



EX<sup>change</sup>  
tensions

## KONSOLIDIERUNG.

Tallinn, Estland. (E-Residenz)

# Weißbuch.

Version 1.0.0

Dezember 2020.

COINsolidation.org ist ein eingetragenes Warenzeichen von COINsolidation International, unter freier und kommerzieller Nutzungslizenz. Nutzungsbedingungen unter:

[www.Coinsolidation.org](http://www.Coinsolidation.org)

COINsolidation International fusionierte mit [www.OpenQbit.com](http://www.OpenQbit.com) für eine auf der Quantenmechanik basierende Technologiekooperation (Quantum Security & Quantum Computing). Diese Fusion ermöglicht die Nutzung, gemeinsame Nutzung und Neugestaltung der von OpenQbit Inc. entwickelten Technologie. (Estland, E-Residency)

## Inhalt

1. Einleitung.....	3
2. Sicherheitsquanten-Computing. ....	6
3. Erstellung eines "Hardware"-Gerätes eines QRNG (Quantum Random Number Generator)..	11
4. Was ist der Quantennachweis (PQu)? .....	17
5. Algorithmus für die Erstellung einer konsolidierten universellen Adresse (CUA) .....	20
6. Algorithmus für doppelte konsolidierte Adressen (DAC) und (HAC). ....	21
Projekt und Lösung durch COINsolidation. ....	23
7. Erstellung der App CUA (Consolidated Universal Address) in 15 Minuten.....	25
8. Erstellen Sie Ihre Ethereum-Krypto-Währungsumrechnung auf Android in nur 15 Minuten. .	29
9. Fahrplan COINsolidation. ....	32
10. Münzverdichtungsmarke (CUAG) - ICO-VERTEILUNGSPLAN.....	33
11. Allgemeine Merkmale des COINsolidation-Tokens:.....	34
12. Grundlegende Konzepte, die in Blockchain-Plattformen angewendet werden. ....	35
13. Was ist blockweise Programmierung? .....	38
14. Anhang "Code für CUA-Algorithmus". ....	38
15. Begriffe.....	38

## 1. Einleitung.

Gegenwärtig sind Fusionen auf dem neuesten Stand, sei es für ein wirtschaftliches, technologisches oder Marktgut.

Wir stellen das erste Modell der Kryptofusion oder Krypto-Token vor, das eine Sicherung zwischen zwei Kryptomonaden, Token oder einer Mischung aus diesen bietet, basierend auf einem Algorithmus zur Erstellung einer konsolidierten Adresse, die in der COINsolidation-Umgebung verwendet und generiert wird.

Wir haben drei Arten von konsolidierten Adressen erstellt.

Die **CUA** (Consolidated Universal Address) wird verwendet, um ein neues (aktives) Token zu konsolidieren und zu erstellen, das vom Benutzer verwendet wird. Die Kombination kann aus drei Typen bestehen: Cryptocurrency-Cryptocurrency, Cryptocurrency-Token oder Token-Token. Im Falle der ZBV wird sie durch eine Token-Token-Beziehung gebildet.

Die **HAC** (Hibric Address Consolidated) wird verwendet, wenn wir eine Adresse bezüglich einer Kryptowährung und/oder eines Tokens und eine normale Adresse für den Vermögenstransfer konsolidieren müssen.

Der **DAC** (Dual Address Consolidated) dient dazu, zwei normale Adressen aus derselben Blockkette oder aus zwei verschiedenen Technologien zu verwalten und zu konsolidieren.

Beginnen wir mit der Betrachtung der Vorteile des CUA.

Eine CUA-Adresse besteht aus der Adresse des COINsolidation-Tokens (statische Adresse) und einem zusätzlichen Token, das als "Colored Coin" bezeichnet wird (variable Adresse). In diesem Fall können wir sehen, dass die ZBV-Adressen immer aus Adressen irgendeiner Art von Vermögenskombination (Kryptosolids oder Token) gebildet werden.

In unserem Fall, wenn wir das COINsolidation-Token und ein OAP-Token konsolidieren, werden wir es als "CUA-Genesis" oder **CUAG (Consolidated Universal Address Genesis)** kennen.

El token COINsolidation esta creado en el *blockchain Ethereum* y usa el standard ERC20 (Ethereum Anfrage für Kommentare 20).

Die Wertmarke "Colored Coin" basiert auf der *Bitcoin-Blockkette* und wird von dieser erzeugt und verwendet den Open Assest Protocol (**OAP**)-Standard.

Beginnen wir mit einem Überblick über das Potenzial und den Nutzen der Konsolidierung von Adressen.

- I. Für die Benutzer, die eine ZBV erstellen, wird es möglich sein, ein Token (**OAP**) zu erstellen, das von dem Benutzer, der die ZBV erstellt hat, angepasst werden kann. Der Benutzer wird die Möglichkeit haben, sein eigenes Token oder Krypto aktiv zu haben, so dass er es bei der Erstellung, Unterstützung oder Erweiterung seiner Firma(n) verwenden kann, so dass er auf einfache und unkomplizierte Weise einen Vorteil in der Welt der Krypto-Token hat.
- II. Unternehmen, die eine ZBV erstellen, verfügen möglicherweise über eine Wertmarke (**OAP**), die sie zur Wertschöpfung in ihrer Lieferkette oder zur Verwendung des Vermögenswerts in Liquiditätstransaktionen auf der Grundlage der wirtschaftlichen Unterstützung ihrer Unternehmensvermögenswerte und -verbindlichkeiten verwenden können.
- III. Für bestehende Kryptomonaden und Token können sie durch die Erstellung einer ZBV ihre Adresse verwenden, die ihren Vermögenswert identifiziert, und durch die Konsolidierung mit dem Token (**OAP**) können sie ihre Nachfrage steigern, indem sie ihren derzeitigen und zukünftigen Investoren ein eigenes Token für ihre Nutzer anbieten.

Beispiel für **CUAG**, wir haben die jeweiligen Adressen von zwei verschiedenen Blockchain:

Adresse Bitcoin- Token - (OAP).

**akXma4vqxvmEqnVAKSM953wYsnjNBhN3GM7**

Anschrift Ethereum - Token COINsolidation - (ERC20).

**0x8390f8abb8fd8ad3bf8457db59f2ed75e015d303**

Durch Anwendung eines Algorithmus zur Konsolidierung der bisherigen Adressen erhalten wir die ZBV-Adresse.

**cua50d0615d303k8X3m9a04fv8qaxbvb8Efqdn8VaAdK3SbMf985435w7Ydsbn5j9NfB2heNd37G5Me70**

\* Weitere Einzelheiten zum Algorithmus finden Sie im Abschnitt 7.- "Algorithmus für die Erstellung einer konsolidierten universellen Adresse".

Wir haben als Ergebnis eine einzige Richtung, die zwei verschiedene Technologien aus zwei verschiedenen Richtungen in einer einzigen Richtung konsolidiert darstellt.

Wir spiegeln dies auf dem Gebiet der Rentabilität und der finanziellen Expansion auf einfache und unkomplizierte Weise wider, indem wir in eine der Wertmarken investieren, die unsere ZBV integriert. Sie erhalten sofort eine Wertmarke auf der Basis der Bitcoin-Blockkette (OAP).

Lassen Sie uns nun zwei Richtlinien betrachten, die wir auch in COINsolidation für die Welt der Kryptomontagen und/oder Tokens vorgeschlagen haben.

COINsolidation Token ist das Projekt zur Konsolidierung von Adressen und zur sofortigen Unterstützung bei der Beschaffung eines benutzerdefinierten Tokens, das im Wachstum jedes Benutzers in der Welt der Kryptomontagen verwendet werden kann. Das Projekt wurde 2018 mit einer Gruppe von Ingenieuren und Finanziers ins Leben gerufen, die daran interessiert sind, den Finanz- und den Technologiesektor zu verschmelzen, Investitionsfonds zu nutzen und innovative Technologien wie Quantencomputer zu nutzen, um die Sicherheit von Vermögenswerten zu gewährleisten, sowie den Zweck zu verfolgen, Werkzeuge zu nutzen, die für jedermann zugänglich sind.

Nach einer Evaluierung verschiedener Entwicklungsmöglichkeiten haben wir uns für die visuelle Blockly-Programmierungsmethodik entschieden. Diese Methodik basiert auf der Verwendung von Erweiterungen oder Modulen (Programme in der Programmiersprache Java) mit einfachen, aber leistungsfähigen Funktionalitäten, um das Geschäft mit Krypto-Assets für jede Person zu erweitern:

- ✓ a.- Unmittelbarer finanzieller ROI für Nutzer, Investoren und Vermögenswerte durch die Möglichkeit, einen immateriellen Vermögenswert (persönliche Tokens) zur ausschließlichen Nutzung durch den Schöpfer und Nutzer des (ZBV) zu schaffen
- ✓ b.- Wir nutzen die Vorteile der Verbindung zweier Blockketten nach Wahl des Benutzers, um aktuelle und zukünftige Investitionen in den kryptoaktiven Markt mit Hilfe des (CUA) zu steigern.
- ✓ c.- Erleichterung der Verwaltung separater Adressen durch deren Konsolidierung in (DAC).
- ✓ d.- Sicherheit auf der Grundlage von Quantum Computing erstellen und verwenden.

Die Herausforderung begann in der Schaffung der Technologie der Erweiterungen ausreichend modular in der Funktionalität und "Größe" dieser letzten war die Herausforderung des Entwicklungsteams von COINsolidation seit den Erweiterungen, die in der Blockly-Methodik und Systeme dieser Art (AppInventor, AppyBuilder, Thunkable, Kondular verwendet werden, usw.) sind in der Regel Erweiterungen (Programme) erstellt, die nicht mehr als 100k - 300k Bytes, mit den Einschränkungen, die in ihrer Größe die Aufgabe der Erstellung von Erweiterungen für die Verwendung in der aktuellen Blockchain haben, waren praktisch unmöglich aufgrund der Bibliotheken, die in ihrer Erstellung verwendet werden, überschreiten zwischen 10MB und 35MB diese Größen für die aktuellen Werkzeuge Blockly Systeme sind nicht funktionsfähig, sie zu verwenden.

Das Team musste Programmierungsmethodik und Bibliotheken erstellen, anpassen und minimieren, um die Erweiterungen mit der optimalen Funktionalität, Sicherheit und Größe zu erhalten.

Nach fast zwei Jahren der Entwicklung und Erprobung haben wir die erste Blockkette "Beta" mit Erweiterungen für Blockly einschließlich des Konsensalgorithmus "Proof of Quantum" mit Quantensicherheit für den Krytomoney-Austausch fertiggestellt.

Derzeit verfügen wir über eine proprietäre Blockkette, die für "Beta"-Tests freigegeben wurde, und bis Ende 2021 werden wir die Produktionsversion zur Informationsverteilung freigegeben. Gegenwärtig basiert unser COINsolidation Token auf den Blockketten Ethereum und Bitcoin, letztere für die Erstellung benutzerdefinierter Token für Benutzer.

## 2. Sicherheitsquanten-Computing.

### Wie funktioniert das Quantencomputing? <sup>(2)</sup>

Der digitale Wandel bringt schneller als je zuvor Veränderungen in der Welt mit sich. Würden Sie glauben, dass das digitale Zeitalter kurz vor dem Ende steht? **Die digitale Kompetenz** wurde bereits als ein Bereich identifiziert, in dem offenes Wissen und zugängliche Möglichkeiten zum Erlernen von Technologie dringend erforderlich sind, um Lücken in der sozialen und wirtschaftlichen Entwicklung zu schließen. Das Lernen von den Schlüsselkonzepten des digitalen Zeitalters wird mit der bevorstehenden Ankunft einer weiteren neuen technologischen Welle, die in der Lage ist, bestehende Modelle mit erstaunlicher Geschwindigkeit und Kraft zu transformieren, noch kritischer werden: die **Quantentechnologien**.

In diesem Artikel vergleichen wir die grundlegenden Konzepte des traditionellen Rechnens und des Quantencomputings; und wir beginnen auch mit der Untersuchung ihrer Anwendung in anderen verwandten Bereichen.

Was sind Quantentechnologien?

Im Laufe der Geschichte hat der Mensch die Technik entwickelt, da er durch die Wissenschaft verstanden hat, wie die Natur funktioniert. Zwischen 1900 und 1930 führte die Untersuchung einiger physikalischer Phänomene, die noch nicht gut verstanden waren, zu einer neuen physikalischen Theorie, der **Quantenmechanik**. Diese Theorie beschreibt und erklärt die Funktionsweise der mikroskopischen Welt, dem natürlichen Lebensraum von Molekülen, Atomen oder Elektronen. Dank dieser Theorie war es nicht nur möglich, diese Phänomene zu erklären, sondern auch zu verstehen, dass die subatomare Realität auf eine völlig kontraintuitive, fast magische Weise funktioniert und dass in der mikroskopischen Welt Ereignisse stattfinden, die in der makroskopischen Welt nicht vorkommen.


Zu diesen **Quanteneigenschaften** gehören Quantenüberlagerung, Quantenverschränkung und Quantenteleportation.

- **Die Quantenüberlagerung** beschreibt, wie sich ein Teilchen gleichzeitig in verschiedenen Zuständen befinden kann.
- **Die Quantenverschränkung** beschreibt, wie zwei beliebig weit voneinander entfernte Teilchen so korreliert werden können, dass bei der Wechselwirkung mit dem einen das andere sich dessen bewusst wird.
- **Die Quantenteleportation** nutzt die Quantenverschränkung, um Informationen von einem Ort im Raum zu einem anderen zu senden, ohne ihn durchqueren zu müssen.

Quantentechnologien basieren auf diesen Quanteneigenschaften der subatomaren Natur.

In diesem Fall erlaubt uns heute das Verständnis der mikroskopischen Welt durch die Quantenmechanik, Technologien zu erfinden und zu entwerfen, die das Leben der Menschen verbessern können. Es gibt viele und sehr unterschiedliche Technologien, die sich Quantenphänomene zunutze machen, und einige von ihnen, wie z.B. Laser oder Magnetresonanztomographie (MRI), begleiten uns seit mehr als einem halben Jahrhundert. Derzeit erleben wir jedoch eine technologische Revolution in Bereichen wie Quantencomputer, Quanteninformation, Quantensimulation, Quantenoptik, Quantenmetrologie, Quantenuhren oder Quantensensoren.

Was ist Quanteninformatik? Zuerst müssen Sie das klassische Rechnen verstehen.



**FIGURA 1.**  
Ejemplos de caracteres en lenguaje binario.

Caracter	Bits
7	111
A	01000001
\$	00100100
:)	0011101000101001

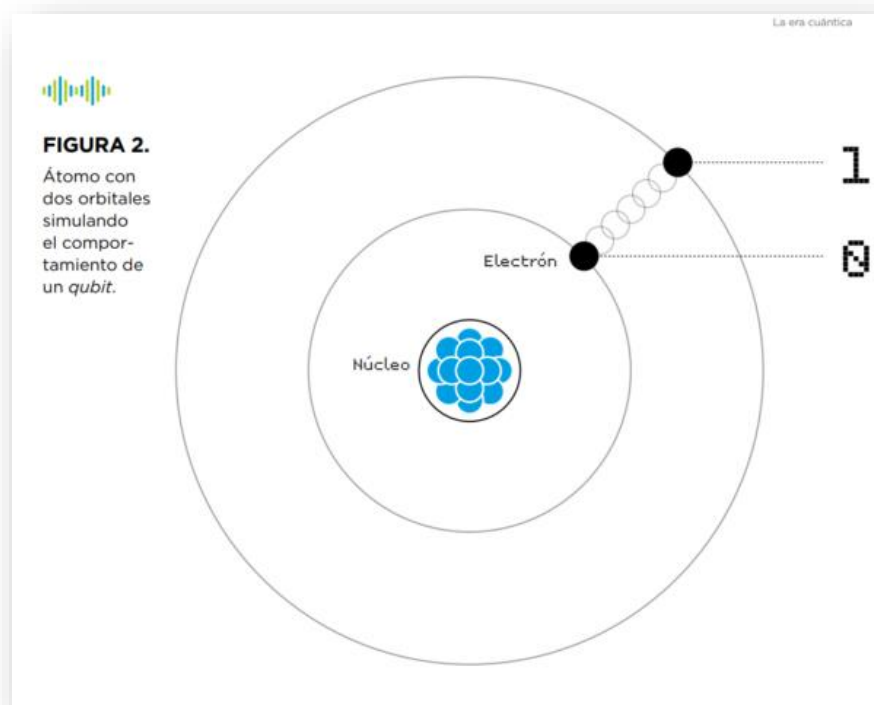
Um zu verstehen, wie Quantencomputer funktionieren, ist es sinnvoll, zunächst zu erklären, wie die Computer funktionieren, die wir täglich benutzen und die wir in diesem Dokument als digitale oder klassische Computer bezeichnen. Diese, wie auch die übrigen elektronischen Geräte wie Tablets oder Mobiltelefone, verwenden Bits als die grundlegenden Speichereinheiten. Dies bedeutet, dass Programme und Anwendungen in Bits kodiert sind, d.h. in binärer Sprache mit Nullen und Einsen. Jedes Mal, wenn wir mit einem dieser Geräte interagieren, zum Beispiel durch Drücken einer Taste auf der Tastatur, werden im Computer Zeichenketten aus Nullen und Einsen erzeugt, zerstört und/oder verändert.

Die interessante Frage ist, was diese Nullen und Einsen physisch im Computer sind. Der Null- und der Eins-Zustand entsprechen elektrischem Strom, der durch mikroskopisch kleine Teile, die als Transistoren bezeichnet werden und als Schalter fungieren, zirkuliert oder nicht. Wenn kein Strom fließt, ist der Transistor "aus" und entspricht Bit 0, und wenn er fließt, ist er "an" und entspricht Bit 1.

Einfacher gesagt, es ist so, als ob die Bits 0 und 1 Löchern entsprechen, so dass ein leeres Loch ein Bit 0 und ein von einem Elektron besetztes Loch ein Bit 1 ist. Deshalb werden diese Geräte als Elektronik bezeichnet. Als Beispiel zeigt Abbildung 1 die binäre Schrift einiger Zeichen. Da wir nun eine Vorstellung davon haben, wie die heutigen Computer funktionieren, wollen wir versuchen zu verstehen, wie die Quanten funktionieren.

### Von Bits zu Qubits

Die grundlegende Informationseinheit im Quantencomputing ist das Quantenbit oder Qubit. Qubits sind per Definition zweistufige Quantensysteme - wir werden hier Beispiele sehen -, die sich wie Bits auf dem niedrigen Niveau befinden können, was einem Zustand niedriger Anregung oder Energie entspricht, der als 0 definiert ist, oder auf dem hohen Niveau, was einem Zustand höherer Anregung entspricht oder als 1 definiert ist. Allerdings, und hier liegt der grundlegende Unterschied zum klassischen Rechnen, können Qubits auch in jedem der unendlichen Zwischenzustände zwischen 0 und 1 liegen, wie z.B. in einem Zustand, der halb 0 und halb 1 oder drei Viertel von 0 und ein Viertel von 1 ist.





## Quantenalgorithmen, exponentiell leistungsfähigeres und effizienteres Rechnen

Der Zweck von Quantencomputern ist es, diese Quanteneigenschaften der *Qubits* als Quantensysteme, die sie sind, zu nutzen, um Quantenalgorithmen auszuführen, die Überlappung und Verschachtelung verwenden, um eine viel höhere Rechenleistung als die Klassiker zu erreichen. Es ist wichtig, darauf hinzuweisen, dass der wirkliche Paradigmenwechsel nicht darin besteht, das Gleiche zu tun wie digitale oder klassische Computer - die aktuellen -, sondern schneller, wie in vielen Artikeln zu lesen ist, sondern dass Quantenalgorithmen es erlauben, bestimmte Operationen auf eine völlig andere Art und Weise durchzuführen, die sich in vielen Fällen als effizienter erweist - d.h. in viel weniger Zeit oder mit viel weniger Rechenressourcen -.

Schauen wir uns ein konkretes Beispiel dafür an, was dies beinhaltet. Stellen wir uns vor, wir sind in Bogotá, und wir wollen die beste Route nach Lima aus einer Million Möglichkeiten kennen, um dorthin zu gelangen ( $N=1.000.000$ ). Um mit Hilfe von Computern den optimalen Weg zu finden, müssen wir 1.000.000 Optionen digitalisieren, was bedeutet, sie für den klassischen Computer in Bitsprache und für den Quantencomputer in *Qubits* zu übersetzen. Während ein klassischer Computer einen nach dem anderen alle Pfade analysieren müsste, bis er den gewünschten gefunden hat, nutzt ein Quantencomputer den als Quantenparallelität bekannten Prozess, der es ihm erlaubt, alle Pfade auf einmal zu berücksichtigen. Dies bedeutet, dass, während der klassische Computer die Reihenfolge von  $N/2$  Schritten oder Iterationen benötigt, d.h. 500.000 Versuche, der Quantencomputer den optimalen Pfad nach nur  $\sqrt{N}$  Operationen auf dem Register findet, d.h. nach 1.000 Versuchen.

Im vorherigen Fall ist der Vorteil quadratisch, in anderen Fällen ist er sogar exponentiell, was bedeutet, dass wir mit  $n$  *Qubits* eine Rechenkapazität von  $2^n$  Bits erreichen können. Um dies zu veranschaulichen, ist es üblich zu zählen, dass wir mit etwa 270 *Qubits* in einem Quantencomputer mehr Basiszustände - mehr unterschiedliche und gleichzeitige Zeichenketten - haben könnten als die Anzahl der Atome im Universum, die auf etwa  $10^{80}$  geschätzt wird. Ein weiteres Beispiel ist, dass man schätzt, dass wir mit einem Quantencomputer mit 2000 bis 2500 *Qubits* praktisch die gesamte heute verwendete Kryptographie (die sogenannte Public-Key-Kryptographie) brechen könnten.

Warum ist es wichtig, etwas über Quantentechnologie zu wissen?

Wir befinden uns in einem Moment der digitalen Transformation, in dem verschiedene aufkommende Technologien wie Blockchain, künstliche Intelligenz, Drohnen, Internet der Dinge, virtuelle Realität, 5G, 3D-Drucker, Roboter oder autonome Fahrzeuge immer mehr Präsenz in verschiedenen Bereichen und Sektoren haben. Diese Technologien, die dazu berufen sind, die Lebensqualität des Menschen zu verbessern, die Entwicklung zu beschleunigen und soziale Auswirkungen zu erzeugen, schreiten heute parallel voran. Nur

selten sehen wir Unternehmen, die Produkte entwickeln, die Kombinationen aus zwei oder mehr dieser Technologien nutzen, wie z.B. Blockchain und IoT oder Drohnen und künstliche Intelligenz. Obwohl sie dazu bestimmt sind, zu konvergieren und damit eine exponentiell größere Wirkung zu erzielen, ist die Konvergenz aufgrund des Anfangsstadiums der Entwicklung, in dem sie sich befinden, und des Mangels an Entwicklern und Personen mit technischem Profil noch immer eine offene Aufgabe.

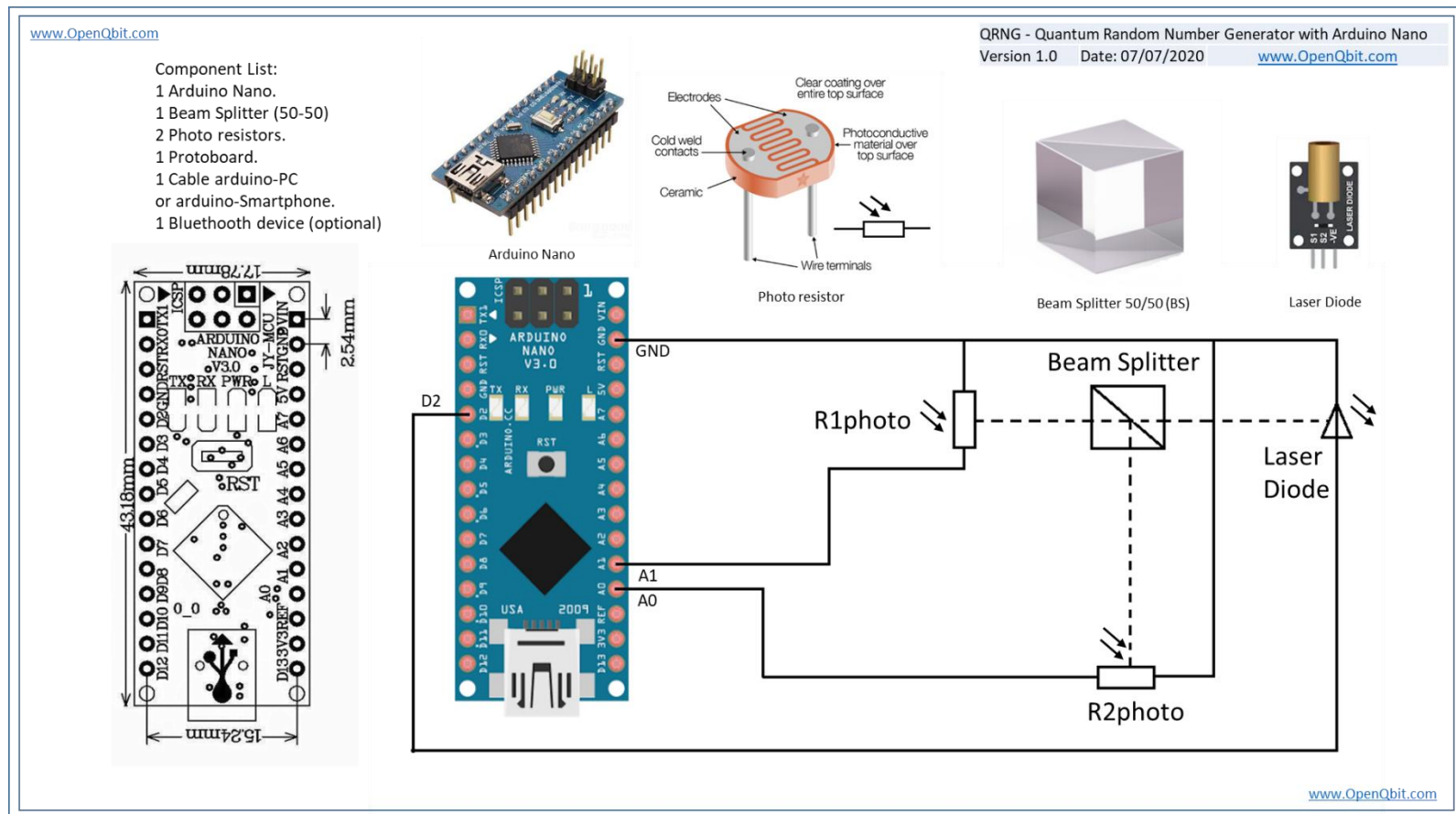
Aufgrund ihres disruptiven Potenzials wird erwartet, dass die Quantentechnologien nicht nur mit all diesen neuen Technologien konvergieren, sondern einen Querschnittseinfluss auf praktisch alle von ihnen haben werden. Die Quanteninformatik wird die Authentifizierung, den Austausch und die sichere Speicherung von Daten bedrohen, was einen großen Einfluss auf jene Technologien hat, bei denen die Kryptographie eine wichtigere Rolle spielt, wie z.B. Cybersicherheit oder Blockchain, und einen geringen negativen Einfluss hat, aber auch in Technologien wie 5G, IoT oder Drohnen in Betracht gezogen werden muss.

### **Wollen Sie sich im Quantencomputing üben?**

Dutzende von Quantencomputer-Simulatoren sind bereits im Netz verfügbar, wobei verschiedene Programmiersprachen wie C, C++, Java, Matlab, Maxima, Python oder Octave bereits im Einsatz sind. Auch neue Sprachen wie Q#, die von Microsoft eingeführt wurden. Sie können eine virtuelle Quantenmaschine über Plattformen wie IBM und Rigetti erforschen und mit ihr spielen.

### 3. Erstellung eines "Hardware"-Gerätes eines QRNG (Quantum Random Number Generator).

Wir werden nun ein physisches "Hardware"-Gerät zur Generierung von Quanten-Zufallszahlen (QRNG) mit kostengünstigen Komponenten erstellen, das sich zu Hause leicht zusammenbauen lässt und etwa 35 USD kostet.



## QRNGv1.0.ino

Software  
Program to arduino nano.

```
/* OpenQbitQRNG Firmware V1.0
 *Author: Guillermo Vidal
 *Copyright © 2020 OpenQbit, Inc.
 *License: MIT
 */
```

```
int triggerQ = 2; // This pin will pulse our quantum circuit
int QuA0Pin = A0; // This pin measures the horizontal polarized photons
int QuA1Pin = A1; // This pin measures the vertically polarized photons
float Qu0 = 0;
float Qu1 = 0;
```

```
void setup() {
  // Just setting up triggerPin and serial connection
  pinMode(triggerQ, OUTPUT); // sets the digital pin 2 as output
  Serial.begin(9600);
}
```

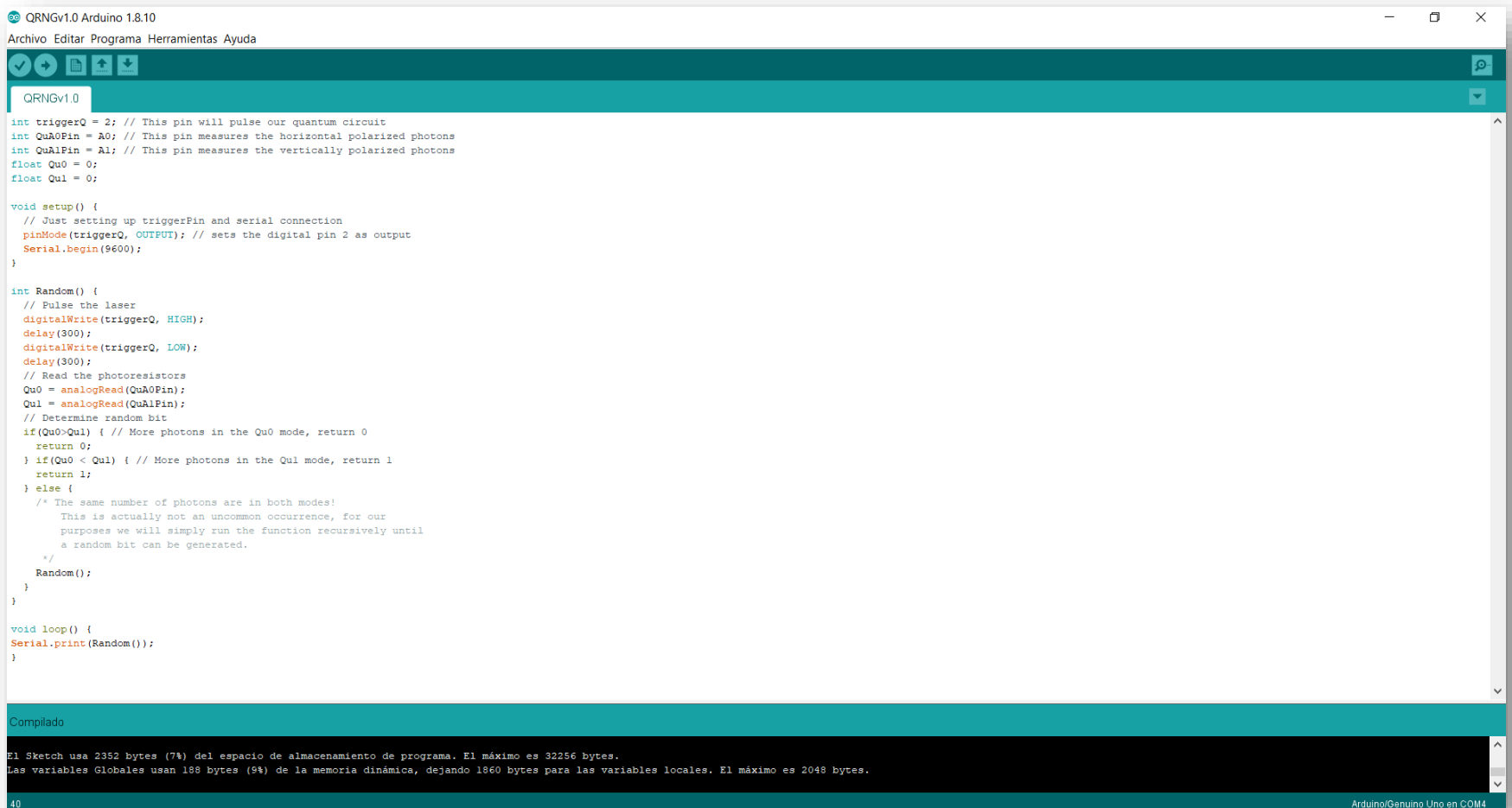
```
int Random() {
  // Pulse the laser
  digitalWrite(triggerQ, HIGH);
  delay(300);
  digitalWrite(triggerQ, LOW);
  delay(300);
  // Read the photoresistors
  Qu0 = analogRead(QuA0Pin);
  Qu1 = analogRead(QuA1Pin);
  // Determine random bit
  if(Qu0>Qu1) { // More photons in the Qu0 mode, return 0
    return 0;
  } if(Qu0 < Qu1) { // More photons in the Qu1 mode, return 1
    return 1;
  } else {
    /* The same number of photons are in both modes!
     This is actually not an uncommon occurrence, for our
     purposes we will simply run the function recursively until
     a random bit can be generated.
    */
    Random();
  }
}
```

```
void loop() {
  Serial.print(Random());
}
```

## Output console

0010110101011110101011010.....

Kompilieren des Programms QRNGv10.ino und Hochladen auf arduino nano....



The screenshot shows the Arduino IDE interface with the file 'QRNGv1.0' open. The code is written in C++ and implements a quantum random number generator. It uses two analog pins (A0 and A1) to measure polarized photons and a digital pin (2) as an output. The 'Random()' function pulses the laser and reads the sensors to generate a random bit. The 'loop()' function prints the generated random bit to the serial monitor.

```
int triggerQ = 2; // This pin will pulse our quantum circuit
int QuA0Pin = A0; // This pin measures the horizontal polarized photons
int QuA1Pin = A1; // This pin measures the vertically polarized photons
float Qu0 = 0;
float Qu1 = 0;

void setup() {
  // Just setting up triggerPin and serial connection
  pinMode(triggerQ, OUTPUT); // sets the digital pin 2 as output
  Serial.begin(9600);
}

int Random() {
  // Pulse the laser
  digitalWrite(triggerQ, HIGH);
  delay(300);
  digitalWrite(triggerQ, LOW);
  delay(300);
  // Read the photoresistors
  Qu0 = analogRead(QuA0Pin);
  Qu1 = analogRead(QuA1Pin);
  // Determine random bit
  if(Qu0>Qu1) { // More photons in the Qu0 mode, return 0
    return 0;
  } if(Qu0 < Qu1) { // More photons in the Qu1 mode, return 1
    return 1;
  } else {
    /* The same number of photons are in both modes!
       This is actually not an uncommon occurrence, for our
       purposes we will simply run the function recursively until
       a random bit can be generated.
    */
    Random();
  }
}

void loop() {
  Serial.print(Random());
}
```

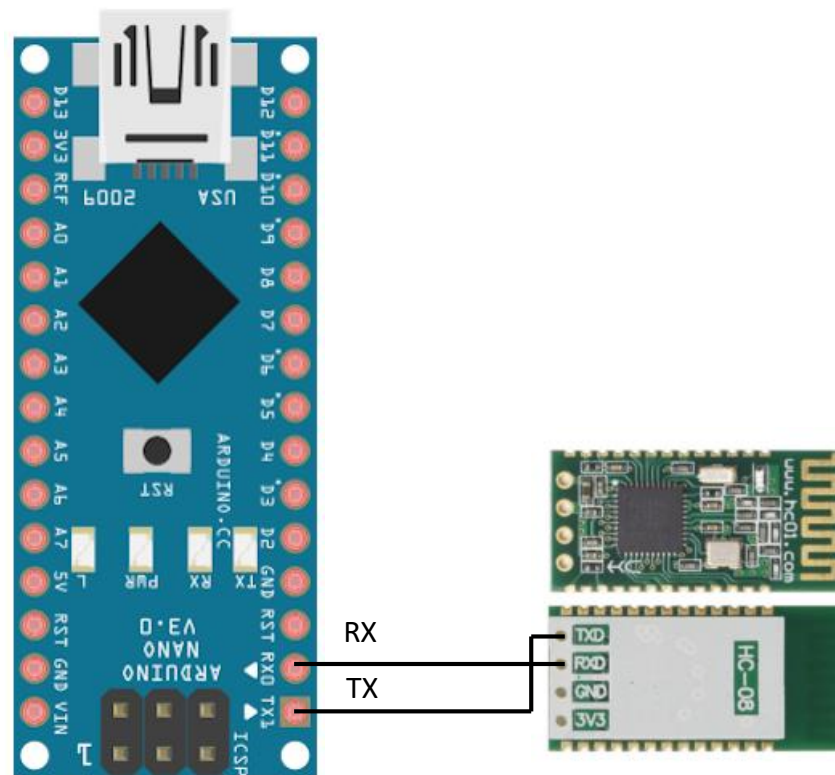
**Compilado**

El Sketch usa 2352 bytes (7%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.  
Las variables Globales usan 188 bytes (9%) de la memoria dinámica, dejando 1860 bytes para las variables locales. El máximo es 2048 bytes.

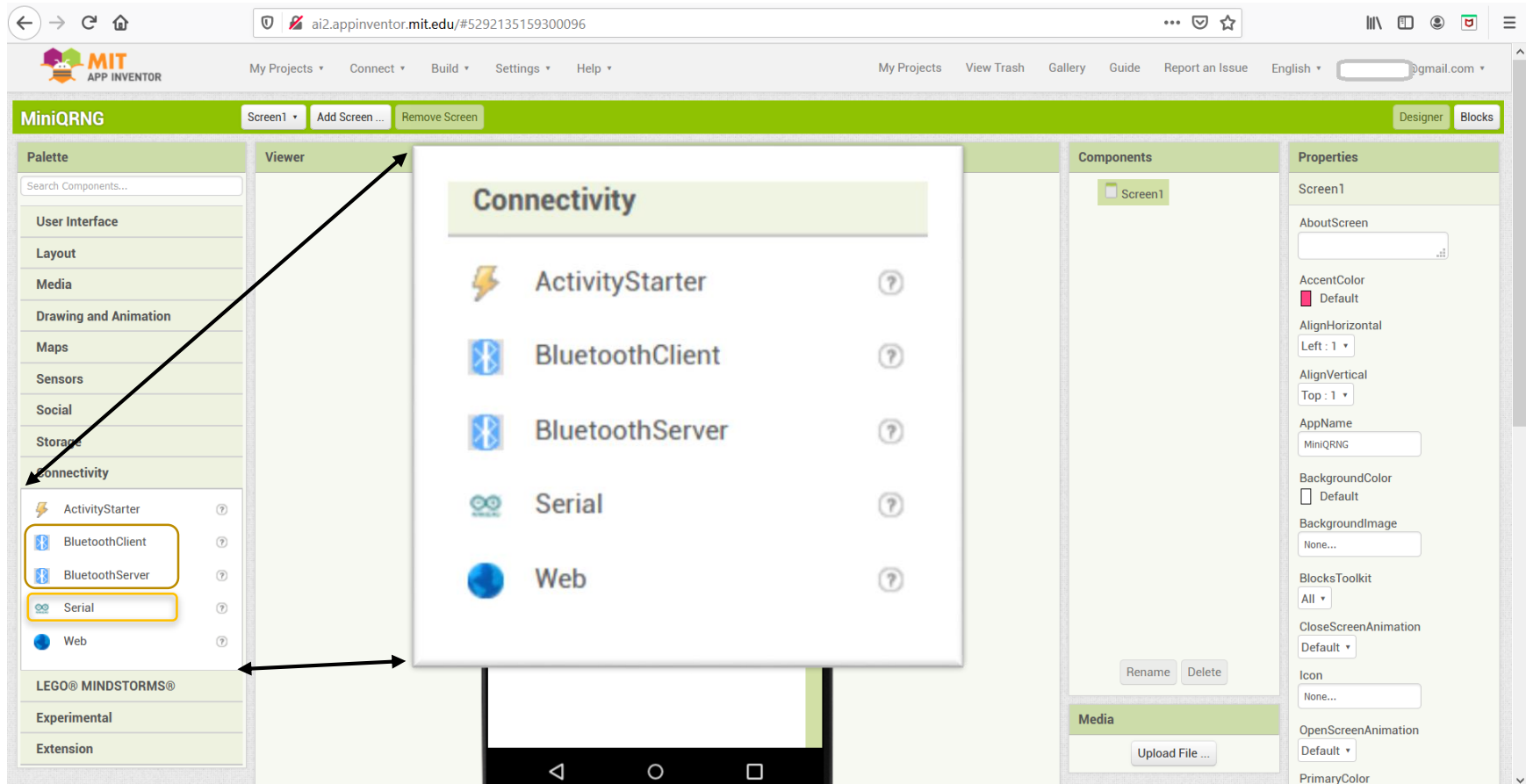
40 Arduino/Genuino Uno en COM4

Es gibt zwei Möglichkeiten, mit dem anstrengenden Nano zu kommunizieren, eine über die serielle Schnittstelle und die andere über eine Bluetooth-Verbindung.

Da die Bluetooth-Verbindung sehr einfach ist, müssen wir nur das HC-08-Modul oder ein ähnliches Modul kaufen und es wie folgt anschließen:



Die folgenden seriellen oder Bluetooth-Komponenten können verwendet werden, um App Inventor mit Arduino zu verbinden:





Jetzt kompiliert und geladen das Programm QRNGv10.ino nur fehlende Kommunikation mit dem mühsamen Nano, um die Daten (Quantenzufallszahlen) zu speichern diese werden im Binärformat sein, jedoch können die erhaltenen Daten leicht in ein anderes Format wie hexadezimal oder dezimal je nach der endgültigen Anforderung übergeben werden.

Um schließlich ein Beispiel zu sehen, wie die serielle oder Bluetooth-Verbindung funktioniert, finden Sie hier einige Referenz-Links.

Denken Sie daran, dass alles durch blockweise Programmierung mit App Inventor getestet werden kann, da dieser bereits Blöcke für die Kommunikation mit dem seriellen Arduino-System oder einem anderen blockweisen System hat, das über ein ähnliches Bluetooth online sein kann.

[http://kio4.com/erfinder/9A0\\_bluetooth\\_RXTX.htm](http://kio4.com/erfinder/9A0_bluetooth_RXTX.htm)

<http://kio4.com/erfinder/index.htm#bluetooth>

<https://community.appinventor.mit.edu/>

Überprüfung des gesamten Projekts der Gestaltung und Verwendung von QRNG-Erweiterungen (Quantum Random Number Generator). Lesen Sie das Benutzerhandbuch unter:

<https://github.com/COINsolidation/UserGuide>





## 4. Was ist der Quantennachweis (PQu)?

PoQu. - "Proof of Quantum" ist ein Konsens-Algorithmus, der für Mini BlocklyChain und COINsolidation entwickelt wurde. Dieser Test ist eine Variante des Proof of Work (PoW), der wie folgt funktioniert.

Der Test of Quantum (PoQu) wird beim Start mit demselben Algorithmus ausgeführt wie der "Test of Work" (PoW), der darauf basiert, den Prozessor des Geräts (PC, Server, Tablet oder Mobiltelefon) in Gang zu setzen, um eine Zeichenfolge zu erhalten, die ein mathematisches Rätsel namens "Hash" darstellt.

Denken Sie daran, dass ein "Hash" ein Algorithmus oder ein mathematischer Prozess ist, der uns beim Einfügen einer Phrase oder einer Art digitaler Information wie Textdateien, Programme, Bilder, Videos, Töne oder anderer unterschiedlicher digitaler Informationen als Ergebnis ein alphanumerisches Zeichen gibt, das die digitale Signatur repräsentiert, die sie in einer einzigartigen und nicht wiederholbaren Weise der Daten darstellt, der Hash-Algorithmus ist unidirektional, d.h. wenn Sie Daten eingeben, um die Signatur "Hash" zu erhalten, kann der umgekehrte Prozess nicht durchgeführt werden, da wir bei einer Signatur "Hash" nicht wissen können, welche Informationen erhalten wurden. Diese Eigenschaft gibt uns einen Sicherheitsvorteil bei der Verarbeitung der Informationen, die wir über das Internet senden. Wie funktioniert das? Stellen Sie sich vor, Sie senden jede Art von Information über unsichere Kanäle und begleiten sie mit ihrem jeweiligen "Quell-Hash", der Empfänger kann beim Empfang der Information den "Hash" der empfangenen Information erhalten, wir nennen ihn "Ziel-Hash" und überprüfen ihn mit dem "Quell-Hash", wenn beide "Hashes" gleich sind, können wir bestätigen, dass die Information in dem Kanal, der gesendet wurde, nicht verändert wurde, ist nur ein Beispiel dafür, wo diese Art von Informationssicherheitsverfahren derzeit verwendet wird.

Gegenwärtig gibt es verschiedene Arten von Algorithmen oder Hash-Verfahren, die sich in der Sicherheitsstufe unterscheiden. Die am häufigsten verwendeten oder bekannten sind: MD5, SHA256 und SHA512.

Beispiel für SHA256:

Wir haben eine Kette oder einen Satz wie folgt: "Mini BlocklyChain ist modular aufgebaut.



Wenn wir einen Hash vom Typ SHA256 auf die vorherige Zeichenfolge anwenden, erhalten wir den nächsten Hash.

**f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db8**

Die obige alphanumerische Zeichenfolge ist die Signatur, die den Satz im obigen Beispiel darstellt

Für weitere Beispiele können wir die Website im Internet nutzen:

<https://emn178.github.io/online-tools/sha256.html>

Im Falle des "Test Work" (PoW)-Algorithmus arbeitet er mit Rechenleistung, um einen vordefinierten Hash zu erhalten.

Stellen wir uns vor, wir hätten den vorherigen "Hash", den wir von der Kette "Mini BlocklyChain ist modular" übernommen haben.

**f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db8**

Zu diesem "Hash" am Anfang setzen wir den Parameter der Schwierigkeit, der einfach darin besteht, Nullen "0" an den Anfang zu setzen, d.h. wenn wir sagen, dass die Schwierigkeit bei 4 liegt, wird sie "0000" + "Hash" haben, dann nennen wir sie "Samen-Hash".

**0000 f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db8**

Wenn wir nun berücksichtigen, dass wir die Eingabeinformation kennen, die die Zeichenkette ist: "Mini BlocklyChain ist modular", fügen wir am Ende der Zeichenkette eine Zahl beginnend bei Null "0" hinzu und nehmen den Hash heraus, den wir "hash nonce" nennen:

**f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db80**

Wir haben Haschisch nonce:

**7529f3ad273fc8a9eff12183f8d6f886821900750bb6b59c1504924dfd85a7c8**

Dann führen wir einen Vergleich der neuen "hash nonce" mit den "hash seed" durch, wenn sie gleich sind, gewinnt der Knoten, der zuerst die Gleichheit findet, die Ausführung der Verarbeitung der aktuellen Transaktion. Wie wir sehen können, basiert dieser Prozess auf der Wahrscheinlichkeit und der Rechenstärke des Geräts, das dem "Proof of Work"-Test eine Konsensgerechtigkeit für alle Knotenpunkte verleiht.

Wenn der "Samen-Hash" nicht mit dem "Hash-Nonce" übereinstimmt, wird die Schwierigkeit um eins erhöht und der "Hash-Nonce" wieder entfernt, die Zahl, die erhöht wird, wird als



"Nonce"-Zahl bezeichnet, sie wird mit dem "Samen-Hash" verglichen, bis sie übereinstimmen oder gleich sind.

Wie wir sehen können, ist die Zahl "nonce" oder Erhöhung diejenige, die dazu beiträgt, den "Hash" der Gleichheit zu erreichen.

Basierend auf dem "Test of Work"-Algorithmus (PoW) basiert der Quantentest-Algorithmus (PoQu) darauf, die Zahl "nonce" wie beim PoW zu erhalten und einen mindestdifficultegrad von 1 bis 5, dies dient nur dazu, das Recht des mobilen Geräts als Kandidat für den Konsens zu gewinnen.

Der Quantentest (PoQu), wird aktiviert, wenn das Mobiltelefon den minimalen PoW-Wert erreicht hat und den Pass gewinnt, um eine Wahrscheinlichkeitszahl im QRNG-System zu erhalten.

Der QRNG (Quantum Random Number Generator) ist ein Quantenzufallszahlengenerator. Dieses System basiert auf der Erzeugung echter Zufallszahlen auf der Grundlage der Quantenmechanik und ist heute das sicherste System zur Erzeugung solcher Zahlen. Für weitere Einzelheiten siehe "Sicherheit bei der Quantenberechnung" in Index 3.

COINsolidation kann sowohl minimale PoW- als auch PoQu-Konzessionsarten implementieren.

Der PoQu-Test basiert auf der Ermittlung der "Nonce"-Zahl. Diese Zahl wird im PoQu-Test als "Magic Number" bezeichnet, und damit bestätigt das Peer-to-Peer-System, ob die Zahl korrekt ist, und dann wird mit dem COINsolidation-QRNG-Serverpool eine Zufallszahl ermittelt. Diese Zufallszahl wird in allen Knoten registriert, es wird eine Liste erstellt, die  $((\text{Knotensumme} / 2)) + 1$  enthält, und aus dieser Liste wird derjenige mit der höchsten prozentualen Wahrscheinlichkeit als Gewinnerkandidat des Konsens (PoQu) ausgewählt, der die aktuelle Transaktionswarteschlange ausführt.

Der PoQu-Algorithmus verwendet auch Tests **des NIST** (National Institute of Standards and Technology), um uns zu versichern, dass die Zufallszahlen im QRNG wirklich Zufallszahlen sind.

<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-22r1a.pdf>

In COINsolidation haben wir einen Block für PoW und einen Block für PoQu implementiert. Diese Blöcke verwenden einen Hash-Typ: SHA256 für die freie Nutzung, für die kommerzielle Nutzung haben wir einen SHA512 und andere Hashes nach Bedarf.

Weitere Einzelheiten zum Konzept von HASH siehe:

[https://es.wikipedia.org/wiki/Funcion\\_hash](https://es.wikipedia.org/wiki/Funcion_hash)



HINWEIS: Der in Mobiltelefonen verwendete Test of Work (PoW) kann nur einen Schwierigkeitsgrad von maximal 5 verwenden, da die mathematische Verarbeitung dieser Geräte nicht dediziert wie bei Servern oder PCs erfolgt. Wir verwenden den PoW-Algorithmus nur, um die Gelegenheit zu erhalten, Ihren Pass oder Ihre Erlaubnis zum Eintritt in das System des Quantenzufallszahlengenerators (QRNG) und damit zur Ausführung des Quantenzufallszahlengenerators (PoQu) zu erhalten. Siehe Verwendung von (PoQu) in Mini BlocklyChain:

<https://github.com/openqbit-diy/MiniBlocklyChain>

## 5. Algorithmus für die Erstellung einer konsolidierten universellen Adresse (CUA)

Unsere CUA-Adressen (CUA = Consolidated Universal Address) werden nach folgendem Algorithmus erstellt:

Schritt 1 - Identifikatoren werden aus den jeweiligen Adressen entfernt, sie sind die alphanumerischen Zeichen, die die Adresse identifizieren, aus der die Blockkette erstellt wurde.

Adresse Bitcoin- Token - (OAP).

**akXma4vqxvmEqnVAKSM953wYsnjNBhN3GM7**

Anschrift Ethereum - Token COINsolidation - (ERC20).

**0x9d08c0ac0f2fdf078c883db6fa617b15776e4b41**

Schritt 2 - Der SHA512(address String-Text) jeder Adresse ohne ihren anfänglichen Bezeichner wird erhalten, indem das "a" von A1 und das "0x" von A2 entfernt wird und die letzten beiden Zeichen jeder Hash-Operation, symbolisiert mit "U", genommen werden. Verifizierungsnummern.

$U(\text{SHA512}(\text{akXma4vqxvmEqnVAKSM953wYsnjNBhN3GM7})) = \text{50}$

$U(\text{SHA512}(\text{9d08c0ac0f2fdf078c883db6fa617b15776e4b41})) = \text{fb}$

Schritt 3 - Die Zeichen jeder Adresse werden eine nach der anderen verkettet, ausgehend von der Adresse, die weniger Zeichen hat, aus denen sie sich zusammensetzt; bei gleicher Anzahl von Zeichen kann die Verkettung von jeder Adresse aus beginnen.



Adresse 1 = A10 [0], A11 [1], A12 [2], A13 [3], A14 [4] ..... A1N[n], A1N+1[n+1].

Adresse 2 = A20 [0], A21 [1], A22 [2], A23 [3], A24 [4] ..... A2N[n], A2N+1[n+1].

Verkettung von Adressen:

A10 [0] + A20 [0] + A10 [1] + A20 [1] + A11 [2] + A22 [2] + .... A1N+1 [n+1] + A2N+2 [n+1]

**\*\*Letzte Zeichen, die nicht verkettet werden können, werden an den Anfang der Zeichenkette gestellt.**

6e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

Schritt 4 - Die Anzahl der Zeichen, die in Schritt 3 verkettet werden konnten, wird dem Anfang der aus Schritt 3 resultierenden Zeichenfolge hinzugefügt.

66e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

Schritt 5 - Die beiden Überprüferpaare aus Schritt 2 jeder Richtung werden am Anfang der aus Schritt 3 resultierenden Kette in der gleichen Reihenfolge A1 + A2 verkettet.

50fb66e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

Schritt 6 - Die **CUAG-Identifikation** (Consolidated Universal Address Genesis) wird am Anfang der in Schritt 5 erstellten Adresse integriert.

**cua**50fb66e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

7

**\*\* Im Falle der Adresskonsolidierung von Bitcoin und Ethereum ergibt sich eine Adresse, die aus 82 hexadezimalen Zeichen besteht.**

## 6. Algorithmus für doppelte konsolidierte Adressen (DAC) und (HAC).

Die Erstellung eines DAC ist die gleiche wie bei der ZBV, der Unterschied besteht darin, dass in den DACs die normalen Adressen für den Empfang von Transaktionen konsolidiert werden, diese Adressen stellen keine Kryptomonedas oder Token dar.

Schritt 1 - Identifikatoren werden aus den jeweiligen Adressen entfernt, sie sind die alphanumerischen Zeichen, die die Adresse identifizieren, aus der die Blockkette erstellt wurde.

18gYNA9c2G9X8HZ8QxWLpLXZauAxFnsJbe (Bitcoin-Adresse)

0x5d2Acdb34c279Aa6d1e94a77F7b18aB938BFb2bB (Dirección Ethereum)



Schritt 2 - Der SHA512(address String-Text) jeder Adresse ohne ihren anfänglichen Bezeichner wird erhalten, indem die "1" von A1 und das "0x" von A2 entfernt wird und die letzten beiden Zeichen jeder Hash-Operation, symbolisiert mit "U", genommen werden. Verifizierungsnummern.

$$U(\text{SHA512}(8\text{gYNA9c2G9X8HZ8QxWLpLXZauAxFnsJbe})) = 48$$

$$U(\text{SHA512}(5\text{d2Acdb34c279Aa6d1e94a77F7b18aB938BFb2bB})) = \text{f3}$$

Schritt 3 - Die Zeichen jeder Adresse werden eine nach der anderen verkettet, ausgehend von der Adresse, die weniger Zeichen hat, aus denen sie sich zusammensetzt; bei gleicher Anzahl von Zeichen kann die Verkettung von jeder Adresse aus beginnen.

Adresse 1 = A10 [0], A11 [1], A12 [2], A13 [3], A14 [4] ..... A1N[n], A1N+1[n+1].

Adresse 2 = A20 [0], A21 [1], A22 [2], A23 [3], A24 [4] ..... A2N[n], A2N+1[n+1].

Verkettung von Adressen:

A10 [0] + A20 [0] + A10 [1] + A20 [1] + A11 [2] + A22 [2] + .... A1N+1 [n+1] + A2N+2 [n+1]

**\*\*Letzte Zeichen, die nicht verkettet werden können, werden an den Anfang der Zeichenkette gestellt.**

**8BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3**

Schritt 4 - Die Anzahl der Zeichen, die in Schritt 3 verkettet werden konnten, wird dem Anfang der aus Schritt 3 resultierenden Zeichenfolge hinzugefügt.

**78BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3**

Schritt 5 - Die beiden Überprüferpaare aus Schritt 2 jeder Richtung werden am Anfang der aus Schritt 3 resultierenden Kette in der gleichen Reihenfolge A1 + A2 verkettet.

**48f378BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3**

Schritt 6 - Die **DAC-Identifikation** (Dual Address Consolidated) wird am Anfang der in Schritt 5 erstellten Adresse integriert.

**dac48f378BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3**



**\*\*** Im Falle der Adresskonsolidierung von Bitcoin und Ethereum ergibt sich eine Adresse, die aus **81** hexadezimalen Zeichen besteht.

Im Falle von HAC (Hibric Address Consolidated) wird in den vorhergehenden, was variiert, sind die Adressen, die verwendet werden, in diesem Fall werden wir eine Adresse, die einen Vermögenswert (Cryptomonedas oder Token) und eine normale Standardadresse von Vermögenstransfers einer Art von Blockkette darstellt, verwenden.

**HINWEIS:** Die Größe der CUA-, HAC- und DAC-Adressen kann je nach den Adressen, aus denen sie sich zusammensetzen, im Einzelfall variieren.

### Projekt und Lösung durch COINsolidation.

Derzeit gibt es verschiedene Arten von Blockchain orientiert, um Vermögenswerte mit unterschiedlichen Eigenschaften, dies führt zu einer unendlichen Anzahl von Arten von Adressen des täglichen Gebrauchs haben, müssen eine strenge Kontrolle zu halten, um Fehler bei der Übertragung zu vermeiden.

Auf der anderen Seite ist die Welt des Kryptomoney und der Token auf Finanzexperten oder in ihrem Fall auf Experten der Blockchain-Technologie beschränkt, so dass es für den Durchschnittsbürger schwierig ist, sich an die Schaffung eines eigenen Kryptomoney oder Token zu wagen.

Wir haben die beiden vorhergehenden Probleme bei der COINsolidation gelöst, indem wir die folgenden Punkte und / oder Werkzeuge geschaffen haben.

Für den Kontrollpunkt von Adressen verschiedener Blockketten haben wir einen Algorithmus geschaffen, der zwei oder mehr Adressen in ihren verschiedenen Kombinationen zusammenführt (verbindet) und als Ergebnis eine einzige Adresse vom Typ CUA, HAC und/oder DAC ergibt.

Bei dieser Lösung wird anstelle von zwei Adressen aus derselben oder verschiedenen Blockketten nur eine konsolidierte Adresse verwendet.

Für das zweite Problem haben wir die Programmiermethodik namens Blockly verwendet. Es handelt sich dabei um ein visuelles Werkzeug, bei dem keine großen Programmierkenntnisse erforderlich sind und jede durchschnittliche Person oder Firma in der Lage sein wird, ihre



eigenen Anwendungen zu erstellen, ohne teure Entwicklungsteams, Zeit und Geld investieren zu müssen.

Wir haben die Erweiterungen (Module) erstellt, um sie einfach zu installieren und zur Erstellung mobiler Anwendungen zu verwenden, und zwar in 15 Minuten. Erstellen Sie ein Beispiel für Ihren eigenen kryptografischen Währungsumtausch oder entwickeln Sie Ihre eigene Währung (Token) innerhalb von Minuten. All dies unter Verwendung modernster Datensicherheit namens PQC (Post-Quantum Cryptography).

Installieren Sie einfach die Erweiterungen auf einem beliebigen kostenlosen Tool wie Appventor, AppyBuidler, Thunkable, Kondular oder anderen, und schon können Sie in wenigen Minuten in die Welt der Kryptomonien und Token-Erstellung eintauchen - alles in Ihrer Handfläche.

Schließlich schafft COINsolidation den Einsatz kostengünstiger Quantensicherheit (Software und Hardware), die zum Schutz von Computerdaten zu Hause eingesetzt werden kann. Gegenwärtig sind die Technologien, die auf Quanten-Computing und Sicherheit basieren, mit hohen Kosten verbunden, die nur Unternehmen mit einem hohen finanziellen Niveau schaffen und nutzen können. Bei COINsolidation sind wir jedoch der Meinung, dass neue Technologien für alle verfügbar sein sollten, dass die Fairness bei der Nutzung der Blockchain und des Quantencomputings für alle gelten sollte, dass wir freie Software (Kryptomonik) und kostengünstige Hardware (Quantensicherheit) schaffen sollten.

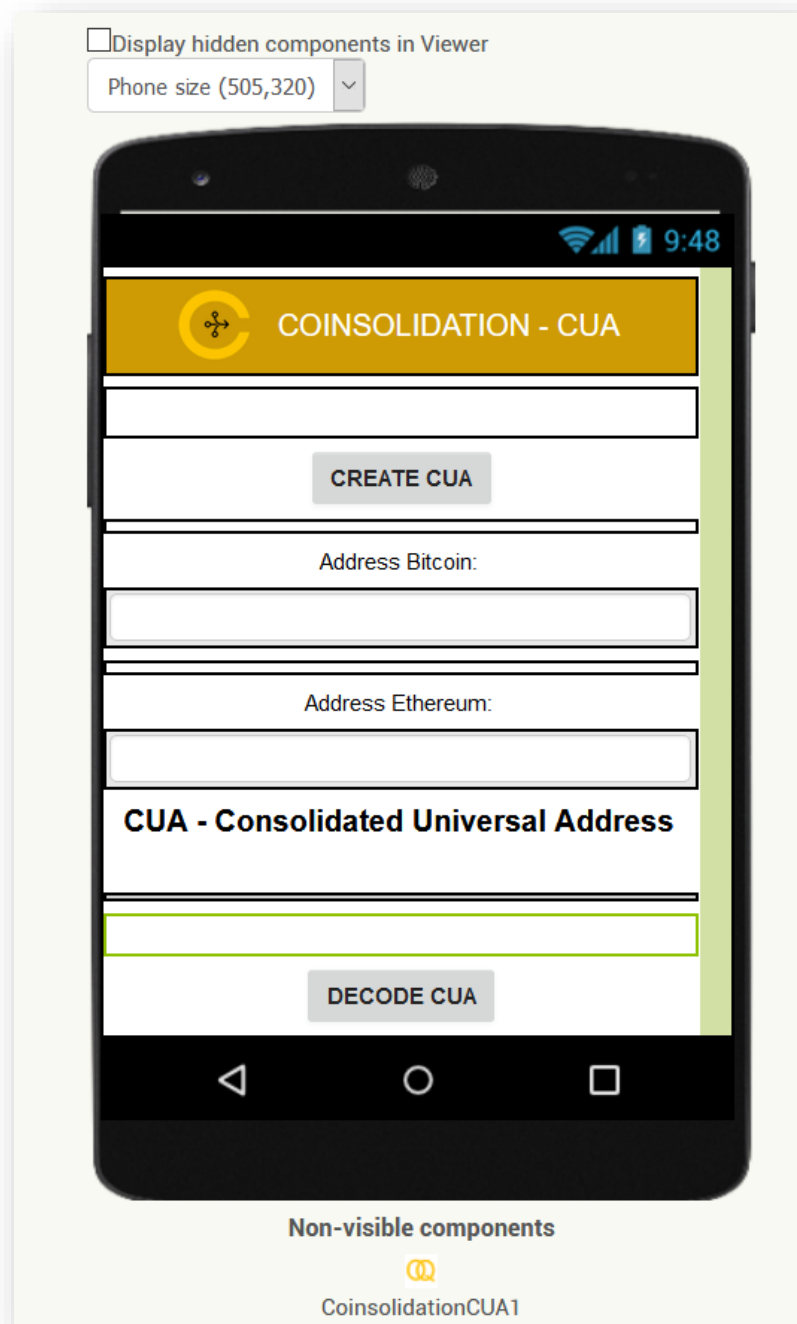




## 7. Erstellung der App CUA (Consolidated Universal Address) in 15 Minuten.

\*Anwendung für Bitcoin- und Ethereum-Münzen (BTC-ETH)

Design-Bildschirm 5 Minuten in <https://appinventor.mit.edu/>





Verwendung der Erweiterung **CoinsolidatioCUA.AIX** (5 Minuten).

The image shows a Scratch script for the COINsolidationCUA.AIX extension. It consists of three event-driven blocks:

- when GenerateCUA .Click**
  - do**
    - set addressCUA . Text to** **call CoinsolidationCUA1 .CoinsolidationEncodeCUA\_BTC\_ETH**
      - hexAddressBitcoin** **InputAddressBitcoin . Text**
      - hexAddressEthereum** **InputAddressEthereum . Text**
- when DecodeCUA .Click**
  - do**
    - call CoinsolidationCUA1 .CoinsolidationDecodeCUA\_BTC\_ETH**
      - hexAddressCUA** **InputAddressCUA . Text**
- when CoinsolidationCUA1 .OutPutAddress**
  - bitcoinStr** **ethereumStr** **checkBitcoin** **checkEthereum**
  - do**
    - set addressBitcoin . Text to** **get bitcoinStr**
    - set addressEthereum . Text to** **get ethereumStr**
    - set verifyBitcoin . Text to** **get checkBitcoin**
    - set verifyEthereum . Text to** **get checkBitcoin**



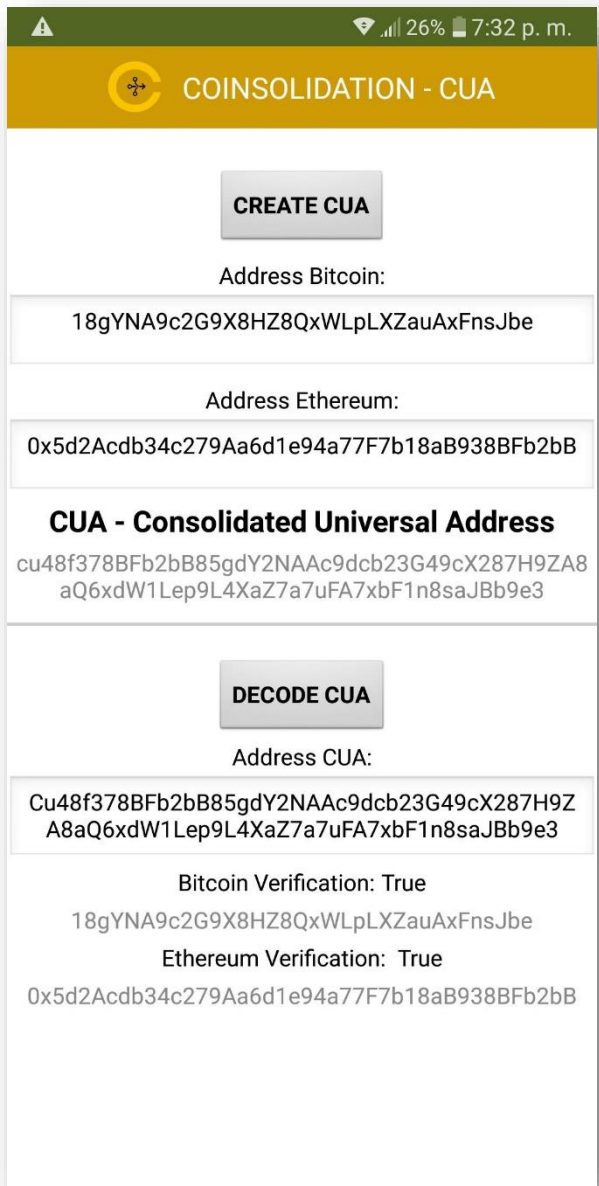
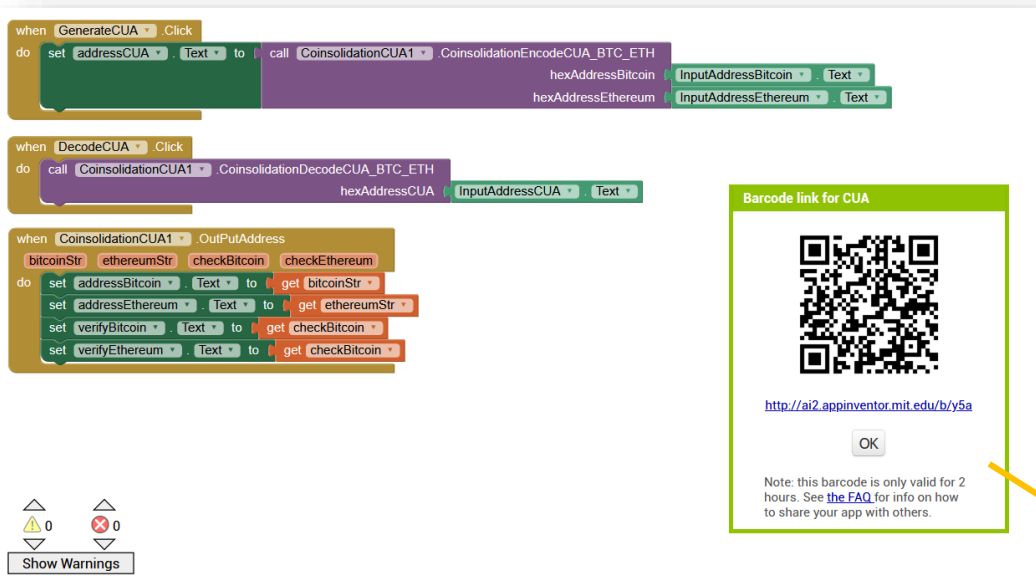
Wir erstellen die Anwendung in **Menu > Build > App** (QR-Code für .apk bereitstellen) - (5 Minuten).

The screenshot displays the app builder interface with the following components:

- Code Blocks:**
  - when GenerateCUA .Click**
    - do set addressCUA . Text to call CoinsolidationCUA1 .CoinsolidationEncodeCUA\_BTC\_ETH
      - hexAddressBitcoin InputAddressBitcoin . Text
      - hexAddressEthereum InputAddressEthereum . Text
  - when DecodeCUA .Click**
    - do call CoinsolidationCUA1 .CoinsolidationDecodeCUA\_BTC\_ETH
      - hexAddressCUA InputAddressCUA . Text
  - when CoinsolidationCUA1 .OutPutAddress**
    - bitcoinStr ethereumStr checkBitcoin checkEthereum
    - do
      - set addressBitcoin . Text to get bitcoinStr
      - set addressEthereum . Text to get ethereumStr
      - set verifyBitcoin . Text to get checkBitcoin
      - set verifyEthereum . Text to get checkBitcoin
- CUA Progress Bar:**
  - 35% progress shown.
  - Status: Compiling part 2 (please wait)
- Warning Indicators:**
  - Warning icon: 0
  - Error icon: 0
  - Show Warnings button
- UI Elements:**
  - Backpack icon (top right)
  - Target, Zoom In, Zoom Out, and Delete icons (bottom right)



Wir installierten die Anwendung auf dem Mobiltelefon vom QR aus mit der Android-Anwendung von AppInventor (MIT AI2 Companion)  
- <https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3>

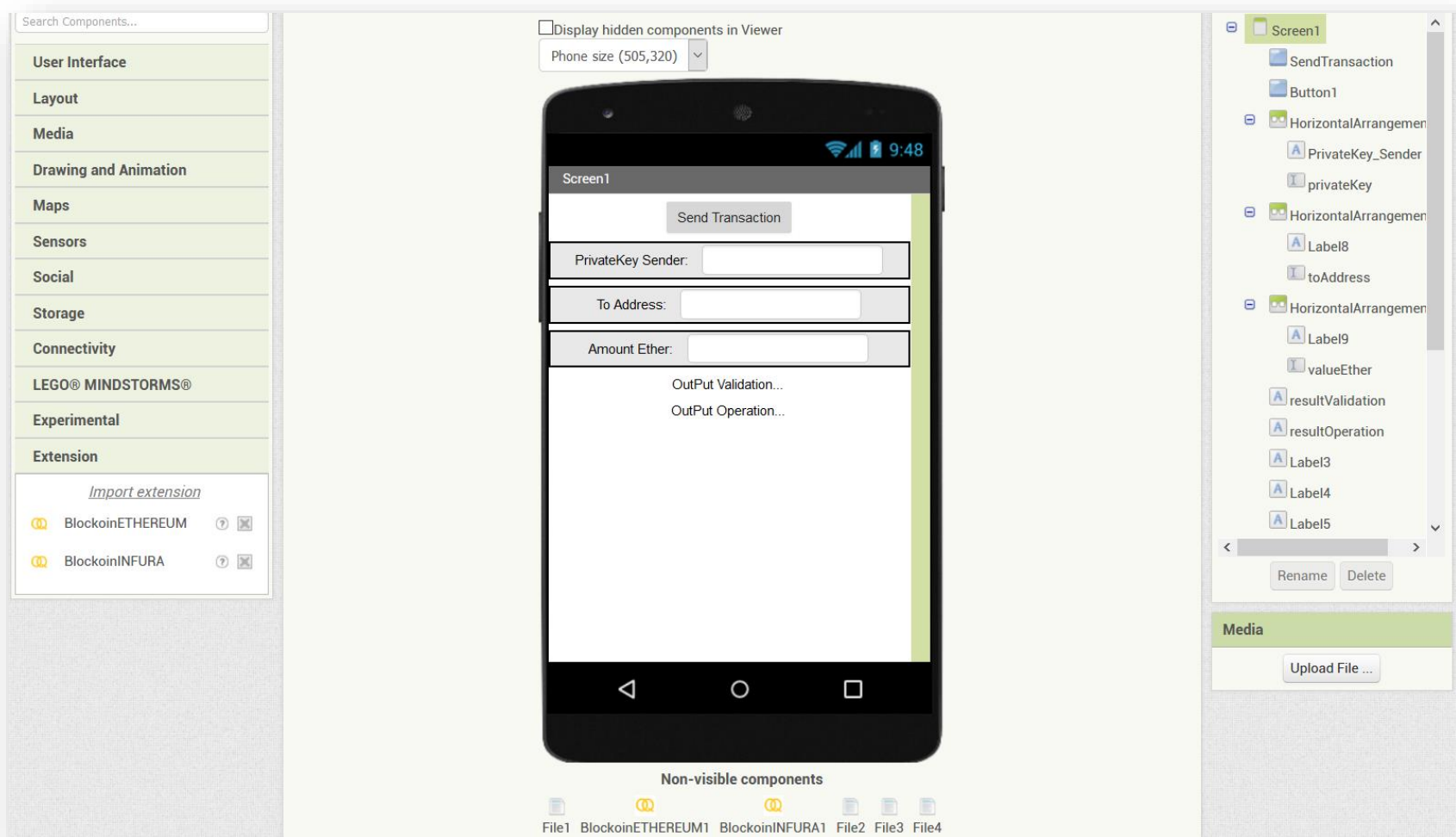


**ANMERKUNG:** Die zur Installation bereitgestellte Anwendung der APK-Datei befindet sich im folgenden Repository: <https://github.com/COINsolidation/App>

Um den Java-Code für die Generierung der CUA-Erweiterung zu überprüfen und einen konsolidierten Algorithmus zur Generierung universeller Adressen zu implementieren, lesen Sie den Anhang "Code für den CUA-Algorithmus" oder konsultieren Sie den Code-Link: <https://github.com/COINsolidation/source>



8. Erstellen Sie Ihre Ethereum-Krypto-Währungsumrechnung auf Android in nur 15 Minuten.  
Entwurf im App Inventor (Bildschirm). - 5 Minuten.





Funktionsblöcke (eth\_SendTransactionEasy) und Ereignis (OutPutSendTransactionEasy) - 5 Minuten

```
when SendTransaction .Click
do
  call BlockchainETHEREUM1 .eth_sendTransactionEasy
    hexPrivateKeySender privateKey . Text
    toAddress toAddress . Text
    valueEther valueEther . Text
```

Daten eingeben:

**PrivateKey:** Primärschlüssel zur Adresse des Absenders.

**toAddress:** Hexadezimale Adresse des Empfängers.

**valueEther:** Geben Sie die Menge an Äther an, die versendet werden soll.

```
when BlockchainETHEREUM1 .OutputSendTransactionEasy
  transactionValidationE transactionOperationE
do
  set resultValidation . Text to get transactionValidationE
  call File1 .SaveFile
    text get transactionValidationE
    fileName "/trasactionValidation.txt"
  set resultOperation . Text to get transactionOperationE
  call File1 .SaveFile
    text get transactionOperationE
    fileName "/trasactionOperation.txt"
```

Speichern Sie die Ergebnisse in Textdateien:

Funktion Datei1: Datei **trasactionValidation.txt**

Speichern Sie die Ergebnisse in Textdateien:

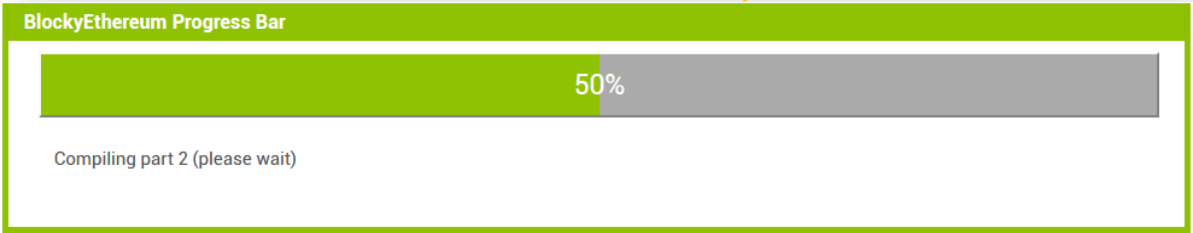
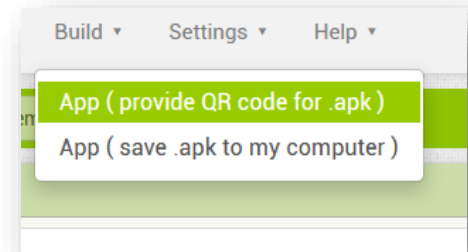
Funktion Datei2: Datei **trasactionValidation.txt**

**\*\***Weitere Einzelheiten siehe Benutzerhandbuch Ethereum Exchange (EEE) Extension im Repository: <https://github.com/COINsolidation/userguide>

**\*\***Repositorio de extensiones COINsolidation: <https://github.com/coinsolidation/Extesions-Cryptocurrencies> o OpenQbit (Blockchain & Quantum Computing) <https://github.com/openqbit-diy>



Wir kompilieren, generieren die APK-Datei, um sie auf dem Android-Gerät zu installieren. - 5 Minuten



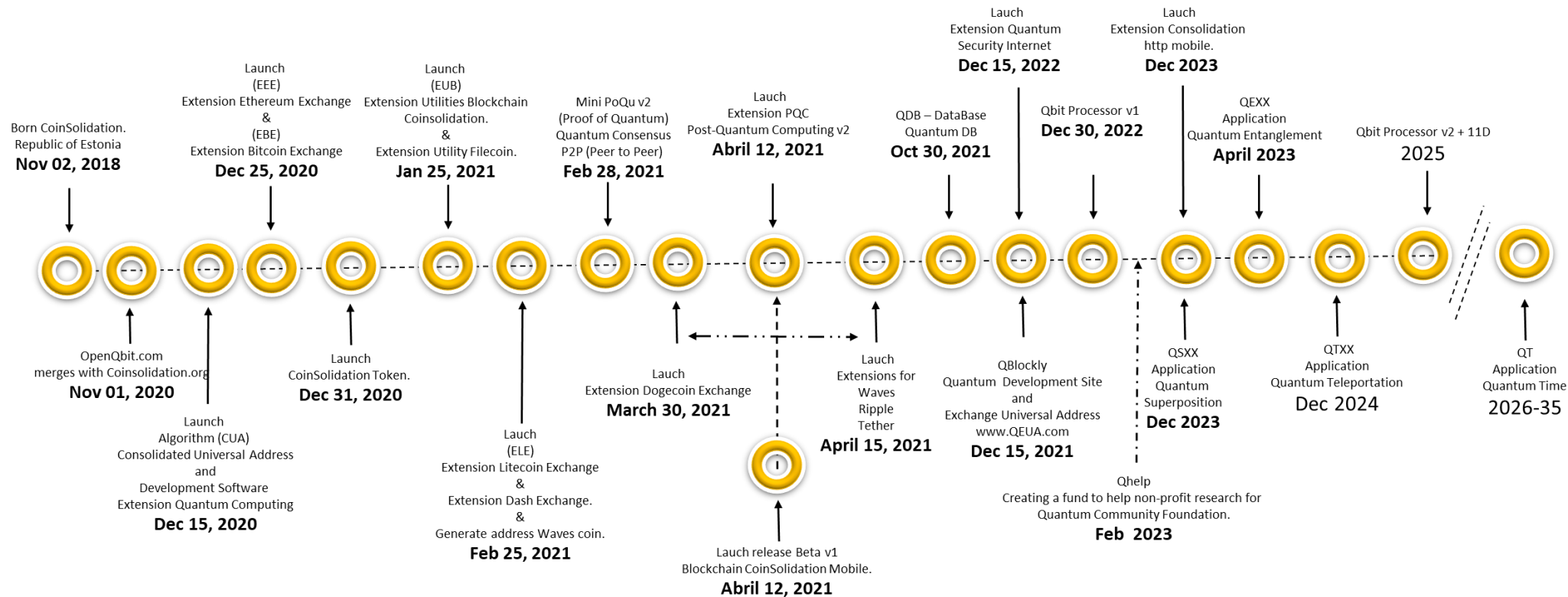
HINWEIS: Wenn die Transaktion ausgeführt wird, dauert es etwa 6 bis 8 Sekunden, bis die Schaltfläche "Transaktion senden" losgelassen wird. Aufgrund der Verbindungszeit mit dem Ethereum-Netzwerk.

Weitere Einzelheiten zur EWR-Erweiterung - (Ethereum Exchange-Erweiterung). Siehe das EEE-Benutzerhandbuch im Link:  
<https://github.com/COINsolidation/UserGuide>



9. Fahrplan COINsolidation.

ROADMAP



\*OpenQbit.com fusioniert mit COINsolidation.org (01. November 2020) / OpenQbit ist auf Quantencomputing und Sicherheitsquanten spezialisiert.  
\*Quantenprozessor Version 1 wird grundlegende Quantenlogik-Gatter für den Heimgebrauch verwenden.





EXchange  
tensions

10.Münzverdichtungsmarke (CUAG) - ICO-VERTEILUNGSPLAN.

Die ICO ist in drei Stufen unterteilt:

The private sale	\$ 0.01 USD	(30/Dec 2020 - 30/Jan 2021)	HARD CAPITAL: \$ 280,000,000.00 USD
ICO FIRST PHASE	\$ 0.01 USD	(31/Jan 2021 - 28/Feb 2021)	SOFT CAPITAL: \$ 10,000,000 USD
ICO SECOND PHASE	\$ 0.15 USD	(1/Mar 2021 - 31/Mar 2021)	

CoinSolidation TOKEN DISTRIBUTION		
	%	TOKENS
TOKEN SALE	70	28,000,000,000.00
TEAM AND DEVELOPMENT	10	4,000,000,000.00
ADVISORS	5	2,000,000,000.00
PARTNERS	5	2,000,000,000.00
EXCHANGES MARKET	1.5	600,000,000.00
MARKETING	5	2,000,000,000.00
COINSOLIDATION FOUNDATION	0.5	200,000,000.00
BLOCKLY DEVELOPER COMMUNITIES	1	400,000,000.00
OPENQBIT DEVELOPMENT AND RESEARCH OF QUANTUM COMPUTING	2	800,000,000.00
TOTAL SUPPLY 100%		40,000,000,000.00

0x9d08c0ac0f2fdf078c883db6fa617b15776e4b41	COINsolidation TOKEN
0xbbF57DE98c59B4C304C9d15BC5FAb01304aeCD97	ICO-ADRESSE
0xa646c054394f85257E18D56Cf5c6b5E603447470	ADRESSE DER KO-KONSOLIDIERUNGSOPERATION



## 11. Allgemeine Merkmale des COINsolidation-Tokens:

Erstellt von: Lugu Samaya.

Name: COINsolidation

Symbol: CUAG - (Consolidated Universal Address Genesis).

Typ: NFT

Insgesamt erzeugte Tokens: 40.000.000.000,00

Anzahl der Dezimalstellen: 18

Startland: Estland

Offizielle Website: [www.COINsolidation.org](http://www.COINsolidation.org)

Unternehmen: COINsolidation International.

Startdatum: 30. Dezember 2020

Konsens-Algorithmus: PQu (Quantennachweis)

Adress-Algorithmus: Konsolidierte Universaladresse (CUA).

Verwendete Sicherheit: PQC (Post-Quantum Cryptography) basierend auf Quantencomputing.

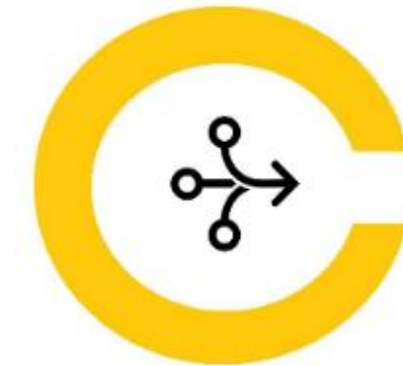
Technologischer Vorschlag: Erweiterungen für blockweise Systeme zur Verwendung von Kryptomonaden und Implementierung von Quantensicherheit.

Technologiepartnerschaften oder Vereinbarungen (Fusion):

Unternehmen: OpenQbit Inc.

Industrie: Quanteninformatik und PQC (Post-Quantum-Kryptographie).

Offizielle Website: [www.OpenQbit.com](http://www.OpenQbit.com)





## 12. Grundlegende Konzepte, die in Blockchain-Plattformen angewendet werden.

### Was ist eine Blockkette?

Die Blockkette wird im Allgemeinen mit Bitcoin und anderen Krypto-Währungen in Verbindung gebracht, aber diese sind nur die Spitze des Eisbergs, da sie nicht nur für digitales Geld verwendet wird, sondern für alle Informationen, die für Benutzer und/oder Unternehmen einen Wert haben können. Diese Technologie, deren Ursprünge auf 1991 zurückgehen, als Stuart Haber und W. Scott Stornetta die erste Arbeit an einer Kette von kryptografisch gesicherten Blöcken beschrieben, wurde erst 2008 bemerkt, als sie mit der Einführung der Bitcoin populär wurde. Gegenwärtig wird seine Verwendung jedoch in anderen kommerziellen Anwendungen nachgefragt, und es wird prognostiziert, dass sie in mittlerer Zukunft auf mehreren Märkten, wie z.B. bei Finanzinstituten oder im Internet der Dinge (Internet of Things IoT), neben anderen Sektoren, wachsen wird.

Die Blockkette, besser bekannt unter dem Begriff Blockkette, ist ein einzelner, vereinbarter Datensatz, der über mehrere Knoten (elektronische Geräte wie PCs, Smartphones, Tablets usw.) in einem Netzwerk verteilt ist. Im Falle der Krypto-Währungen können wir uns das als das Buchhaltungsbuch vorstellen, in dem jede der Transaktionen aufgezeichnet wird.

Seine Funktionsweise kann kompliziert zu verstehen sein, wenn wir auf die internen Details seiner Umsetzung eingehen, aber die Grundidee ist einfach zu verfolgen.

Sie wird in jedem Block gespeichert:

- 1.- eine Anzahl gültiger Datensätze oder Transaktionen,
- 2.- Informationen über diesen Block,
- 3.- seine Verknüpfung mit dem vorherigen Block und dem nächsten Block durch den Hash jedes Blocks –ein eindeutiger Code, der wie der Fingerabdruck des Blocks aussehen würde.

Daher hat **jeder Block** einen **bestimmten und unverrückbaren Platz innerhalb der Kette**, da jeder Block Informationen aus dem Hash des vorherigen Blocks enthält. Die gesamte Kette wird auf jedem Netzwerkknoten gespeichert, der die Blockkette bildet, so dass **eine exakte Kopie der Kette auf allen Netzwerkteilnehmern gespeichert wird**.

### Was ist eine Adresse oder ein Konto innerhalb der Blockkette der Ethereum-Plattform?

Es handelt sich um eine Zeichenfolge von 42 Zeichen in der Ethereum-Plattform, die eine Zahl in hexadezimaler Basis darstellt, in der die im Ethereum definierten Vermögenswerte



hinterlegt oder versandt werden. Bei anderen Blockketten-Plattformen kann z.B. die Anzahl der Zeichen des Kontos oder der Adresse unterschiedlich sein:

**0x5d2Acdb34c279Aa6d1e94a77F7b18aB938BFb2bB**

### **Was ist ein Kryptomoney?**

Es handelt sich um eine digitale oder virtuelle Währung, die als Tauschmittel fungieren soll. Es verwendet Kryptographie (digitale Sicherheit), um Transaktionen zu sichern und zu verifizieren, sowie um die Schaffung neuer Einheiten eines bestimmten Kryptomoney zu kontrollieren.

### **Was ist ein Zeichen?**

Tokens sind digitale Assets, die innerhalb eines bestimmten Projekt-Ökosystems verwendet werden können.

Der Hauptunterschied zwischen Tokens und Krypto-Währungen besteht darin, dass erstere eine andere Blockkettenplattform (nicht ihre eigene) benötigen, um zu funktionieren. Ethereum ist die gebräuchlichste Plattform zur Erstellung von Tokens, vor allem wegen seiner intelligenten Vertragsfunktion. Die auf der Ethereum-Blockkette erzeugten Wertmarken sind allgemein als ERC-20-Marken bekannt, obwohl es auch andere, speziellere Arten von Marken gibt, wie z.B. die ERC-721-Marke, die hauptsächlich für Sammelobjekte (Karten, Verwendung in Videospielen, Kunstwerke usw.) verwendet wird.

### **Was ist ein Austausch?**

Eine Krypto-Währungsbörse ist der Treffpunkt, an dem der Austausch von Krypto-Währungen im Austausch gegen Fiat-Geld oder andere Krypto-Währungen stattfindet. In diesen Online-Tauschbörsen wird der Marktpreis generiert, der den Wert der Kryptomien auf der Grundlage von Angebot und Nachfrage markiert.

### **Was sind Wechselkurse?**

Dies sind die Kurse des Wertes eines Äthers oder einer anderen Krypto-Währung in der Umlaufwährung des jeweiligen Landes. Zum Beispiel hat ein Äther am Tag der Erstellung dieses Handbuchs einen Wert in US-Dollar von \$430,94

### **Was ist eine Transaktion?**

Es handelt sich um die Ausführung oder Übertragung einer Art von nicht materiellem Vermögenswert, dem innerhalb des Ethereum-Systems ein vorab festgelegter Wert



zugewiesen werden kann und der später in einen materiellen Wert für ein Unternehmen oder eine Person umgewandelt werden kann.

### Was ist txHash?

Es handelt sich um eine hexadezimale Zahl, die es ermöglicht, das Ergebnis jeder Transaktion im Detail nachzuvollziehen.

### Welche Arten von Transaktionen gibt es?

Sie haben zwei Arten, eine ist die Transaktion "offline", die ohne die Notwendigkeit einer Verbindung zum Hauptnetzwerk von Ethereum erstellt wird, kann gespeichert werden, bis Sie sich für eine Verbindung zum Netzwerk von Ethereum entscheiden und die Transaktion freigeben, haben den Vorteil der Sicherheit, weil die gesamte Transaktion offline verarbeitet wird, was jede Anomalie verhindert, die in der Netzwerkverbindung sein könnte. Die andere Transaktion ist die "Online"-Transaktion, die immer mit dem Internet verbunden sein muss, mit den Sicherheitsvor- und -nachteilen, die sie mit sich bringt.

### Was ist eine Blockchain-Adresse?

Eine Adresse oder ein Konto besteht aus drei Teilen, der Adresse, dem öffentlichen Schlüssel und dem privaten Schlüssel. Diese beiden Schlüssel sind eine Folge von Zahlen und Zeichen im Hexadezimalformat, die zum Senden und Empfangen (aktiv) oder Äther (digitale Währung) verwendet werden.

Der Primärschlüssel sollte niemals mit irgendjemandem geteilt werden, da er die Freigabe des Saldos (unterzeichnet die Transaktionen) auf dem Konto autorisiert.

Der öffentliche Schlüssel ist der gesamten Öffentlichkeit bekannt und wird mit jedermann geteilt, da er als Referenz dient, um zu bestätigen, dass die Transaktion wertmäßig korrekt ist und an wen sie gesendet wird.

Beispiele für Komponenten des Ethereum-Netzwerkmanagements:

```
{  
  "private": "429a043ea6393b358d3542ff2aab9338b9c0ed928e35ec0aed630b93adb14a1c",  
  "public":  
    "049b4b7e72701a09d3ee09165bba460f2549494a9d9fd7a95aaac57c2827eac162fd9e105b  
    2461cd6594ca8ca6a8daf10fe982f918be1b0060c87db9cfbcd289a8",  
  "address": "88ab6dcecc3603c7042f4334fc06db8e8d7062d5"  
}
```



### 13. Was ist blockweise Programmierung?

**Blockly** ist eine **visuelle Programmiermethodik**, die aus einem einfachen Satz von Befehlen besteht, die wir kombinieren können, als wären sie die Teile eines Puzzles. Es ist ein sehr nützliches Werkzeug für diejenigen, die **lernen** wollen, **wie man auf** intuitive und einfache Weise **programmiert**, oder für diejenigen, die bereits programmieren können und das Potenzial dieser Art der Programmierung erkennen wollen. Es basiert auf der JavaScript-Sprache und wurde von der Firma Google und dem MIT entwickelt.

Blockly ist eine Form des Programmierens, bei der man keinen Hintergrund in irgendeiner Computersprache benötigt, weil es nur das Zusammenfügen von Grafikblöcken ist, als ob wir Lego oder ein Puzzle spielen würden, man braucht nur etwas Logik und das war's!

Jeder kann Programme für Mobiltelefone (Smartphones) erstellen, ohne sich mit diesen schwer verständlichen Programmiersprachen herumschlagen zu müssen, indem er einfach Blöcke in grafischer Form zusammensetzt.

### 14. Anhang "Code für CUA-Algorithmus".

Verweis auf Github: <https://github.com/coinsolidation/source>

### 15. Begriffe.

Nutzungsbedingungen siehe auf der Website [www.coinsolidation.org](http://www.coinsolidation.org) oder <https://github.com/coinsolidation/Terms>

Unterstützung bei der kommerziellen Nutzung.  
[support@coinsolidation.org](mailto:support@coinsolidation.org)

Verkaufsblockkette geschäftliche Nutzung.  
[sales@coinsolidation.org](mailto:sales@coinsolidation.org)

Rechtliche Informationen und Lizenzfragen oder Bedenken  
[legal@coinsolidation.org](mailto:legal@coinsolidation.org)

Soziale Vernetzung:

Twitter: <https://twitter.com/ecoinsolidation>

Facebook: <https://www.facebook.com/coinsolidation>