|  |
| --- |
| 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

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

Erklärung von Fachbegriffen

# Abbildungsverzeichnis

**No table of figures entries found.**

# Tabellenverzeichnis

Tabelle 1: Funktionale Anforderungen 4

Tabelle 2: Technische Anforderungen 4

Tabelle 3: Qualitätsanforderungen 4

# Literaturverzeichnis

*Script.* 02 10 2017. https://en.bitcoin.it/wiki/Script.

# 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 |