



Dokumentation zum Blockchain Projekt: BlockCertify

Duale Hochschule Baden-Württemberg Stuttgart

von Gruppe 4:

Tobias Böck

Daniel Widmayer

Theo Remmert

Jacob Haase

Dezember 2020

Matrikelnummern

9087213, 7903581

6942414, 4691004

Kurs

Blockchain

Gutachter

Marvin König

Michael Wehrstein

Inhaltsverzeichnis

1	Einleitung	1
2	Problemstellung	2
2.1	Nutzerrollen & Funktionalität	2
2.2	Anforderung an das System	3
3	Konzept	4
3.1	Vorteile von Blockchain	4
3.2	Lösungsarchitektur	4
4	Umsetzung	6
4.1	Smart Contract	6
4.2	Web-Anwendung	7
4.3	Blockchain	8
5	Fazit	9
6	Ausblick	10
	Literaturverzeichnis	i

1 Einleitung

Viele Online Kurs Anbieter, stellen ein Zertifizierungssystem zur Verfügung, um den Teilnehmer nach Abschluss eines Kurses mit einem Zertifikat auszuzeichnen. Diese Zertifikate können dann als Bild oder als Verlinkung auf Social Media oder für Bewerbungen angegeben werden. Dabei liegt das Vertrauen bei dem jeweiligen Online Kurs Anbieter. Das Format der Zertifikate sind dabei je nach Anbieter unterschiedlich. Außerdem können Zertifikate nur zu Kursen des Anbieters ausgestellt werden. Für ein allgemeines Zertifizierungssystem, das vertrauen bietet, unabhängig von dritten Parteien ist und allgemein für beliebige Kurse Zertifikate ausstellen kann, bietet sich die Blockchain an. In dieser Dokumentation beschreibt zunächst die Problemstellung für ein dezentrales Zertifizierungssystem. Daraufhin wird das Konzept zur Umsetzung und die Umsetzung an sich beschrieben. Die Dokumentation schließt mit einem Fazit und einem Ausblick ab.

2 Problemstellung

2.1 Nutzerrollen & Funktionalität

Das dezentrale Anwendung soll zur Zertifizierung zwei Nutzerrollen abdecken, die Aussteller (Issuer) und Teilnehmer Rolle.

Beide Rollen haben folgende User Stories:

1. Als Teilnehmer möchte ich...

- erhaltene Zertifikate anschauen
- teilnehmende Kurse anschauen

2. Als Aussteller möchte ich...

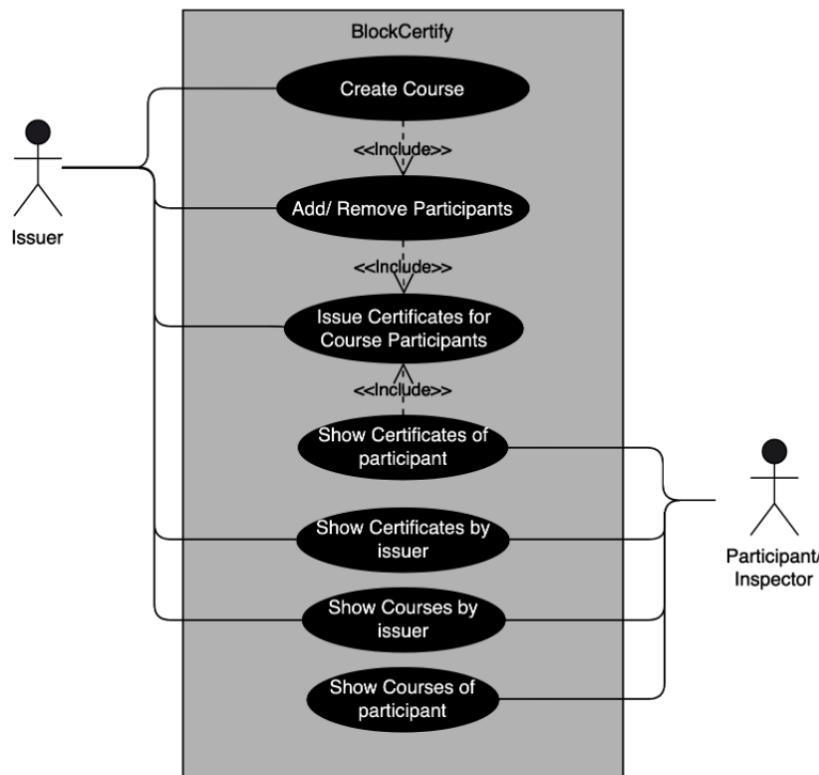
- Einen Kurs mit Teilnehmern erstellen
- Teilnehmer hinzufügen und entfernen
- Für ausgewählte Teilnehmer meines Kurses Zertifikate ausstellen

3. Als Teilnehmer oder Issuer möchte ich nach einer Adresse Suchen und folgendes Ergebnis erhalten...

- Kurse, die von dieser Adresse ausgestellt wurden
- Kurse, an denen diese Adresse teilnimmt
- Zertifikate, die von dieser Adresse ausgestellt wurden
- Zertifikate, die diese Adresse erhalten hat

4. Als Teilnehmer oder Issuer möchte ich eine benutzerfreundliche Oberfläche, die über eine "LogInSeite auf Probleme mit der MetaMask Verbindung aufmerksam macht.

Daraus ergibt sich folgendes Use-Case-Diagramm:



Die Benutzeroberfläche umfasst alle Funktionen für Issuer und Teilnehmer. Beide Rollen sehen und bedienen die gleiche Benutzeroberfläche.

2.2 Anforderung an das System

Das Zertifizierungssystem speichert Kurse und Zertifikate dezentral. Zertifikate können nach Ausstellung nicht verändert werden. Zertifikate beinhaltet die Aussteller Adresse und die Teilnehmer Adresse, sodass zu einem Zertifikat der Aussteller identifiziert werden kann. Das Zertifikat ist nur wertvoll, wenn die Aussteller Adresse von einer vertrauenswürdigen Organisation stammt.

3 Konzept

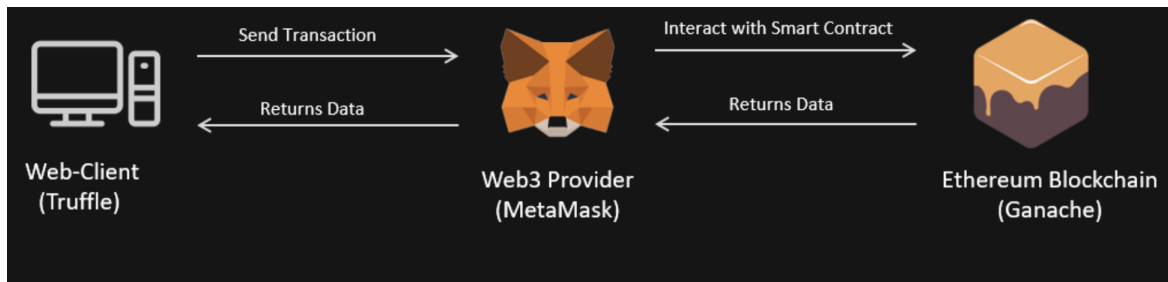
3.1 Vorteile von Blockchain

Zur Lösung des Problems, bietet sich die Blockchain Technologie an. Die Technologie ermöglicht das Abspeichern von Informationen, die danach nicht manipuliert werden dürfen. Zusätzlich werden Informationen dezentral und unabhängig von dritten Parteien gespeichert. Für die Technologie wird ein dezentrales Netzwerk eingesetzt. Zusammen mit einem Konsens Mechanismus werden die Daten (Transaktionen) über viele verteilte Knoten im Netzwerk als identische Kopie gespeichert. Dadurch wird Manipulation eines einzelnen Teilnehmers verhindert und Vertrauen gewährleistet.

Für das Zertifizierungs Projekt werden diese Eigenschaften benötigt. Die Unveränderlichkeit ist wichtig, damit eine Zertifikat nach der Ausstellung nicht gelöscht, oder verändert wird. Das ist vor allem wichtig in Bezug auf Online Kurse von Universitäten, die nach mehreren Jahren die Teilnahme an einem Kurs nachweisen müssen. Darüber hinaus ist die Abspeicherung des Zertifikats über eine Blockchain eine allgemeine, von Online-Kursanbietern unabhängige Lösung, auf die vertraut werden kann.

3.2 Lösungsarchitektur

Die Umsetzung erfolgt nach dem Beispiel des DHBW Truffle Projekts (Marvin König, 2020). Die Anwendung wird vom Nutzer ausgeführt und sendet mit Hilfe von verschiedener Eingaben Transaktionen zur Blockchain. Daraufhin wird die Transaktion mit anderen Transaktionen in einem Block zusammengefasst, der auf mehreren Knoten des Netzwerks, und dadurch sicher und unveränderlich abgespeichert wird.



Die Anwendung wird dem Nutzer, bzw dem Web-Client über einen Server zur Verfügung gestellt. Für die Verbindung zur Blockchain benötigt der Nutzer das MetaMask Web-Browser Plugin. MetaMask ist ein Zwischenspieler, der die Befehle von der Nutzeroberfläche erhält und mit der Blockchain kommuniziert. D. h. bei einer Nutzereingabe wird MetaMask aufgerufen, das dann alle weiteren Schritte mit der Blockchain durchführt. Dadurch können Transaktionen in die Blockchain gespeichert, sowie ausgelesen werden. Die Rollen Aussteller und Teilnehmer werden von einer Benutzeroberfläche abgedeckt. D. h. dass der Aussteller und Teilnehmer über alle Funktionen verfügt. Außerdem ist die Anwendung offen für jeden, also permissionless, mit einer permissionless Blockchain. Um dennoch nachzuweisen, dass ein Zertifikat einen Wert hat, und nicht von irgendeinem Teilnehmer gefälscht wurde, muss die Adresse des Zertifikats überprüft werden. Falls diese Adresse von einem vertrauenswürdigen Aussteller stammt, z. B. die Adresse einer Universität, ist das Zertifikat wertvoll.

4 Umsetzung

4.1 Smart Contract

Zur Umsetzung wurde die Ethereum Blockchain verwendet. Die Ethereum Blockchain bietet die Möglichkeit ein kleines Programm, sog. Smart Contract auf die Blockchain als eigene Transaktion zu beschreiben. Dazu wird die Solidity Programmiersprache verwendet.

Im Smart Contract werden die Datentypen von Kursen und von Zertifikaten, jeweils durch ein Struct definiert. Diese beihalten alle wichtigen Felder:

Zum Zertifikat wird eine imageId als String (z. B. "01" oder "06") angegeben. Dieser String ein Bild (z. B. images/01.png), das in der Benutzeroberfläche zum Zertifikat angezeigt wird.

Der Smart Contract speichert die Daten anhand von Mappings in die Blockchain. Diese Mappings wurden so angelegt, das häufige anfragen schnell durchgeführt werden:

- Aus Ausstellersicht werden ausgestellte Zertifikate u. Kurse dargestellt
- Aus Teilnehmersicht werden erhaltente Zertifikate u. teilnehmende Kurse dargestellt

Daraus ergeben sich folgende Mappings:

```
// map issuer to certificate
mapping(address => Certificate[]) public issuerCertificates;
// map participant to certificate
mapping(address => Certificate[]) public participantCertificates;

// map issuer to course
mapping(address => uint[]) public issuerCourses;
// map participant to course
```



```
mapping(address => uint[]) public participantCourses;

// map course id to course
mapping(uint => Course) public courses;
```

Anzumerken ist, dass die Kurse in einem separaten Mapping gespeichert werden und in anderen Mappings über eine Kurs Id referenziert werden. Wenn ein Kurs Inhaber Teilnehmer zum Kurs hinzufügt oder entfernt, muss dadurch nur ein Eintrag in letztem Mapping verändert werden. Es muss zusätzlich ein Eintrag aus dem Mapping, welches Teilnehmer zu Kursen zuordnet, verändert werden, um Anomalien zu vermeiden. Dieser und andere Abläufe werden in Funktionen des Smart Contracts beschrieben.

Jede Funktionsimplementation ist durch Kommentare beschrieben:

```
// Adds / issues a course with a title.
// The course issuer will be set to the address that calls this method
function addCourse (string memory _title) public {
    ...
}
...
```

4.2 Web-Anwendung

Die Web-Anwendung wird über einen Server mit Hilfe von NodeJS bereitgestellt. Die Benutzeroberfläche wurde mit HTML, CSS und dem Bootstrap Framework erstellt. Um MetaMask bei Nutzerinteraktionen aufzurufen, wurde Javascript mit der Truffle, der JQuery und der Web3.js API eingesetzt. Sobald in der Oberfläche z. B. ein Formular abgeschickt wird, wird Javascript über JQuery Events aufgerufen, woraufhin die Daten ausgelesen werden:

```
$('#submit').click(async function() {
    var title = $('#course-title-input').val();
    ...
    await App.addCourse(title);
    ...
});
```

Über Truffle und Web3.js wird daraufhin eine Transaktion an MetaMask weitergeleitet:

```
function addCourse: async function (title) {  
    return App.contracts.Certification.deployed() ...  
}
```

alle anderen Funktionen wurde dieses Schema ähnlich implementiert. Die Benutzeroberfläche sieht dann wie folgt aus:

4.3 Blockchain

Als Blockchain wird eine persönliche Ethereum Blockchain verwendet, die über die Ganache Umgebung emuliert wird. So können Transaktionen zu Testzwecken gebührenfrei und schnell durchgeführt werden. Falls die dezentrale Anwendung für die Öffentlichkeit zur Verfügung gestellt wird, ist weiterer Arbeitsaufwand notwendig.

5 Fazit

Abschließend lässt sich festhalten, dass Blockchain für unseren Anwendungsfall eine sehr gute Plattform bietet. Jegliche Zertifizierung findet öffentlich statt und es kann leicht nachvollzogen werden, von wem und aus welchem Grund ein Zertifikat erstellt wurde. Dazu kommt der plattformbedingte Vorteil grundsätzlicher Unveränderlichkeit getätigter Transaktionen auf einer Blockchain. Durch die Nutzung von Smart Contracts ist die dargestellte Lösungsarchitektur zudem in der Lage, die durch unsere User Stories definierten Anforderungen alle umzusetzen. Verschiedene Kurse können vom Endanwender der Blockchain auf Knopfdruck erstellt und die Teilnehmerzertifikate verwaltet werden. Das Interface bietet dabei ebenso wie die Zertifizierung an sich eine hohe Übersichtlichkeit und Transparenz. Nachteile unseres Ansatzes finden sich hingegen im Zertifikats-Mapping der Smart Contracts. Zu den Basisfunktionalitäten des Programs gehörend, findet an dieser Stelle die Verknüpfung zwischen Transaktion und Zertifikat statt. In der Umsetzung war das Erstellen und Ausarbeiten der Mappings intensiv und zeitaufwendig. Zwar sind im konkreten Projekt Aspekte wie das User-Feedback und Error-Handling noch ausbaufähig und auch das Nutzen, sowie Entwickeln der Anwendung bedingt durch MetaMask teilweise kompliziert und zeitaufwendig, allerdings Überwiegen die gebotenen Vorteile im Großen und Ganzen. Solidity bietet in Sachen Blockchain eine solide Basis für Smart Contract basierte Blockchain-Anwendungen. Somit war es möglich, auch in kurzer Zeit eine funktionierendes Programm, mitsamt Interface und Administration mit den wichtigsten Anforderungen zu erstellen.

6 Ausblick

Ausgehend von der bestehenden Anwendung bietet sich in mehreren Punkten die Möglichkeit des Ausbaus oder der Optimierung. Die hier genannten Optionen dienen primär der Integration in andere öffentliche Netzwerke und Anwendungsgebiete. Durch die Verlinkung auf Lernplattformen beispielsweise, kann nicht nur die Verbreitung, sondern auch die Nutzung unserer Blockchain gesteigert werden. Neben der erhöhten Reichweite in Nutzerkreisen, hat dies zudem den Vorteil transparenter und digitaler Zertifizierungen in allerlei Anwendungsgebieten, wie öffentliche Kurse oder interne Zertifizierungsprogramme. Um die Blockchain gegen Missbrauch, insbesondere Fake-Zertifizierungen zu wappnen, wird zudem noch eine CA¹ benötigt, um Zertifizierern eine Vertrauensgrundlage zu geben. Andernfalls wäre eine solche nur bei persönlichem Kennen und dem Austausch einzelner Blockchain Teilnehmer gewährleistet. Abschließend gibt es die Möglichkeit, die bisher lokal laufende Blockchain öffentlich zu integrieren. Nebst besserer Verfügbarkeit sind dadurch auch weiterführende Verwendung in Cloud-Infrastrukturen und Web-Services möglich.

¹ Certificate Authority

Literaturverzeichnis

Marvin König, N. V. (2020), 'Dhbw truffle project'. <https://github.com/mkqavi/dhbw-truffle-project>.