsselte Signatur erhält, könnte der Angreifer sie an die API senden. Die API wird also um die Daten zu erhalten. Die API wird denken, dass der Kunde, der die Signatur gesendet hat, das Ethereum-Konto, die mit der Signatur assoziiert ist, hat, und wird Ihnen daher die Daten liefern. Aus diesem Grund zeichnet die API für jede Adresse die Nonce auf, die Sie für Ihre nächste Anfrage verwenden müssen, und nach jeder erfolgreichen Anfrage wird die Nonce um 1 erhöht.

Für den Fall, dass ein Kunde die Anzahl der zuvor erfolgreich durchgeführten Anfragen verloren hat und daher nicht weiß, welches Nonce für die nächste Anfrage verwendet werden muss, muss er nur das Wort "nonce" digital unterschreiben und es über die GET-Methode an die API senden. Die API wird dann die entsprechende Ethereum-Adresse von der Signatur extrahieren und den Nonce Wert zurückgeben.

```
1 // Die Felder innerhalb von <> sind die benutzerdefinierten Parameter der Anfrage
2 GET "<host>:<port>/nonce/<id>&<sig>"
3 // Beispiel
4 GET "88.11.176.171:5000/nonce/0xfd1291b6148929ca751fcc218339d53e29519ee130650007b3243553e
5 3fb931205af92aa2a909da5a080332bd07130f6c93cf1a5b8992e30343f2e330ed38f851c"
```

3.3.2 Edit

Über diese Ressource kann der Gesundheitsdienstleister die API-Datenbank extern bearbeiten, um die Krankengeschichte eines Patienten zu aktualisieren. Die Daten, die das aktuelle Patientenprofil überschreiben, werden im Body des HTTP-Requests im JSON-Format gesendet.

```
1 PUT "<host>:<port>/edit/<sig>"
2 //BEISPIEL
3 PUT "88.11.176.171:5000/edit/490&0x801ed6e95cecd11c9b3d3c559ac9b962925d808a495e0a428370fc
4 d6e1ffe6827d026306f457ac07bae7f2c9012e4053f6c7e7f5e9c3461208b8953427c09dbd1b"
```

3.3.3 Patient

Dies ist die Ressource, mit der die Daten direkt angefordert werden können. Die Struktur dieser Ressource ist komplexer, da Sie neben der Kategorie auch die Kennung des Patienten, auf den Sie zugreifen möchten, hinzufügen müssen.

```
1 GET "<host>:<port>/patient/<contract_address>&<category>&<nonce>&<sig>"
2 //BEISPIEL
3 GET "88.1.23.1:5001/patient/0xf9993526ca0bb81508fc0d8722549758833688b0&medications&489&0x
4 b8338e86815319dc5c275bedd25da35262ba4ed3277227a96d10f56a13a38e8e64aa073d40c6c6aaea7d3d50c
5 ba50d54d95be81ed9da493d22543c03466d2aed1b"
```

Die Krankengeschichte eines Patienten besteht aus folgenden Kategorien: Personaldaten, Allergien, Medikamente, Labore, Verfahren, Impfungen, Zustände, Bilder. Jede Kategorie ist eine Liste von sogenannten "Ressourcen". Ressourcen stellen granulare klinische Konzepte dar, wie z. B. das Ergebnis

eines Bluttests oder eines dem Patienten verordneten Medikaments. Alle Arten von Ressourcen (mit Ausnahme von Bildern) folgen strengen Standards, die vorschreiben, welche Art von Informationen die Ressource enthält und welche Struktur sie hat. Diese Normen werden vom FHIR und Argonaut-Projekt, einer Initiative der internationalen Gesundheitsorganisation HL7, auferlegt.

```
{
"resourceType": "MedicationPrescription",
"dateWritten": "25-05-2013T19:32:52",
"status": "active",
"patient": { "reference": "Patient/123456789", "display": "Oscar Balcells" },
"prescriber": { "reference": "Practicioner/987654321", "display": "Doktor Smith" }
"reasonCodeableConcept": {
"coding": [{
"system": "https://browser.ihtsdotools.org/", "code": "13645005",
"display": "Chronisch obstruktive Lungenerkrankung"
}]
},
"dosageInstruction": [{
"text": "3 mal pro Tag",
"timingSchedule": {
"event": [{ "start": "04-08-2013", "end": "11-05-2013" }],
"repeat": { "frequency": 3, "duration": 1, "units": "d" }
},
"route": {
"coding": [{
"system": "https://browser.ihtsdotools.org/", "code": "394899003",
"display": "Orale Verabreichung der Behandlung"
}]
},
"doseQuantity": { "value": 10, "units": "ml", "system": "https://unitsofmeasure.org" }
}],
"dispense": {
"validityPeriod": { "start": "04-08-2013", "end": "05-30-2013" },
"numberOfRepeatsAllowed": 20,
"quantity": { "value": 100, "units": "mcg", "code": "ug" },
"expectedSupplyDuration": { "value": 40, "units": "days", "code": "d" }
}
}
```

Code-Abschnitt 1: Eine FHIR Ressource

3.4 Portale

Wie bereits erwähnt, erlaubt das System die Integration von externen Portale, ohne dass diese mit dem Blockchainsystem in Verbindung gebracht werden müssen.

Die Automatisierung der Extraktion von Daten aus einem Portal erfordert ein einzigartiges Programm, das an den Betrieb des Portals angepasst ist. Für jedes Portal werden unterschiedliche Anforderungen gestellt. Allen gemeinsam ist jedoch, dass sie über Session-Cookies funktionieren. Aus diesem Grund funktioniert der PortalProfile-Vertrag in der Blockchain für alle von mir getesteten Portale.

In meinem GitHub-Repository finden Sie das Programm, mit dem ich Daten aus dem Quironsalud-Portal [3] automatisch extrahiere: https://gist.github.com/OscarBalcells/01206b812d9c4851414143959a5d460b.

4 Ergebnisdiskussion

4.1 Vergleich mit zentralem Ansatz (Telematik-Infrastruktur)

Aus den oben genannten Gründen kann das in diesem Dokument beschriebene System entweder als dezentral oder stark dezentral betrachtet werden. Allerdings muss auch die Existenz von Systemen, die die gleichen Probleme mit unterschiedlichen Architekturen lösen, berücksichtigt werden, um die positiven und negativen Eigenschaften eines auf Blockchain basierenden Systems hervorzuheben. Konkret wird in diesem Abschnitt die zentrale Architektur untersucht.

[3] https://www.quironsalud.es/idcsalud-client/cm/portal-paciente

Bestes Beispiel dafür ist die Deutsche Telematik-Infrastruktur (TI) [16], die zum zweiten Mal durch das E-Health-Gesetz von 2015 gestartet wurde. In dieser Infrastruktur wird derzeit das Versichertenstammdatenmanagement (VSDM) von mehr als 70 Millionen Deutschen gespeichert und es sind in Zukunft neue Funktionalitäten vorgesehen. Die VSDMs beinhalten Basisinformationen über den Versicherten (Geschlecht, Geburtsdatum, Adresse, Versicherungsstatus), Diagnosedaten (ICD-Diagnose) sowie nützliche Daten für Notfälle (Allergien) oder Medikamentenpläne.

Die TI funktioniert wie folgt:

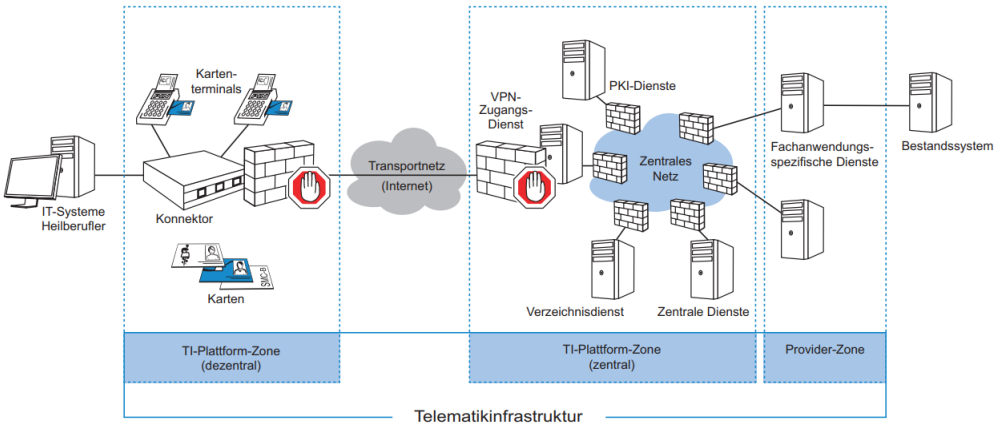

Abbildung 5: Whitepaper Telematik-Infrastruktur: gematik.de

Jeder im System registrierte Patient hat eine Karte (eGk). Der Chip dieser Karte speichert Verwaltungsinformationen, Notfalldaten und den Schlüssel, der für die Interaktion mit der zentralen Datenbank der Infrastruktur benötigt werden. Ähnlich wie eine SIM-Karte ist sie mit einem PIN-Code und einem PUK-Code geschützt, wodurch das Risiko des Diebstahls und der Datenextraktion minimiert wird. Eine GK reicht jedoch nicht aus, um mit der zentralen Datenbank interagieren zu können. Um darauf zugreifen zu können, benötigt man auch ein Terminal und eine mit einem anderen Passwort geschützte Karte, die nur Ärzte, Apotheker oder andere Heilberufe haben. Diese beiden Karten sind mit dem Terminal kombiniert, sodass nur ein legaler Zugriff auf das System möglich ist. Die Informationen werden an einen zertifizierten Router gesendet, der allen Kliniken, Krankenhäusern und Versicherungsgesellschaften haben, und der Router kontaktiert dann die zentrale Datenbank. Die zentrale Datenbank dieses Systems wird von Arvato, einer Tochtergesellschaft von Bertelsmann, aus Berlin betrieben.

Diese Sicherheit besteht vor allem darin, dass Hacker nicht über diese Hardware verfügen, auf die theoretisch, nur bestimmte berechtigte Personen oder Institutionen zugreifen können. Der CCC hat gerade sensible Zugangskarten zum deutschen System erhalten und die Gesamtsicherheit als unzureichend nachgewiesen [34].

4.1.1 Vorteile des dezentralen Systems

Das System, um das es in diesem Dokument geht, sucht nach etwas ganz anderem. Anstatt, wie in der TI, nach dem Vertrauen der Benutzer zu suchen, wendet es die Blockchain-Technologie an, sodass der Benutzer dem System nicht vertrauen muss. Dieser grundlegende Unterschied spiegelt sich in der Verwendung von Smart Contracts wider, die völlig transparent, prüfbar und für den Benutzer anpassbar sind. Auch in der Desktop-Anwendung, wo der Code zugänglich und veränderbar ist, oder in den APIs, die an die Bedürfnisse der Lieferanten angepasst werden können, um neue Sicherheitsmaßnahmen zu ergreifen. Die Tatsache, dass es keine zentrale Datenbank gibt, bedeutet, dass es keinen zentralen Punkt gibt, auf dem das gesamte System ruht. Die Informationen werden ausgeteilt, ebenso wie das Risiko, dass sie gestohlen werden. Wenn ein Hacker die Informationen der 70 Millionen im System registrierten Deutschen stehlen wollte, müsste er den privaten Schlüssel von 70 Millionen Deutschen stehlen oder auch die 72.000 verschiedenen APIs hacken. Die Blockkette fügt der zentralen Datenbank keine weitere Schutzebene hinzu, macht diese zentrale Datenbank jedoch unnötig. Verschiedene Initiativen wie "TI-frei" [28] oder "Freiheit für 1 Prozent" [29] spiegeln bereits diese Risiken der Datenspeicherung von

Millionen von Deutschen in der TI wider, lehnen die Integration des Systems in ihre Kliniken und Zentren ab und opfern damit 1% ihrer vom Bund als Geldstrafe erhobenen Sanktionen.

Derzeit werden Unterschriften gesammelt, um der Bundesregierung ein Manifest zur Beendigung der IT-Entwicklung vorzulegen.

4.1.2 Nachteile des dezentralen Systems

Der Hauptnachteil des in diesem Projekt vorgeschlagenen dezentralen Systems besteht darin, dass die Lieferanten ihre eigene Infrastruktur, das API, selbst aufbauen müssen, um teilnehmen zu können. Es gibt viele kleine und mittlere Kliniken in Deutschland, die nicht über die technologischen Mittel verfügen, um einen Server 24 Stunden am Tag, 7 Tage die Woche zu betreiben. Sie könnten diesen Service an Softwareunternehmen auslagern. Dies ist nicht die ideale Lösung, da sie die Patientendaten in die Hände Dritter legen und auch dazu beitragen würde, das System etwas mehr zu zentralisieren. Sie haben auch kein fortgeschrittenes Wissen über digitale Sicherheit, was zu einer Zunahme der Anzahl von Computerangriffen führen könnte. Aus diesem Grund ist es notwendig, dass dieses dezentrale System Realität wird, ein einfacher, sicherer und kostengünstiger Weg als Router oder Mini-Server, damit viele dieser kleinen Konsultationen teilnehmen und zusammenarbeiten können. Das sieht in Spanien etwas anders aus. Hier gibt es sehr große Krankenhausketten, die sich mit eigenen IT-Ableitungen am System beteiligen könnten.

4.2 Weiterentwicklung

Dieses Projekt kann auf verschiedene Weise weiterentwickelt werden. Die erste und wichtigste ist die Entwicklung einer iOS oder Android App. Das Handy ist einfach das von den meisten Menschen am häufigsten verwendete Werkzeug und keiner der Patienten, mit denen ich gesprochen habe, bringt seinen Computer zum Arzt. Die Entwicklung einer solchen App gibt dem Patienten die Möglichkeit, mit dem Arzt zu sprechen und ihm effizient seine Krankengeschichte zu zeigen. Viele Patienten, mit denen ich gesprochen habe, würden gerne das iPad zum Arzt bringen, anstatt eine Menge Papierkram mit sich herumtragen zu müssen. Papierinformationen sind für den Arzt einfach schwieriger zu verstehen und zu verwenden.

Ein weiterer Bereich, in dem das Projekt verbessert werden könnte, ist die Nutzung von Patientendaten. Es gibt eine große Chance, den Patienten den Zugang zu klinischen Studien zu erleichtern. Eine der größten Herausforderungen für Forscher, die klinische Studien entwickeln, besteht heute darin, Patienten mit spezifischen Profilen zu finden. In den USA halten 80% der klinischen Studien die Registrierungfristen nicht ein. Während sich 25% der Patienten für klinische Studien qualifizieren können, nehmen weniger als 5% der erwachsenen Krebspatienten tatsächlich teil. Allein in diesem Land ist es möglich, mehr als 250.000 zusätzliche Patienten in Studien aufzunehmen.

Die Anwendung könnte die Krankengeschichte des Patienten scannen, um ihm verschiedene klinische Studien anzubieten, an denen er/sie interessiert sein könnte, teilzunehmen.

Eine weitere wichtige Entwicklung betrifft die dezentrale Plattform, auf der die Systemlogik arbeitet. Um die Zugriffsberechtigungen zu regeln, müssen Patienten mit der Blockchain interagieren, indem sie Transaktionen senden. Diese Transaktionen haben variable monetäre Kosten, die je nach dem Preis des Ethers, der nativen Kryptowährung von Ethereum, schwanken. Für die Zukunft könnte das Ethereum-System abgespalten werden und eine, von Ethereum unabhängige, Blockchain bilden, in der Transaktionen billiger oder sogar kostenlos wären. Damit dies funktioniert, müssten die Miners es jedoch sicher aufbewahren. Anreize, diese Bergleute für das Netzwerk zu gewinnen, können z. B. anonymisierte Daten sein. Auf diese Weise wären die gleichen Forscher die Bergleute und sie würden Informationen im Austausch für ihre Rechenleistung erhalten. Eine weitere Möglichkeit wäre, Proof-of-Authority anstelle Proof-of-Work zu haben, um Rechenressourcen zu sparen. Die Bergleute wären in diesem Fall die gleichen registrierten Organisationen. Dies würde die Fähigkeit des Netzwerks, Transaktionen zu verarbeiten, erheblich erhöhen.

5 Zusammenfassung

Mit diesem Projekt habe ich ein System beschrieben und entwickelt, das den Austausch und Zugang zu medizinischen Informationen erleichtert. Dieses System ist dezentralisiert und wird als Alternative zu Systemen mit zentralisierter Architektur wie z.B. Telematikinfrastruktur präsentiert. Darüber hinaus habe ich dieses System so konzipiert, dass es mit der bestehenden Infrastruktur, den Patientenportale der spanischen Krankenhäuser, kompatibel ist. Das Endziel dieses Projekts ist es, dass die Patienten es benutzen, und deshalb ist diese letzte Funktion in der Anfangsphase des Systems so wichtig. Wie bereits erwähnt, gibt es noch wichtige Funktionalitäten, die hinzugefügt werden könnten, und mein Ziel ist es, daran weiter zu arbeiten. Andererseits habe ich aus den von mir durchgeführten Interviews gesehen, dass auf Seiten verschiedener Patientengruppen mit chronischen Krankheiten Interesse besteht, dies nutzen zu können, so dass ich ihnen in einigen Monaten eine Beta-Phase des Systems anbieten möchte, damit sie es ausprobieren können.

6 Unterstützungsleistungen

Ich wollte vor allem Xose Neguillo danken, dem Arzt, der mir geholfen hat, dieses Problem zu verstehen und mein Projekt an die Bedürfnisse der Patienten anzupassen. Ich möchte mich auch bei meinen Eltern, Herrn Ihlau (meinem Betreuer im Projekt) und allen Patienten und Experten, die sich die Mühe gemacht haben, mir bei diesem Projekt zu helfen, bedanken

Quellen

[1] Guy Zyskind, Oz Nathan, Alex Pentland. *Decentralizing Privacy: Using Blockchain to Protect Personal Data 2015 IEEE Security and Privacy Workshops 2015 180-184*

[2] Peterson, Deeduvanu, Kanjamala, Boles. *A Blockchain based approach to health information exchange networks. NIST Workshop Blockchain Healthcare (Vol. 1, pp. 1-10)*

[3] Andreessen Horowitz Podcast: Dark data in healthcare
https://a16z.com/2019/01/15/dark-data-healthcare-patients-platforms-hipaa/

[4] Mobihealthnews: Google Health shuts down
https://www.mobihealthnews.com/11453/official-google-health-shuts-down-because-it-couldnt-scale

[5] Medcitynews: Microsoft Healthvault shutting down
https://medcitynews.com/2019/04/microsoft-healthvault-is-officially-shutting-down-in-november

[6] Fast Healthcare Interoperability Resources (FHIR) http://hl7.org/fhir

[7] Assistance, H. C. (2003). Summary of the HIPAA privacy rule. Office for Civil Rights.

[8] Spanisches Gesetz: Boletín Oficial del Estado, Ley 41/2003
https://www.boe.es/eli/es/l/2002/11/14/41/con

[9] Liu, Xiaoguang Wang, Ziqing Jin, Chunhua Li, Fagen Li, Gaoping. (2019). *A Blockchain-based Medical Data Sharing and Protection Scheme. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2937685.*

[10] Xu, J., Xue, K., Li, S., Tian, H., Hong, J., Hong, P., Yu, N. (2019). *Healthchain: A Blockchain-Based Privacy Preserving Scheme for Large-Scale Health Data. IEEE Internet of Things Journal, 6(5), 8770-8781.*

[11] Bundesgesundheitsministerium: E-Health Gesetz
www.bundesgesundheitsministerium.de/service/begriffe-von-a-z/e/e-health-gesetz.html

[12] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*

[13] Gavin Wood, Vitalik Buterin. *Ethereum: A secure decentralised generalised transaction ledger." Ethereum project yellow paper 151.2014 (2014): 1-32.*

[14] Adam Back. *Hashcash: A Denial of Service Counter-Measure, 2002*

[15] Nick Szabo. *Smart Contracts: Building Blocks for Digital Markets, 1996*

[16] Patienten-Portal "La Meva Salut" https://lamevasalut.gencat.cat/es

[17] Patienten-Portal "Quironsalud" https://www.quironsalud.es/es/portal-paciente

[18] Hyperledger Foundation, Fabric https://www.hyperledger.org/projects/fabric

[19] Harvard Business Review. *A Big Step Toward Giving Patients Control Over Their Health Care Data* https://hbr.org/2019/03/a-big-step-toward-giving-patients-control-over-their-health-care-data

[20] Jiang, Shan and Cao, Jiannong and Wu, Hanqing and Yang, Yanni and Ma, Mingyu and He, Jianfei *Blochie: a blockchain-based platform for healthcare information exchange, 2018 IEEE International Conference on Smart Computing (SMARTCOMP), 49–56, 2018 IEEE*

[21] NodeJS https://nodejs.org/de/about/

[22] ElectronJS https://electronjs.org/

[23] ReactJS https://reactjs.org

[24] Docker https://de.wikipedia.org/wiki/Docker_*Software*

[25] Python https://de.wikipedia.org/wiki/Python_*Programmiersprache*

[26] PDF.js https://mozilla.github.io/pdf.js/

[27] pdf-to-text https://www.npmjs.com/package/pdf-to-text

[28] Telematik-Infrastruktur Frei https://www.ti-frei.de/

[29] Freiheit für 1 Prozent https://www.freiheit-fuer-ein-prozent.de/

[30] *Gesellschaft für Telematikanwendungen der Gesundheitskarte: Datenschutzwhitepaper* https://www.gematik.de/telematikinfrastruktur/

[31] SNOMED CT Browser https://browser.ihtsdotools.org/

[32] LOINC https://loinc.org/

[33] RxNorm https://www.nlm.nih.gov/research/umls/rxnorm/index.html

[34] 36th Chaos Communication Congress (36C3) https://www.heise.de/newsticker/meldung/36C3-Unsichere-Patientendaten-die-Telematik-Infrastruktur-des-Gesundh