|  |
| --- |
| debug.png |
| Anforderungsspezifikation  Bitcoin Script Debugger  **Projektteam x**  **Rolf Zurbrügg**  **Samuel Egger**  **X0.3 - 12.11.17** |
| **Berner Fachhochschule**  Departement für Technik und Informatik |

**Inhaltsverzeichnis**

[1 Zweck des Dokuments 3](#_Toc495439568)

[2 Projektvision 3](#_Toc495439569)

[3 Projektziele 3](#_Toc495439570)

[4 Systemabgrenzung 3](#_Toc495439571)

[4.1 Prozessumfeld 3](#_Toc495439572)

[4.2 Systemumfeld 3](#_Toc495439573)

[4.3 Randbedingungen 3](#_Toc495439574)

[5 Anforderungen 4](#_Toc495439575)

[5.1 Quellen und Vorgehen 4](#_Toc495439576)

[5.2 Funktionale Anforderungen 4](#_Toc495439577)

[5.3 Technische Anforderungen 4](#_Toc495439578)

[5.4 Qualitätsanforderungen 4](#_Toc495439579)

[6 Glossar 5](#_Toc495439580)

[7 Abbildungsverzeichnis 5](#_Toc495439581)

[8 Tabellenverzeichnis 5](#_Toc495439582)

[9 Literaturverzeichnis 5](#_Toc495439583)

[10 Anhang 5](#_Toc495439584)

[10.1 Abstimmung der Anforderungen 5](#_Toc495439585)

[10.2 Definition of Ready - Checklist 5](#_Toc495439586)

[11 Versionskontrolle 5](#_Toc495439587)

# Zweck des Dokuments

Dieses Dokument beschreibt die Ziele und Anforderungen für den Bitcoin Script Debugger.

# Projektvision

Bitcoins können gekauft und verkauft werden. Der Wechsel von Besitzern wird mittels Locking und Unlocking Scripts geregelt. Die Script Sprache ist eine Bitcoin eigene Sprache. Sie ist nicht Turing mächtig ist und Stack basiert. Die Script Sprache besteht aus so genannten OP Codes. Das Locking und Unlocking Script wird aneinander gehängt und dann evaluiert. Wenn am Schluss der Evaluation True auf dem Stack steht, dann ist eine Transaktion gültig.

Vision dieses Projektes ist es, eine Webseite basierend auf HTML, CSS und JavaScript zu entwickeln, welche dem Nutzer erlaubt Bitcoin Locking und Bitcoin Unlocking Scripts zu Testen / Debugen. Zur Unterstützung des Debuggings soll der Stack visualisiert werden. Der visualisierte Stack soll den Nutzer beim Verstehen des Script Ablaufs unterstützen in dem er den Zustand des Stacks nach Ausführung von jedem OP Code anzeigt.

Durch dieses Projekt soll es Benutzern erst mal ermöglicht werden, Bitcoin Script in einem sicheren Umfeld Offline zu Testen. Dies ist wichtig, weil der Besitz und Handel von Bitcoins auf Asymmetrischer Kryptographie basiert. Das bedeutet, dass man zum Handeln von Bitcoins Public und Private Keys benötigt. Diese werden mittels Script Sprache dann geprüft. Deswegen ist der Sicherheitsaspekt essenziell bei diesem Projekt.

Dieses Projekt soll eine Open Source Lösung sein, welche hauptsächlich zu Schulungs- und Forschungszwecken eingesetzt wird. Deswegen wird kein neues Geschäftsmodell geschaffen.

# Projektziele

Die wichtigsten Ziele der Stakeholder sind:

Es soll eine Webseite in HTML, CSS und Java Script realisiert werden, welche dem Nutzer eine schrittweise Ausführung von Bitcoin-Scripten erlaubt. Zur Unterstützung des Nutzers soll der aktuelle Stack visualisiert werden. Dies soll den Nutzer beim Debugging unterstützen.

In Bitcoin-Scripten kommen Public und Private Keys zum Einsatz. Auf Grund der Verwendung dieser sensitiven Daten ist es essenziell, dass die Webseite auch Offline funktioniert.

Weitere Ziele mit geringerer Wichtigkeit sind Features zu implementieren wie Syntax Validierung, generieren von Public, Private Key Pairs zum Testen direkt auf der Webseite, so wie der Generierung von benötigten Hashes.

Ein weiteres Ziel ist es dem Nutzer einen direkten Download der Webseite zu ermöglichen, damit er dies nicht manuell tun muss.

Stakeholder in diesem Projekt sind:

* Dozenten und Studierende mit entsprechender Fachrichtung
* Bitcoin Enthusiasten
* Nutzer welche Non Standard Scripts testen, entwickeln oder nutzen wollen

# Systemabgrenzung

## Prozessumfeld

Die entwickelte Webseite soll aus Sicherheits- und Usabilitygründen eine autonome Lösung darstellen, welche primär zu Forschungs- und akademischen Zwecken dient. Deswegen sind keine Geschäftsprozesse bei der Umsetzung relevant.

## Systemumfeld

Das System ist als Schulungssoftware in sich abgeschlossen. Auf Grund dessen besteht keine Interaktion mit anderen Systemen.

## Randbedingungen

Technische Rahmenbedingungen

* Offline lauffähige Webapplikation
* Umsetzung mit JavaScript

Organisatorische Vorgaben

* Abgabe des Source Code für Review am 23.12.2017
* Code Reviews erfolgen bis am 15.01.2018
* Projektpräsentation am 18.01.2018
* Projektabschluss am 20.01.2018

Standards

* OP Code Interpretation muss äquivalent zum original Bitcoin Source Code geschehen. Der Bitcoin Source Code gilt als Standard für alle Bitcoin Implementationen.
* Alle kryptographischen Algorithmen müssen gemäss ihren Standards implementiert sein.

# Anforderungen

## Quellen und Vorgehen

Für die Umsetzung von diesem Projekt ist ein solides Verständnis von Bitcoins erforderlich.

Literatur die zum Selbststudium der Blockchain und der Bitcoin Script Sprache genutzt worden sind.

* <https://en.bitcoin.it/wiki/Main_Page>
* <https://www.youtube.com/watch?v=R5XipD_X4B8>
* <https://github.com/bitpay/bitcore-lib>
* Unterlagen von Kai Brünnler aus dem Modul BTI7537

Weiterhin ist eine Solides Verständnis von Java Script nötig. Die verwendeten mittel um diese im Selbststudium zu erlernen sind:

* Online Kurs bei Udemy ("JavaScript: Understanding the Weird Parts“)
* <https://www.udemy.com/understand-javascript/learn/v4/content>

Der Projektumfang und die Ziele sind bereits durch den Auftraggeber Kai Brünnler via Projekt-ausschreibung (Bitcoin-Skript-Debugger) vorgegeben. In der Beschreibung befindet sich ein zusätzlicher Verweis auf ein ähnliches Projekt (https://webbtc.com/script), dessen Funktionsumfang als Referenz für die Erhebung der Anforderungen hinzugezogen werden kann.

* Literatur
* Webpages
* Selbststudium

## Funktionale Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Status** | **Prio** | **Beschreibung** |
| **F1.1** | Entwurf | **M** | Die Applikation erlaubt die Eingabe von Locking und Unlocking Scripts. |
| **F1.2** | Entwurf | **P1** | Die Syntax des Bitcoin Scripts wird farblich hervorgehoben (Syntaxhighlighting). |
| **F1.3** | Entwurf | **P1** | Syntaxfehler werden in Echtzeit (d.h. während der Eingabe) erkannt und angezeigt. |
| **F2.1** | Entwurf | **M** | Die Locking und Unlocking Scripts können ausgeführt werden und das Resultat wird dargestellt. |
| **F2.2** | Entwurf | **M** | Zeilenweises Ausführen der Bitcoin Scripts. Bei jedem Ausführungsschritt werden die aktuelle Operation und der Zustand des Stacks visuell hervorgehoben. |

Tabelle 1: Funktionale Anforderungen

Attribute:

ID: eindeutige Identifikation

Status: Entwurf / geprüft / freigegeben

Priorität: Muss / Optional P1, P2, P3 / Wunsch (Nice to have)

## Technische Anforderungen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Status** | **Prio** | | **Beschreibung** | |
| **T1.1** | Entwurf | M | Das Projekt muss als offline Webapplikation ausgeführt werden können. | |
| **T2.1** | Entwurf | M | Der Source Code des Projekts ist Open Source. | |
| **T3.1** | Entwurf | M | Der vorgegebene Technologie Stack ist HTML, CSS und Java Script | |

Tabelle 2: Technische Anforderungen

## Qualitätsanforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Status** | **Prio** | **Beschreibung** |
| **Q1.1** | Entwurf | M | Es ist durch manuelle Tests sicherzustellen, dass das Projekt den erwarteten Anforderungen entspricht. |

Tabelle 3: Qualitätsanforderungen

# Glossar

**HTML (Hypertext Markup Language)**

HTML ist eine Markup Sprache welche von der Organisation w3c standardisiert und gepflegt wird. Mittels HTML können Webseiten und Webaplikationen erstellt werden.

Weitere informationen zu HTML können unter:

HTML 5 Spezifikation: <https://www.w3.org/TR/html5/>

Geschichte von HTML: <https://en.wikipedia.org/wiki/HTML>

Nachgelesen werden.

**Markup Language**

Markup sprachen erlauben die annotation eines Documentes in einer weisse welche Syntaxtisch unterscheidbar vom reinen Text ist. In Markup Sprachen werden sogenante Tags verwendet, diese dienen dazu um dem Text welcher von diesem Tag betroffen ist eine bestimmte darstellung / funktion zu verleien, bsp: Hyperlink oder Unterstrichen.

Weiter informationen zu Markup Languages können unter: <https://en.wikipedia.org/wiki/Markup_language>

Gefunden werden.

**CSS (Cascading Style Sheet)**

Mittels CSS wird die Präsentation einer Webseite einschliesslich Farbe Lay-out und Fonts beschrieben. Es erlaubt einem die Präsentation an zu passen für verschiedene Endgeräte wie grosse Bildschirme, kleine Bildschirme, Drucker und andere. CSS ist unabhänging von HTML es kann also auch mit XML-basierten markup sprachen verwendet werden.

Weiter Informationinen zu CSS finden sie unter:

<https://www.w3.org/standards/techs/css>

<https://en.wikipedia.org/wiki/Cascading_Style_Sheets>

**Java Script (JS)**

1„Der als **ECMAScript** ([ECMA](https://de.wikipedia.org/wiki/Ecma_International) 262) standardisierte Sprachkern von JavaScript beschreibt eine [dynamisch typisierte](https://de.wikipedia.org/wiki/Dynamische_Typisierung), [objektorientierte](https://de.wikipedia.org/wiki/Objektorientierte_Programmierung), aber [klassenlose](https://de.wikipedia.org/wiki/Klasse_(Objektorientierung)) Skriptsprache.“

Weiter Informationen zu können unter:

<https://de.wikipedia.org/wiki/JavaScript>

gefunden werden.

**Bitcoin**

2„Bitcoin (englisch sinngemäß für „digitale Münze“) ist eine digitale [Währung](https://de.wikipedia.org/wiki/W%C3%A4hrung), gleichzeitig auch der Name des weltweit verwendbaren dezentralen [Buchungssystems](https://de.wikipedia.org/wiki/Buchungssystem) sowie die vereinfachende Bezeichnung einer kryptografisch legitimierten Zuordnung von Arbeits- oder Rechenaufwand.[[4]](https://de.wikipedia.org/wiki/Bitcoin#cite_note-faz16082013-4)[[5]](https://de.wikipedia.org/wiki/Bitcoin#cite_note-NOZ-privates_Geld-5) [Überweisungen](https://de.wikipedia.org/wiki/%C3%9Cberweisung_(Zahlungsverkehr)) werden von einem [Zusammenschluss von Rechnern](https://de.wikipedia.org/wiki/Rechnernetz) über das Internet mithilfe einer speziellen [Peer-to-Peer](https://de.wikipedia.org/wiki/Peer-to-Peer)-Anwendung abgewickelt, sodass anders als im herkömmlichen Bankverkehr keine zentrale Abwicklungsstelle benötigt wird.“

Detailiertere Informationen können unter:

<https://de.wikipedia.org/wiki/Bitcoin>

gefunden werden.

**Blockchain**

3“Eine Blockchain (auch Block Chain, [englisch](https://de.wikipedia.org/wiki/Englische_Sprache) für Blockkette) ist eine kontinuierlich erweiterbare Liste von [Datensätzen](https://de.wikipedia.org/wiki/Datensatz), genannt „Blöcke“, welche mittels [kryptographischer Verfahren](https://de.wikipedia.org/wiki/Kryptographie) miteinander verkettet sind. Jeder Block enthält dabei typischerweise einen [kryptographisch sicheren Hash](https://de.wikipedia.org/wiki/Kryptologische_Hashfunktion) des vorhergehenden Blocks, einen [Zeitstempel](https://de.wikipedia.org/wiki/Zeitstempel) und [Transaktionsdaten](https://de.wikipedia.org/wiki/Transaktion_(Informatik)). ”

Weiter informationionen können unter:  
 <https://de.wikipedia.org/wiki/Blockchain>

Gefunden werden.

**Bitcoin Script**

Bitcoin benutzt ein Scripting basiertes System für Transaktionen. Die Bitcoin Script Sprache ist eine Stack basierte und nicht Turing mächtige (keine schleifen möglich) Script Sprache. Die Auswertung geschieht von Links nach rechts.

Ein Bitcoin Script ist im essentiellen eine liste von instructionen welche auf der Transaktion festghalten sind. Die liste von instructionen definiert wie jemenand zugriff auf die Bitcoins in der Transaktion erhält.

Weitere Informationen können unter:  
 <https://en.bitcoin.it/wiki/Script>

Gefunden werden.

**OPcode**

4„Ein Opcode, auch op code oder operation code, ist eine Zahl, die die Nummer eines [Maschinenbefehls](https://de.wikipedia.org/wiki/Maschinensprache) für einen bestimmten [Prozessortyp](https://de.wikipedia.org/wiki/Prozessor) angibt.“

Im bezug auf Bitcoins beziehen sich die Opcodes auf die einzelnen befehle in der Scripting Sprache.

Weitere informationen zu Opcodes im algemeinen:  
 <https://de.wikipedia.org/wiki/Opcode>

Bitcoin Opcodes:

<https://en.bitcoin.it/wiki/Script>

**Open Source**

5„Als Open Source (aus [englisch](https://de.wikipedia.org/wiki/Englische_Sprache) open source, wörtlich offene Quelle) wird [Software](https://de.wikipedia.org/wiki/Software) bezeichnet, deren [Quelltext](https://de.wikipedia.org/wiki/Quelltext) öffentlich und von Dritten eingesehen, geändert und genutzt werden kann. Open-Source-Software kann meistens kostenlos genutzt werden.“

Weiter informationen zu diesem Thema:

<https://de.wikipedia.org/wiki/Open_Source>

**Asymmetrische Kryptographie**

6„Ein asymmetrisches Kryptosystem oder Public-Key-Kryptosystem ist ein [kryptographisches](https://de.wikipedia.org/wiki/Kryptographie) [Verfahren](https://de.wikipedia.org/wiki/Kryptosystem), bei dem im Gegensatz zu einem [symmetrischen Kryptosystem](https://de.wikipedia.org/wiki/Symmetrisches_Kryptosystem) die kommunizierenden Parteien keinen gemeinsamen geheimen Schlüssel zu kennen brauchen. Jeder Benutzer erzeugt sein eigenes [Schlüsselpaar](https://de.wikipedia.org/wiki/Schl%C3%BCssel_(Kryptologie)), das aus einem geheimen Teil ([privater Schlüssel](https://de.wikipedia.org/wiki/Privater_Schl%C3%BCssel)) und einem nicht geheimen Teil ([öffentlicher Schlüssel](https://de.wikipedia.org/wiki/%C3%96ffentlicher_Schl%C3%BCssel)) besteht. Der öffentliche Schlüssel ermöglicht es jedem, Daten für den Besitzer des privaten Schlüssels zu [verschlüsseln](https://de.wikipedia.org/wiki/Verschl%C3%BCsselung), dessen digitale Signaturen zu prüfen oder ihn zu authentifizieren. Der private Schlüssel ermöglicht es seinem Besitzer, mit dem öffentlichen Schlüssel verschlüsselte Daten zu [entschlüsseln](https://de.wikipedia.org/wiki/Entschl%C3%BCsselung), [digitale Signaturen](https://de.wikipedia.org/wiki/Digitale_Signatur) zu erzeugen oder sich zu [authentisieren](https://de.wikipedia.org/wiki/Authentifizierung).“

Weiter Informationen zu diesem Thema können unter:

<https://de.wikipedia.org/wiki/Asymmetrisches_Kryptosystem>

gefunden werden.

# Abbildungsverzeichnis

**No table of figures entries found.**

# Tabellenverzeichnis

Tabelle 1: Funktionale Anforderungen 4

Tabelle 2: Technische Anforderungen 4

Tabelle 3: Qualitätsanforderungen 4

# Literaturverzeichnis

**Webseite**

*https://www.w3.org/TR/html52/introduction.html#introduction* 6

**Webseite**

https://en.wikipedia.org/wiki/HTML 6

**Webseite**

https://en.wikipedia.org/wiki/Markup\_language 6

**Webseite**

https://www.w3.org/standards/techs/css 6

**Webseite**

https://de.wikipedia.org/wiki/JavaScript 6

**Webseite**

https://de.wikipedia.org/wiki/Bitcoin 7

**Webseite**

https://de.wikipedia.org/wiki/Blockchain 7

**Webseite**

https://en.bitcoin.it/wiki/Script 7

**Webseite**

https://de.wikipedia.org/wiki/Opcode

**Webseite**

https://de.wikipedia.org/wiki/Open\_Source

**Webseite**

https://de.wikipedia.org/wiki/Asymmetrisches\_Kryptosystem 8

# Zitate

1 <https://de.wikipedia.org/wiki/JavaScript>

2 <https://de.wikipedia.org/wiki/Bitcoin>

3 <https://de.wikipedia.org/wiki/Blockchain>

4 <https://de.wikipedia.org/wiki/Opcode>

5 <https://de.wikipedia.org/wiki/Open_Source>

6 <https://de.wikipedia.org/wiki/Asymmetrisches_Kryptosystem>

# Anhang

## Abstimmung der Anforderungen

Bei der Abstimmung der Anforderungen sind keine Probleme aufgetreten.

## Definition of Ready - Checklist

Testszenario auflisten

# Versionskontrolle

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Datum** | **Beschreibung** | **Autor** |
| X0.1 | 02.10.2017 | Dokument erstellt | Rolf Zurbrügg |
| X0.2 | 10.10.2017 | Erfassen der Anforderungen | Samuel Egger |
| X0.3 | 12.11.2017 | Überarbeitung und Korrekturen | Rolf Zurbrügg |
| X0.4 | 12.11.2017 | Überarbeitung diverser Punkte | Rolf Zurbrügg |
| X0.5 | 12.11.2017 | Erweiterung des Glossars und Literatur Verzeichnisses | Rolf Zurbrügg |