



EX^{change}
tensions

COINsolidatie.

Tallinn, Estland. (E-residentie)

White Paper - Witboek.

versie 1.0.0

December 2020.

COINsolidation.org is een geregistreerd handelsmerk van COINsolidation International, onder vrije en commerciële licentie. Gebruiksvoorwaarden op: www.Coinsolidation.org

COINsolidation International is gefuseerd met www.OpenQbit.com voor samenwerking op het gebied van kwantummechanica (Quantum Security & Quantum Computing). Deze fusie maakt het gebruik, het delen en de re-engineering mogelijk van technologie die is ontwikkeld door OpenQbit Inc. (Estland, E-residentie)

Inhoud

1. Inleiding.....	3
2. Security Quantum Computing.....	7
3. Aanmaken van "Hardware" apparaat van een QRNG (Quantum Random Number Generator). 12	
4. Wat is het bewijs van Quantum (PQu)?	18
5. Algoritme voor het creëren van een geconsolideerd universeel adres (CUA).....	21
6. Algoritme voor dual consolidated address (DAC) en (HAC).	22
Project en oplossing door COINsolidatie.....	24
7. Creatie van App CUA (Consolidated Universal Address) in 15 minuten.	25
8. Maak uw Ethereum crypto valutawissel op Android in slechts 15 minuten.	29
9. Stappenplan COINsolidatie.	32
10. COINsolidatie Token (CUAG) - ICO DISTRIBUTION PLAN.....	33
11. Algemene kenmerken van de COINsolidatiepenning:	34
12. Basisconcepten toegepast in Blockchain-platforms.	35
13. Wat is Blockly-programmering?	38
14. Bijlage "Code voor CUA-algoritme"	38
15. Voorwaarden.....	38

1. Inleiding.

Op dit moment zijn fusies actueel, of het nu gaat om een economisch, technologisch of marktgoed.

We presenteren het eerste model van crypto-fusie of crypto-tokens dat een back-up biedt tussen twee cryptomonaden, tokens of een mengsel daarvan, op basis van een algoritme om een geconsolideerd adres te creëren dat wordt gebruikt en gegenereerd in de COINsolidatie omgeving.

We hebben drie soorten geconsolideerde adressen gecreëerd.

De **CUA** (Consolidated Universal Address) wordt gebruikt om een nieuw (actief) token te consolideren en te creëren dat door de gebruiker kan worden gebruikt. De combinatie kan bestaan uit drie soorten: Cryptocurrency-Cryptocurrency, Cryptocurrency-Token of Token-Token. In het geval van de CUA wordt deze gevormd door een Token-Token-relatie.

Het **HAC** (Hibric Address Consolidated) wordt gebruikt wanneer we een adres moeten consolideren met betrekking tot een cryptocurrency en/of token en een normaal adres voor de overdracht van activa.

De **DAC** (Dual Address Consolidated) wordt gebruikt om twee normale adressen van dezelfde blokken of van twee verschillende technologieën te beheren en te consolideren.

Laten we beginnen met het bekijken van de voordelen van de CUA.

Een CUA-adres bestaat uit het COINsolidatietoken-adres (statisch adres) en een extra token dat bekend staat als de "Colored Coin" (variabel adres). In dit geval kunnen we zien dat de CUA-adressen altijd gevormd worden door adressen van een of andere combinatie van activa (Cryptosolids of penningen).

In ons geval, wanneer we de COINsolidatie penning en een OAP penning consolideren zullen we deze kennen als de "CUA genesis" of **CUAG (Consolidated Universal Address Genesis)**.

El token COINsolidation esta creado en el *blockchain Ethereum* y usa el standard ERC20 (Ethereum Request for Comments 20).

De "gekleurde munt" is gebaseerd op en gemaakt door de *Bitcoin-blokken* en maakt gebruik van de Open Asset Protocol (**OAP**) standaard.

Laten we beginnen met het evalueren van het potentieel en het voordeel van het consolideren van adressen.

- I. Voor de gebruikers die een CUA aanmaken zal het mogelijk zijn om een token (**OAP**) aan te maken dat kan worden aangepast door de gebruiker die de CUA heeft aangemaakt, de gebruiker zal de mogelijkheid hebben om zijn eigen token of crypto actief te hebben zodat hij het kan gebruiken in de creatie, ondersteuning of uitbreiding van zijn bedrijf(en), op een eenvoudige en gemakkelijke manier zal hij een troef hebben in de wereld van crypto-tokens.
- II. Voor bedrijven die een CUA creëren, kunnen ze een token (**OAP**) hebben dat ze kunnen gebruiken om waarde te creëren in hun toeleveringsketen of het actief te gebruiken in liquiditeitstransacties op basis van de economische steun van hun bedrijfsactiva en -passiva.
- III. Voor bestaande cryptomonaden en tokens door het creëren van een CUA, zullen ze in staat zijn om uw adres te gebruiken dat uw bedrijfsmiddel identificeert en door het te consolideren met het token (**OAP**) zullen ze in staat zijn om hun vraag te laten groeien door het aanbieden van hun huidige en toekomstige investeerders hun eigen token voor hun gebruikers.

Voorbeeld van **CUAG**, we hebben de respectievelijke adressen van twee verschillende Blockchain:

Adres Bitcoin- Token - (OAP).

akXma4vqxvmEqnVAKSM953wYsnjNBhN3GM7

Adres Ethereum - Token COINsolidation - (ERC20).

0x8390f8abb8fd8ad3bf8457db59f2ed75e015d303

Door een algoritme toe te passen om de vorige adressen te consolideren, verkrijgen we het CUA-adres.

cua50d0615d303k8X3m9a04fv8qaxbvb8EfQdn8VaAdK3SbMf985435w7Ydsbn5j9NfB2heNd37G5Me70

* Voor meer details over het algoritme zie paragraaf 7.- "Algoritme voor het creëren van een geconsolideerd universeel adres".

We hebben dus een enkele richting die twee verschillende technologieën uit twee verschillende richtingen vertegenwoordigt, geconsolideerd in één richting.

Wij weerspiegelen dit op het gebied van winstgevendheid en financiële expansie op een eenvoudige en ongecompliceerde manier door te investeren in een van de penningen die onze CUA direct integreert krijgt u een penning op basis van de Bitcoin-blokketen (OAP).

Laten we nu eens kijken naar twee richtlijnen die we ook in COINsolidatie naar voren hebben gebracht voor de wereld van de cryptomontages en/of tokens.

COINsolidation token is het project om adressen te consolideren en een directe ondersteuning te hebben bij het verkrijgen van een aangepaste token om te gebruiken in de groei van elke gebruiker in de wereld van cryptomontages. Het project is geboren in 2018 met een groep ingenieurs en financiers die geïnteresseerd zijn in het samenvoegen van de financiële en technologische sectoren, om te profiteren van investeringsfondsen en om innovatieve technologie zoals quantum computing in te zetten om activa te beveiligen, evenals het doel om instrumenten te gebruiken die voor iedereen beschikbaar zouden kunnen zijn.

Na een evaluatie van verschillende ontwikkelingsmogelijkheden hebben we gekozen voor de optie van de visuele Blockly programmeermethodologie. Deze methodiek is gebaseerd op het gebruik van extensies of modules (programma's in java programmeertaal) met eenvoudige maar krachtige functionaliteiten om de business uit te breiden in crypto-assets voor elke persoon, om dit te bereiken moeten we de volgende punten afdekken:

- ✓ a.- Onmiddellijke financiële ROI voor gebruikers, investeerders en activa door het kunnen creëren van een niet-materieel actief (persoonlijke penningen) voor het exclusieve gebruik van de maker en gebruiker van de (CUA)
- ✓ b.- We gebruiken de voordelen van het samenvoegen van twee blockchains bij de keuze van de gebruiker om de huidige en toekomstige investeringen in de crypto-actieve markt met behulp van de (CUA) te laten groeien.
- ✓ c.- Vergemakkelijken van het beheer van afzonderlijke adressen door ze te consolideren in (DAC).
- ✓ d.- Creëer en gebruik beveiliging op basis van Quantum Computing.

De uitdaging begon met het creëren van de technologie van extensies die voldoende modulair in functionaliteit en "grootte" zijn, dit laatste was de uitdaging van het ontwikkelingsteam van COINsolidation aangezien de extensies die gebruikt worden in de Blockly methodologie en systemen van dit type (AppInventor, AppyBuilder, Thunkable, Kondular), etc) worden meestal extensies (programma's) gemaakt die niet groter zijn dan 100k - 300k bytes, met de beperkingen die in hun omvang de taak hebben om extensies te maken voor gebruik in de huidige Blockchain waren vrijwel onmogelijk omdat de bibliotheken die worden gebruikt in hun creatie groter zijn dan tussen 10MB en 35MB deze maten voor de huidige tools Blockly-systemen zijn niet functioneel om ze te gebruiken.

Het team moest de programmeermethodologie en de bibliotheken creëren, aanpassen en minimaliseren om de uitbreidingen met de optimale functionaliteit, veiligheid en omvang te verkrijgen.

Na bijna twee jaar van ontwikkeling en testen hebben we de eerste blockchain "beta" met behulp van extensies voor Blockly inclusief het consensus algoritme "Proof of Quantum" met behulp van quantum beveiliging voor cryptomoney uitwisseling afgerond.

Op dit moment hebben we een eigen blockchain die is vrijgegeven voor "Beta"-testen en tegen het einde van 2021 zullen we de productieversie vrijgeven voor informatieverbreiding. Op dit moment is ons COINsolidatietoken gebaseerd op het Ethereum en de Bitcoin-blokketen, de laatste voor het maken van aangepaste tokens voor gebruikers.

2. Security Quantum Computing.

Hoe werkt quantumcomputing? ⁽²⁾

De digitale transformatie brengt sneller dan ooit tevoren veranderingen in de wereld teweeg. Zou u geloven dat het digitale tijdperk ten einde loopt? **Digitale geletterdheid** is al geïdentificeerd als een gebied waar open kennis en toegankelijke mogelijkheden om over technologie te leren dringend nodig zijn om de lacunes in de sociale en economische ontwikkeling aan te pakken. Het leren van de sleutelbegrippen van het digitale tijdperk zal nog kritischer worden met de aanstaande komst van een nieuwe technologische golf die in staat is om bestaande modellen met verbazingwekkende snelheid en kracht te transformeren: de **kwantumtechnologieën**.

In dit artikel vergelijken we de basisconcepten van traditional computing en quantum computing; en we beginnen ook de toepassing ervan in andere gerelateerde gebieden te onderzoeken.

Wat zijn kwantumtechnologieën?

In de loop van de geschiedenis heeft de mens de technologie ontwikkeld zoals hij door de wetenschap heeft begrepen hoe de natuur werkt. Tussen 1900 en 1930 leidde de studie van enkele nog niet goed begrepen natuurkundige verschijnselen tot een nieuwe natuurkundige theorie, **Quantum Mechanica**. Deze theorie beschrijft en verklaart de werking van de microscopische wereld, de natuurlijke habitat van moleculen, atomen of elektronen. Dankzij deze theorie is het niet alleen mogelijk geweest om deze fenomenen te verklaren, maar is het ook mogelijk geweest om te begrijpen dat de subatomaire werkelijkheid op een volledig contra-intuïtieve, bijna magische manier werkt, en dat er in de microscopische wereld gebeurtenissen plaatsvinden die niet in de macroscopische wereld voorkomen.

Deze **kwantumeigenschappen** omvatten kwantumsuperpositie, kwantumverstrengeling en kwantumteleportatie.

- **Quantum superpositie** beschrijft hoe een deeltje in verschillende toestanden tegelijk kan zijn.
- **Quantumverstrengeling** beschrijft hoe twee deeltjes die zo ver uit elkaar liggen als gewenst, zodanig met elkaar in verband kunnen worden gebracht dat de ander zich er bij de interactie met de ene bewust van is.
- **Kwantumteleportatie** gebruikt kwantumverstrengeling om informatie van de ene plaats naar de andere in de ruimte te sturen zonder er doorheen te hoeven reizen.

Kwantumtechnologieën zijn gebaseerd op deze kwantumeigenschappen van het subatomaire karakter.

In dit geval stelt het begrip van de microscopische wereld door middel van Quantum Mechanics ons vandaag de dag in staat om technologieën uit te vinden en te ontwerpen die het leven van mensen kunnen verbeteren. Er zijn veel en zeer verschillende technologieën die gebruik maken van kwantumfenomenen en sommige daarvan, zoals lasers of magnetische resonantie beeldvorming (MRI), zijn al meer dan een halve eeuw bij ons. Op dit moment zijn we echter getuige van een technologische revolutie op gebieden als quantumcomputing, quantuminformatie, quantumsimulatie, quantumoptiek, quantummetrologie, quantumklokken of quantumsensoren.

Wat is quantum computing? Eerst moet je de klassieke informatica begrijpen.




FIGURA 1.
Ejemplos de caracteres en lenguaje binario.

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

Om te begrijpen hoe kwantumcomputers werken, is het handig om eerst uit te leggen hoe de computers die we dagelijks gebruiken, en die we in dit document zullen aanduiden als digitale of klassieke computers, werken. Deze gebruiken, net als de rest van de elektronische apparaten zoals tablets of mobiele telefoons, bits als de fundamentele eenheden van het geheugen. Dit betekent dat programma's en applicaties gecodeerd zijn in bits, dat wil zeggen in binaire taal van nullen en enen. Elke keer dat we met een van deze apparaten communiceren, bijvoorbeeld door een toets op het toetsenbord in te drukken, worden er reeksen nullen en enen aangemaakt, vernietigd en/of gewijzigd in de computer.

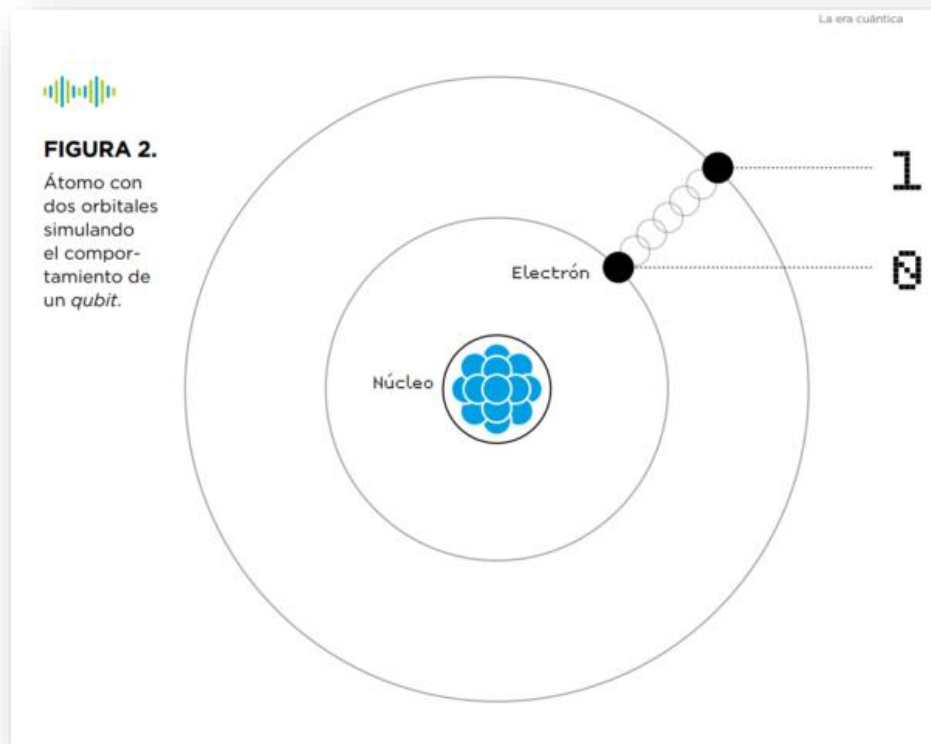
De interessante vraag is, wat zijn deze nullen en enen fysiek in de computer? De nul- en één-toestanden komen overeen met de elektrische stroom die al dan niet circuleert door microscopische onderdelen die transistors worden genoemd en die als schakelaars fungeren. Als er geen stroom loopt, is de transistor "uit" en komt hij overeen met bit 0, en als hij loopt is hij "aan" en komt hij overeen met bit 1.

Simpeler, het is alsof de bits 0 en 1 overeenkomen met gaten, zodat een leeg gat een beetje 0 is en een gat bezet door een elektron een beetje 1. Dit is de reden waarom deze apparaten elektronica worden genoemd. Als voorbeeld toont figuur 1 het binaire schrift van sommige

karakters. Nu we een idee hebben van hoe de huidige computers werken, laten we proberen te begrijpen hoe kwantums werken.

Van bits tot qubits

De fundamentele eenheid van informatie in kwantumberekeningen is de kwantumbit of qubit. Qubits zijn per definitie kwantumsystemen op twee niveaus - we zien hier voorbeelden - die net als bits op het lage niveau kunnen staan, dat overeenkomt met een toestand van lage excitatie of energie gedefinieerd als 0, of op het hoge niveau, dat overeenkomt met een toestand van hogere excitatie of gedefinieerd als 1. Echter, en hier ligt het fundamentele verschil met de klassieke rekenkunst, qubits kunnen ook in een van de oneindige tussenliggende toestanden tussen 0 en 1 liggen, zoals een toestand die half 0 en half 1 is, of driekwart van 0 en een kwart van 1.



Kwantumalgoritmen, exponentieel krachtiger en efficiënter computergebruik

Het doel van kwantumcomputers is om gebruik te maken van deze kwantumeigenschappen van *qubits*, zoals kwantumsystemen die ze zijn, om kwantumalgoritmen te draaien die gebruik maken van overlappende en interleaving om veel meer rekenkracht te bieden dan de klassiekers. Het is belangrijk om erop te wijzen dat de echte verandering van het paradigma niet bestaat uit het doen van hetzelfde wat digitale of klassieke computers doen - de huidige - maar dat kwantumalgoritmes het mogelijk maken om bepaalde bewerkingen

op een totaal andere manier uit te voeren die in veel gevallen efficiënter blijken te zijn - dat wil zeggen, in veel minder tijd of met veel minder rekenkundige middelen -.

Laten we eens kijken naar een concreet voorbeeld van wat dit inhoudt. Laten we ons voorstellen dat we in Bogotá zijn en dat we de beste route naar Lima willen weten uit een miljoen opties om er te komen ($N=1.000.000$). Om met behulp van computers de optimale route te vinden moeten we 1.000.000 opties digitaliseren, wat betekent dat ze moeten worden vertaald in bittaal voor de klassieke computer en in *qubits* voor de kwantumcomputer. Terwijl een klassieke computer een voor een alle paden zou moeten analyseren tot het vinden van de gewenste, maakt een kwantumcomputer gebruik van het proces dat bekend staat als kwantumparallelisme en dat het mogelijk maakt om alle paden in één keer te overwegen. Dit houdt in dat, terwijl de klassieke computer de volgorde van $N/2$ stappen of iteraties nodig heeft, dat wil zeggen 500.000 pogingen, de kwantumcomputer het optimale pad zal vinden na slechts \sqrt{N} bewerkingen op het register, dat wil zeggen 1.000 pogingen.

In het vorige geval is het voordeel kwadratisch, maar in andere gevallen is het zelfs exponentieel, wat betekent dat we met n *qubits* een rekencapaciteit kunnen verkrijgen die gelijk is aan 2^n bits. Om dit te illustreren is het gebruikelijk om te tellen dat we met ongeveer 270 qubits meer basistoestanden zouden kunnen hebben in een kwantumcomputer - meer verschillende en gelijktijdige tekenreeksen - dan het aantal atomen in het universum, dat wordt geschat op ongeveer 10^{80} . Een ander voorbeeld is dat we met een kwantumcomputer van tussen de 2000 en 2500 *qubits* naar schatting vrijwel alle huidige cryptografie zouden kunnen breken (de zogenaamde publieke sleutelcryptografie).

Waarom is het belangrijk om de kwantumtechnologie te kennen?

We bevinden ons in een moment van digitale transformatie waarbij verschillende opkomende technologieën zoals blockchain, kunstmatige intelligentie, drones, Internet of things, virtual reality, 5G, 3D-printers, robots of autonome voertuigen meer en meer aanwezig zijn in meerdere velden en sectoren. Deze technologieën, die geroepen zijn om de kwaliteit van het leven van de mens te verbeteren en de ontwikkeling te versnellen en sociale gevolgen te genereren, gaan vandaag de dag op een parallelle manier vooruit. Slechts zelden zien we bedrijven producten ontwikkelen die gebruik maken van combinaties van twee of meer van deze technologieën, zoals blockchain en IoT of drones en kunstmatige intelligentie. Hoewel ze voorbestemd zijn om te convergeren en zo een exponentiële grotere impact te genereren, is de eerste fase van de ontwikkeling waarin ze zich bevinden en de schaarste aan ontwikkelaars en mensen met een technisch profiel een nog hangende taak.

Vanwege hun ontwrichtend vermogen wordt verwacht dat kwantumtechnologieën niet alleen convergeren met al deze nieuwe technologieën, maar dat ze ook een transversale invloed hebben op vrijwel al deze technologieën. Quantum computing vormt een bedreiging

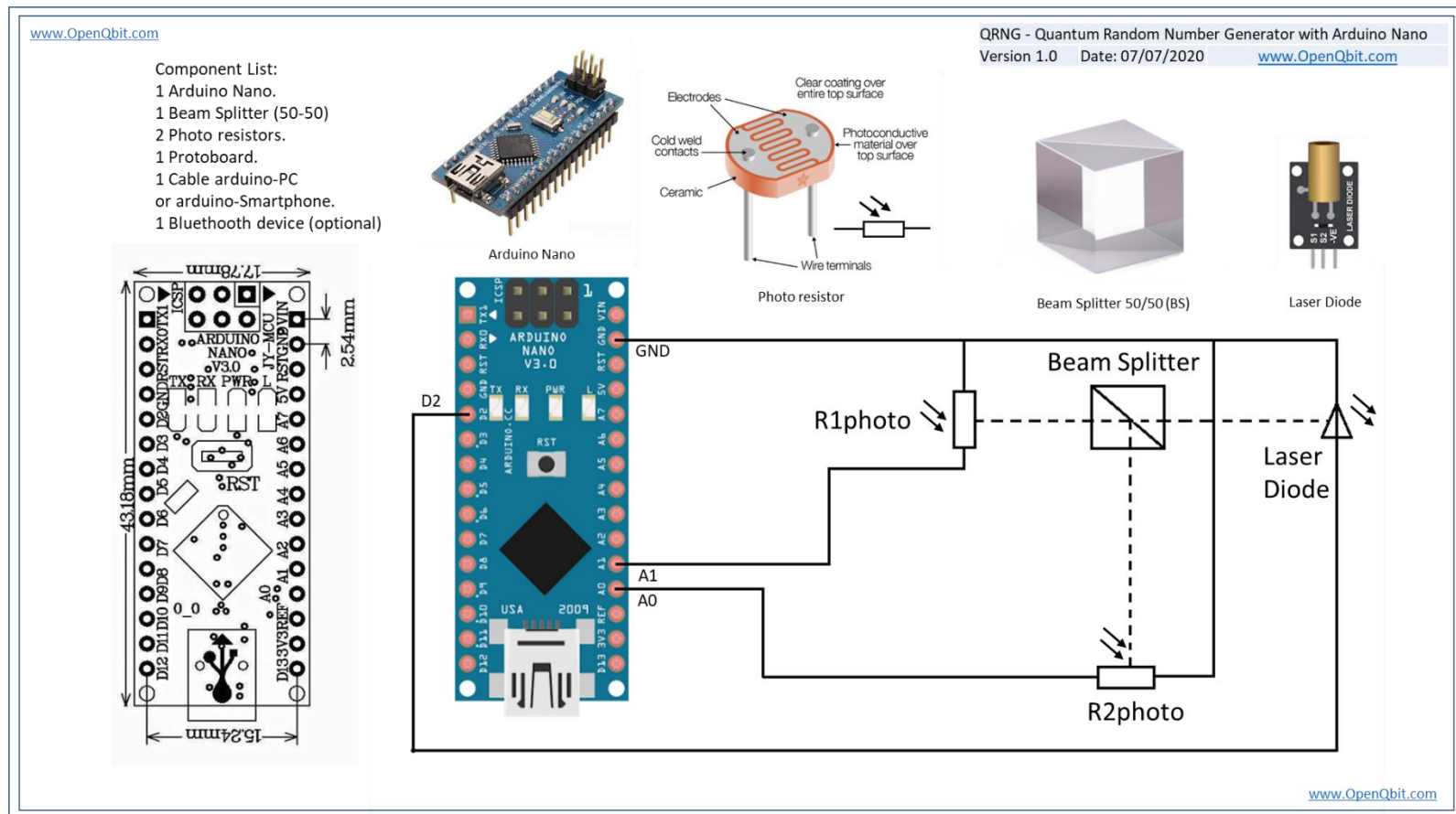
voor de authenticatie, uitwisseling en veilige opslag van gegevens en heeft een grote impact op technologieën waar cryptografie een meer relevante rol speelt, zoals cyberveiligheid of blokketen, en een kleine negatieve impact, maar moet ook in aanmerking worden genomen bij technologieën zoals 5G, IoT of drones.

Wil je oefenen in quantum computing?

Er zijn al tientallen kwantumcomputers beschikbaar op het net met verschillende programmeertalen die al in gebruik zijn, zoals C, C++, Java, Matlab, Maxima, Python of Octave. Ook nieuwe talen zoals Q#, gelanceerd door Microsoft. U kunt met een virtuele kwantummachine verkennen en spelen via platforms zoals IBM en Rigetti.

3. Aanmaken van "Hardware" apparaat van een QRNG (Quantum Random Number Generator).

We zullen nu een fysiek "Hardware" apparaat maken om Quantum Random Numbers (QRNG) te genereren met goedkope componenten die gemakkelijk thuis kunnen worden geassembleerd en ongeveer 35 dollar kosten.



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);
}
```

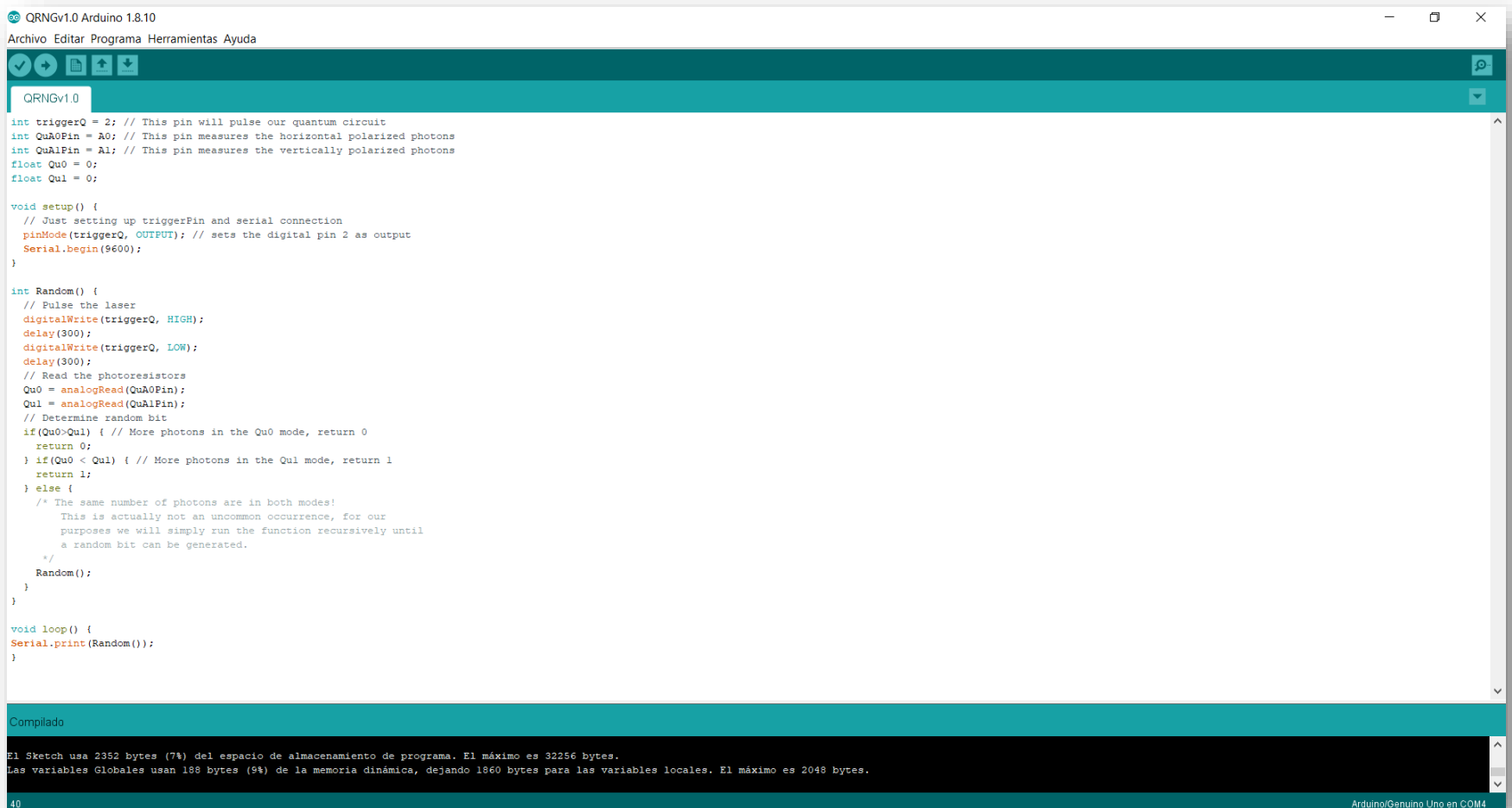
```
int Random() {
  // Pulse the laser
  digitalWrite(triggerQ, HIGH);
  delay(300);
  digitalWrite(triggerQ, LOW);
  delay(300);
  // Read the photoresistors
  Qu0 = analogRead(QuA0Pin);
  Qu1 = analogRead(QuA1Pin);
  // Determine random bit
  if(Qu0>Qu1) { // More photons in the Qu0 mode, return 0
    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.....

Het samenstellen van het QRNGv10.ino programma en het uploaden naar 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 photoresistors (QuA0Pin and QuA1Pin) to measure polarized photons. The 'Random()' function pulses a laser (triggerQ) and reads the sensor values to determine 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());
}
```

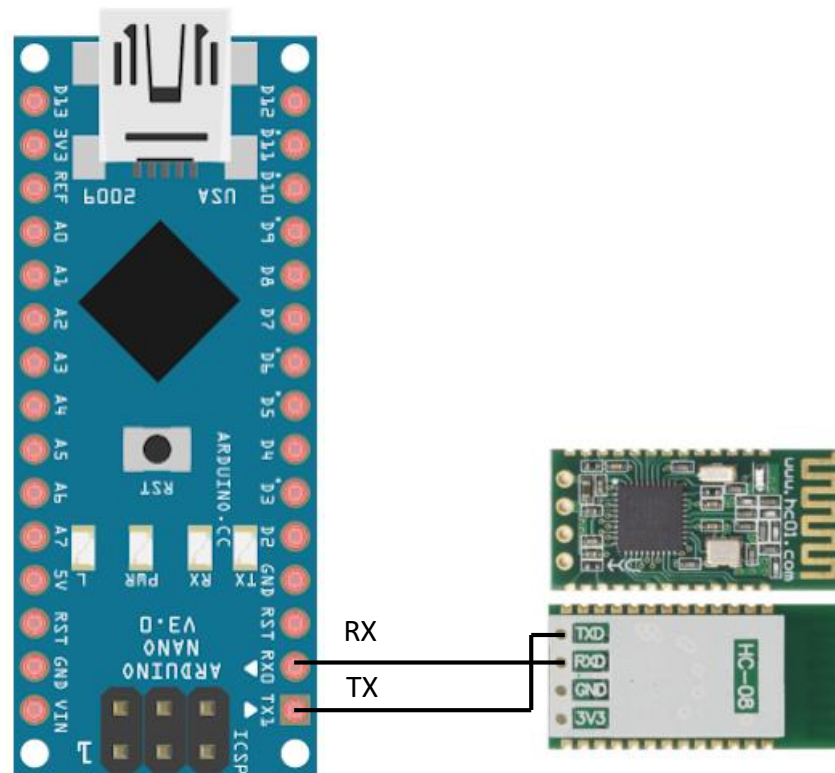
Compilado

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.

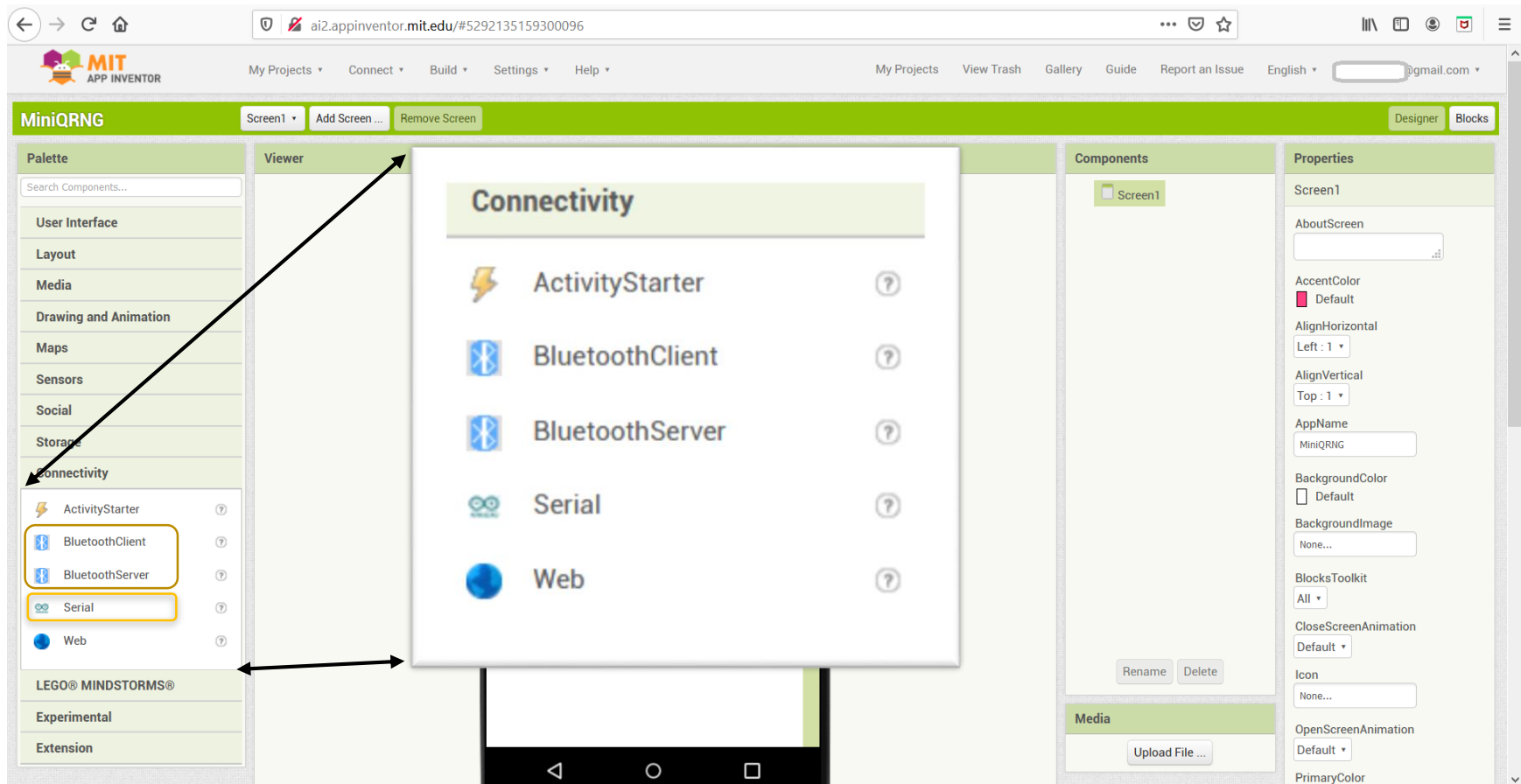
40 Arduino/Genuino Uno en COM4

Er zijn twee manieren om te communiceren met de lastige nano, een is via de Seriële poort en de andere is via een Bluetooth-verbinding.

Voor de bluetooth verbinding is het heel eenvoudig om de HC-08 module of een soortgelijke te kopen en deze als volgt aan te sluiten:



De volgende seriële of Bluetooth-componenten kunnen worden gebruikt om App Inventor met Arduino te verbinden:





Nu gecompileerd en geladen het programma QRNGv10.ino alleen ontbreken communiceren met de lastige nano om de gegevens op te slaan (quantum random nummers) deze zal worden in binaire formaat, maar de verkregen gegevens kunnen gemakkelijk worden doorgegeven aan een ander formaat zoals hexadecimale of decimale, afhankelijk van de uiteindelijke eis.

Tot slot, om een voorbeeld te zien van hoe de seriële of Bluetooth-verbinding werkt, zijn hier enkele referentiekoppelingen.

Vergeet niet dat alles is door middel van Blockly programmering worden getest met App Inventor dit heeft al blokken voor de communicatie met arduino seriële of andere blockly type systeem kan worden door middel van soortgelijke bluetooth online.

http://kio4.com/appinventor/9A0_bluetooth_RXTX.htm

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

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

Het gehele project van ontwerp en gebruik van QRNG (Quantum Random Number Generator) uitbreidingen bekijken. Bekijk de gebruikershandleiding op:

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



4. Wat is het bewijs van Quantum (PQu)?

PoQu. - "Proof of Quantum" is een consensus algoritme ontwikkeld voor Mini BlocklyChain en COINsolidation, deze test is een variant op de Proof of Work (PoW) die als volgt werkt.

De Test of Quantum (PoQu) bij het opstarten wordt uitgevoerd met hetzelfde algoritme als de "Test of Work" (PoW) is gebaseerd op het zetten van de processor van het apparaat (PC, Server, Tablet of Mobile Phone) om te werken aan een reeks van karakters die een wiskundige puzzel is genaamd een "hash".

Vergeet niet dat een "hash" een algoritme of wiskundig proces is dat bij het introduceren van een zin of een bepaald type digitale informatie zoals tekstbestanden, programma's, afbeeldingen, video, geluid of andere diverse soorten digitale informatie ons als resultaat een alfanumeriek karakter geeft dat de digitale handtekening vertegenwoordigt die het op een unieke en niet-reproduceerbare manier van de gegevens weergeeft, het hash-algoritme is unidirectioneel, dit betekent dat wanneer u een gegevens in te voeren om de handtekening "hash" zijn omgekeerde proces kan niet worden uitgevoerd, met een handtekening "hash" kunnen we niet weten welke informatie werd verkregen deze eigenschap geeft ons een veiligheid voordeel om de informatie die we sturen via het internet te verwerken. Hoe werkt het? Stel je voor dat het verzenden van elke vorm van informatie via niet-veilige kanalen en begeleiden met de respectieve "bron hash", de ontvanger bij de ontvangst van de informatie kan de "hash" van de ontvangen informatie zullen we het noemen "bestemming hash" en controleer het met de "bron hash" als beide "hashes" zijn hetzelfde kunnen we bevestigen dat de informatie niet is gewijzigd in het kanaal dat werd verzonden, is slechts een voorbeeld waar dit type van informatiebeveiliging proces wordt momenteel gebruikt.

Momenteel zijn er verschillende soorten algoritmen of hasjprocessen die verschillen in het beveiligingsniveau. De meest gebruikte of bekende zijn: MD5, SHA256 en SHA512.

Voorbeeld van SHA256:

We hebben een ketting of zin als volgt: "Mini BlocklyChain is modulair.

Als we een SHA256 hash op de vorige string toepassen, geeft dat ons de volgende hash.

f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db8



De bovenstaande alfanumerieke tekenreeks is de handtekening die de zin in het bovenstaande voorbeeld weergeeft

Zo kunnen we de site bijvoorbeeld ook op het internet gebruiken:

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

In het geval van het "Testwerk" (PoW) algoritme werkt het door gebruik te maken van rekenkracht om een vooraf gedefinieerde hash te verkrijgen.

Laten we ons voorstellen dat we de vorige "hasj" hebben die we uit de "Mini BlocklyChain is modulair" keten hebben gehaald.

f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db8

Aan deze "hash" in het begin zetten we de parameter van de moeilijkheid die is gewoon om nullen "0" in het begin, dat wil zeggen als we zeggen dat de moeilijkheid is van 4 het zal hebben "0000" + "hash" om dit zullen we het noemen "zaad hash"

0000 f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db8

Nu rekening houdend met het feit dat we de invoerinformatie kennen die de string is: "Mini BlocklyChain is modulair" voegen we aan het einde van de string een getal toe dat begint met nul "0" en we halen de hash eruit om het "hash nonce" te noemen:

f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db80

We hebben hasj nonce:

7529f3ad273fc8a9eff12183f8d6f886821900750bb6b59c1504924dfd85a7c8

Dan voeren we een vergelijking van de nieuwe "hash nonce" met de "hash seed" als ze gelijk zijn aan het knooppunt dat eerst de gelijkheid vindt zal de uitvoering van de verwerking van de huidige transactie te winnen. Zoals we kunnen zien is dit proces gebaseerd op de waarschijnlijkheid en de rekenkracht van het apparaat, wat de "Proof of Work" test een consensusgelijkheid geeft voor alle knooppunten.

Als de "hasj" niet samenvalt met de "hasj nonce", wordt de moeilijkheidsgraad met één verhoogd en wordt de "hasj nonce" weer verwijderd, het getal dat wordt verhoogd wordt het "nonce" getal genoemd, het wordt vergeleken met de "hasj" totdat ze samenvallen of gelijk zijn.

Zoals we kunnen zien is het getal "nonce" of verhoging het getal dat zal helpen om de "hasj" van gelijkheid te verkrijgen.



Gebaseerd op het "Test of Work" (PoW) algoritme, is het Quantum Test (PoQu) algoritme gebaseerd op het verkrijgen van het getal "nonce" zoals PoW dat doet en met behulp van een minimum moeilijkheidsgraad variërend van 1 tot 5, dit dient alleen om het recht van het mobiele apparaat om een kandidaat te zijn om een consensus te winnen.

De Quantum Test (PoQu), wordt geactiveerd wanneer de mobiele telefoon de minimale PoW heeft voltooid en de pass wint om een waarschijnlijkheidsnummer in het QRNG-systeem te verkrijgen.

De QRNG (Quantum Random Number Generator) is een Quantum Random Number Generator, dit systeem is gebaseerd op het genereren van echte willekeurige getallen op basis van kwantummechanica is vandaag de dag het veiligste systeem om dergelijke getallen te genereren. Voor meer details zie "Quantum Computation Security" in index 3.

COINsolidatie kan zowel minimale PoW- als PoQu-concessietypes implementeren.

De PoQu test is gebaseerd op het verkrijgen van het "nonce" nummer. Dit nummer in de PoQu test staat bekend als het "Magic Number" en hiermee zal het Peer to Peer systeem bevestigen of het nummer correct is en vervolgens zal een willekeurig nummer worden verkregen met de COINsolidation QRNG server pool. Dit willekeurige getal wordt in alle knooppunten geregistreerd, er wordt een lijst gemaakt met **((Node Addition) /2)) +1** en uit deze lijst wordt degene gekozen met het hoogste waarschijnlijkheidspercentage om de winnaar van de consensus te zijn (PoQu) en deze zal de huidige transactiewachtrij uitvoeren.

Het PoQu algoritme maakt ook gebruik van **NIST** (National Institute of Standards and Technology) testen om ons te verzekeren dat de willekeurige getallen in de QRNG echt willekeurige getallen zijn.

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

In COINsolidatie hebben we een blok voor PoW en een blok voor PoQu geïmplementeerd. Deze blokken gebruiken een hashtype: SHA256 voor vrij gebruik, voor commercieel gebruik hebben we een SHA512 en andere hashes zoals vereist.

Voor meer details over het concept HASH zie:

https://es.wikipedia.org/wiki/Funcion_hash

OPMERKING: De Test of Work (PoW) die in mobiele telefoons wordt gebruikt, kan slechts een maximale moeilijkheidsgraad van 5 gebruiken, aangezien de wiskundige verwerking van deze apparaten niet is toegewijd zoals servers of pc's. We gebruiken het PoW-algoritme alleen om uw pasje of toestemming te krijgen om het Quantum Random Number Generator



(QRNG)-systeem in te voeren en daarmee het Quantum Random Number Generator (PoQu)-algoritme uit te voeren. Zie gebruik van (PoQu) in Mini BlocklyChain:

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

5. Algoritme voor het creëren van een geconsolideerd universeel adres (CUA)

Onze CUA-adressen (Consolidated Universal Address) worden aangemaakt met behulp van het volgende algoritme:

Stap 1.- Identifiers worden verwijderd van de respectievelijke adressen, het zijn de alfanumerieke karakters die het adres identificeren van waaruit de blokken is gecreëerd.

Adres Bitcoin- Token - (OAP).

akXma4vqxvmEqnVAKSM953wYsnjNBhN3GM7

Adres Ethereum - Token COINsolidation - (ERC20).

0x9d08c0ac0f2fdf078c883db6fa617b15776e4b41

Stap 2.- De SHA512 (adres String-Text) van elk adres zonder de initiële identificatiecode wordt verkregen door de "a" van A1 en de "0x" van A2 te verwijderen en de laatste twee karakters van elke hashbewerking te nemen, gesymboliseerd met "U". Verificatienummers.

$U(\text{SHA512}(\text{kXma4vqxvmEqnVAKSM953wYsnjNBhN3GM7})) = \text{50}$

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

Stap 3.- De karakters van elk adres worden één voor één aaneengeschaakeld vanaf het adres dat minder karakters bevat, in geval van dezelfde hoeveelheid karakters kan de aaneenschakeling vanaf elk adres beginnen.

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

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

Aaneenschakeling van adressen:

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

****De laatste karakters die niet aaneengeschaakeld kunnen worden, worden aan het begin van de string gezet.**

6e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777



Stap 4.- Het aantal karakters dat in stap 3 zou kunnen worden samengevoegd, wordt toegevoegd aan het begin van de string die uit stap 3 voortvloeit.

66e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

Stap 5.- De twee paar verificateurs uit stap 2 van elke richting worden samengevoegd aan het begin van de keten die voortvloeit uit stap 3 in dezelfde volgorde A1 + A2.

50fb66e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

Stap 6.- De **CUAG** (Consolidated Universal Address Genesis) identificatie wordt geïntegreerd aan het begin van het in stap 5 gecreëerde adres.

cua50fb66e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

In het geval van Bitcoin en Ethereum adresconsolidatie geeft het een adres bestaande uit **82 hexadecimale karakters.

6. Algoritme voor dual consolidated address (DAC) en (HAC).

De oprichting van een DAC is hetzelfde als de CUA, het verschil is dat in de DAC's zij worden gebruikt om normale adressen te consolideren om transacties te ontvangen, deze adressen vertegenwoordigen geen cryptomonedas of token.

Stap 1.- Identifiers worden verwijderd van de respectievelijke adressen, het zijn de alfanumerieke karakters die het adres identificeren van waaruit de blokketen is gecreëerd.

18gYNA9c2G9X8HZ8QxWLpLXZauAxFnsJbe (Bitcoin-adres)

0x5d2Acdb34c279Aa6d1e94a77F7b18aB938BFb2bB (Dirección Ethereum)

Stap 2.- De SHA512 (adres String-Text) van elk adres zonder de initiële identificatiecode wordt verkregen door de "**1**" van A1 en de "**0x**" van A2 te verwijderen en de laatste twee karakters van elke hashbewerking te nemen, gesymboliseerd met "U". Verificatienummers.

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

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

Stap 3.- De karakters van elk adres worden één voor één aaneengeschakeld vanaf het adres dat minder karakters bevat, in geval van dezelfde hoeveelheid karakters kan de aaneenschakeling vanaf elk adres beginnen.

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



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

Aaneenschakeling van adressen:

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

****De laatste karakters die niet aaneengeschakeld kunnen worden, worden aan het begin van de string gezet.**

8BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3

Stap 4.- Het aantal karakters dat in stap 3 zou kunnen worden samengevoegd, wordt toegevoegd aan het begin van de string die uit stap 3 voortvloeit.

78BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3

Stap 5.- De twee paar verificateurs uit stap 2 van elke richting worden samengevoegd aan het begin van de keten die voortvloeit uit stap 3 in dezelfde volgorde A1 + A2.

48f378BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3

Stap 6.- De **DAC** (Dual Address Consolidated) identificatie wordt geïntegreerd aan het begin van het in stap 5 gecreëerde adres.

dac48f378BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3

****In het geval van Bitcoin en Ethereum adresconsolidatie geeft het een adres dat bestaat uit 81 hexadecimale karakters.**

In het geval van HAC (Hibric Address Consolidated) wordt toegepast in de vorige wat varieert zijn de adressen die worden gebruikt, in dit geval zullen we gebruik maken van een adres dat een actief (Cryptomonedas of token) en een normale standaard adres van de overdracht van activa van een bepaald type van blokken.

OPMERKING: De grootte van de CUA-, HAC- en DAC-adressen kan per geval verschillen, afhankelijk van de adressen waaruit ze bestaan.



Project en oplossing door COINsolidatie.

Momenteel zijn er verschillende soorten Blockchain gericht op activa van verschillende kenmerken, dit leidt tot een oneindig aantal soorten adressen van het dagelijks gebruik moeten houden een strakke controle om te voorkomen dat het maken van fouten in de overdracht.

Aan de andere kant is de wereld van cryptomoney en penningen beperkt tot financiële experts of in hun geval experts in blockchain technologie, dus het is moeilijk voor de gemiddelde persoon om zich te wagen aan de creatie van hun eigen cryptomoney of penning.

We hebben de twee eerdere problemen in COINsolidatie opgelost door de volgende punten en/of tools te maken die we hebben gecreëerd.

Voor het controlepunt van adressen van verschillende blockchains hebben we een algoritme gecreëerd waarbij het twee of meer adressen in hun verschillende combinaties consolideert (samenvoegt), waardoor een enkel adres van het type CUA, HAC en/of DAC ontstaat.

Bij deze oplossing wordt in plaats van het versturen van twee adressen uit dezelfde of een andere blokken, slechts één geconsolideerd adres gebruikt.

Voor het tweede probleem hebben we de programmeermethodologie Blockly gebruikt, dit is een visueel hulpmiddel waarbij geen grote kennis van programmeren nodig is en elke doorsnee persoon of bedrijf in staat zal zijn om zijn eigen applicaties te maken zonder dure ontwikkelingsteams, tijd en geld te hoeven investeren.

We hebben de uitbreidingen (modules) gemaakt om ze gewoon te installeren en te gebruiken om mobiele applicaties te maken, in 15 minuten. Voorbeeld van uw eigen crypto valuta wisselen of het ontwikkelen van uw eigen valuta (token) in minuten. Dit alles met behulp van state-of-the-art databeveiliging genaamd PQC (Post-Quantum Cryptografie).

Installeer de extensies gewoon op een gratis tool zoals Appventor, AppyBuider, Thunkable, Kondular of andere en binnen enkele minuten kun je de wereld van cryptomonies en tokencreatie in de palm van je hand betreden.

Tot slot creëert COINsolidatie het gebruik van goedkope kwantumbeveiliging (software en hardware) die gebruikt kan worden om computerdata thuis te beschermen. Op dit moment hebben de technologieën op basis van quantum computing en beveiliging hoge kosten die alleen bedrijven met een hoog financieel niveau kunnen maken en gebruiken. In COINsolidatie zijn we echter van mening dat nieuwe technologieën voor iedereen beschikbaar moeten zijn, de eerlijkheid van het gebruik van de Blockchain en Quantum

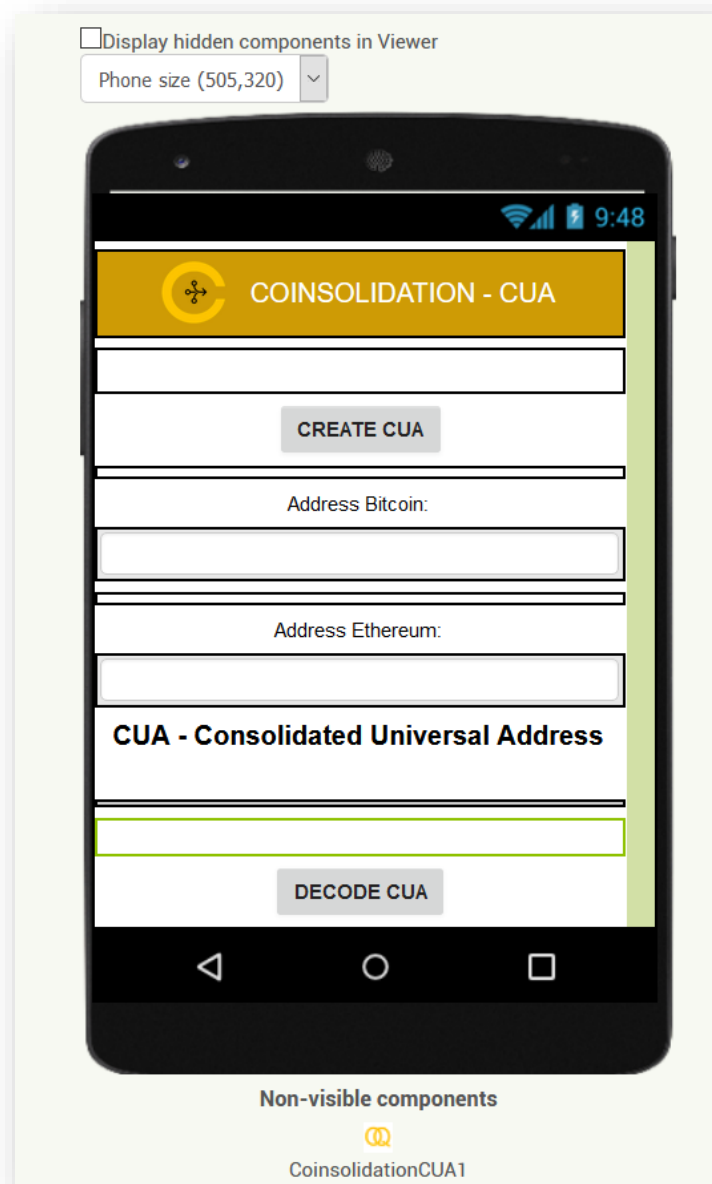


Computing voor iedereen moet zijn, we maken gratis software (cryptomonics) en goedkope hardware (quantum security).

7. Creatie van App CUA (Consolidated Universal Address) in 15 minuten.

*App voor Bitcoin- en Ethereummunten (BTC-ETH)

Ontwerp scherm 5 minuten in <https://appinventor.mit.edu/>





Gebruik van uitbreiding **CoinsolidatioCUA.AIX** (5 minuten).

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**



We maken de applicatie aan in **Menu > Build > App** (geef QR-code voor .apk) - (5 minuten).

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

do

bitcoinStr

ethereumStr

checkBitcoin

checkEthereum

set **addressBitcoin** . Text to

get **bitcoinStr**

set **addressEthereum** . Text to

get **ethereumStr**

set **verifyBitcoin** . Text to

get **checkBitcoin**

set **verifyEthereum** . Text to

get **checkBitcoin**

0

0

Show Warnings

CUA Progress Bar

35%

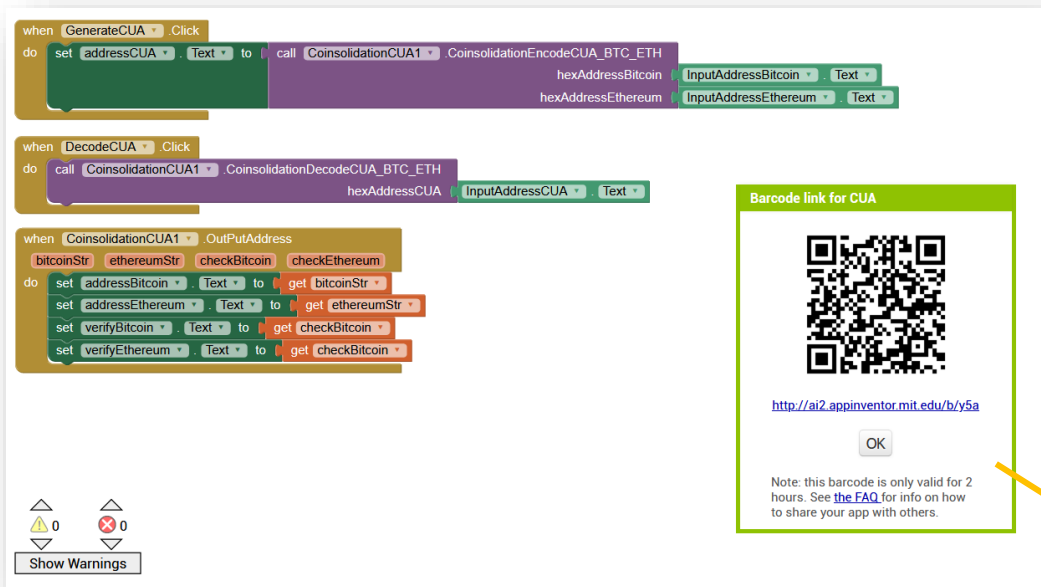
Compiling part 2 (please wait)

+

-

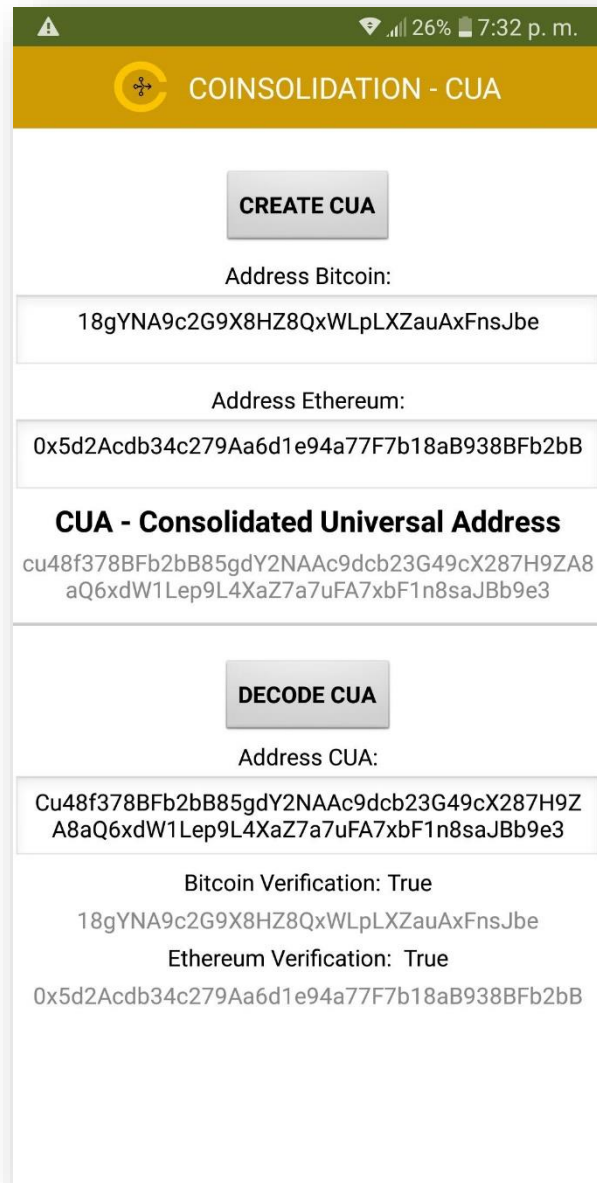


We hebben de applicatie op de mobiele telefoon geïnstalleerd vanuit de QR met behulp van ApplInventor's Android applicatie (MIT AI2 Companion) - <https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3>.



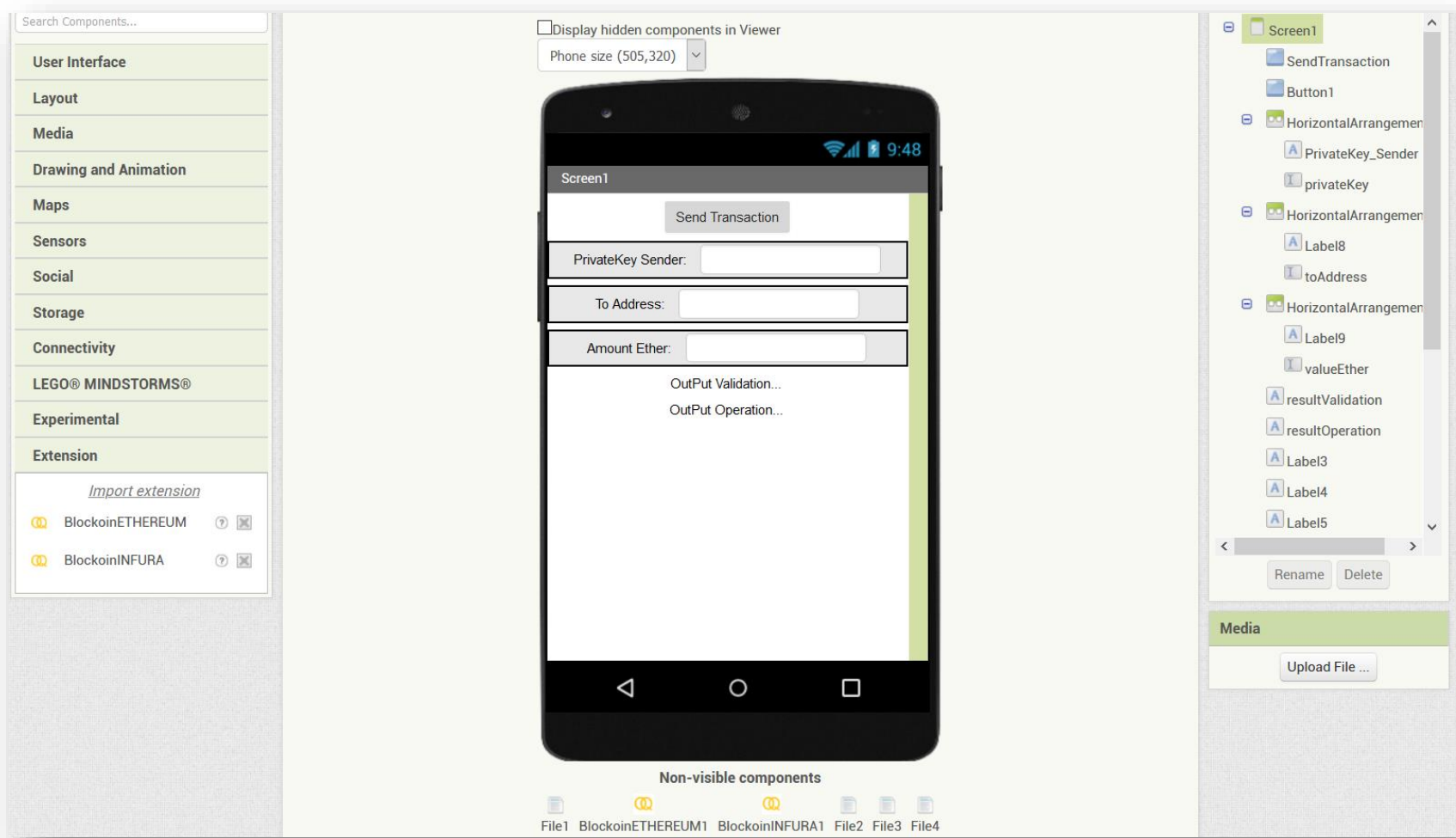
OPMERKING: De APK-bestandsapplicatie die klaar is om te worden geïnstalleerd bevindt zich in de volgende opslagplaats: <https://github.com/COINsolidation/App>.

Om de Java-code voor CUA-uitbreiding te herzien en een geconsolideerd universeel algoritme voor adresgeneratie te implementeren, moet de bijlage "Code voor CUA-algoritme" worden herzien of de code-link <https://github.com/COINsolidation/source> worden geraadpleegd.



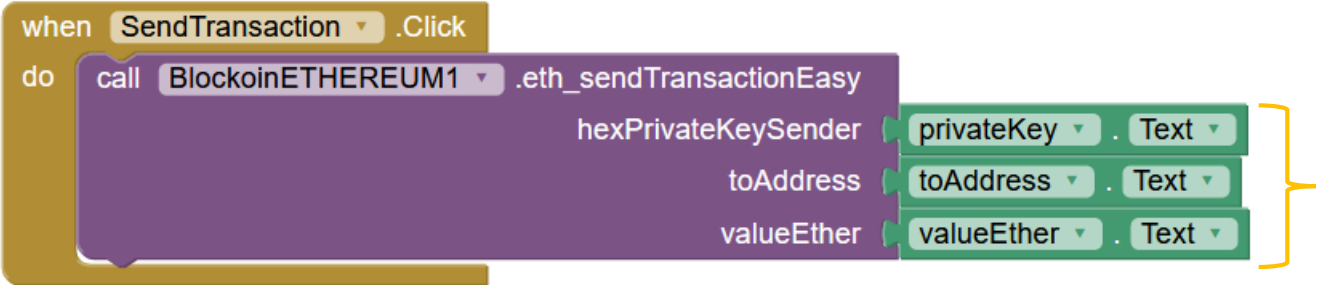


8. Maak uw Ethereum crypto valutawissel op Android in slechts 15 minuten.
Ontwerp in App Inventor (Scherm). - 5 minuten.





Funcatieblokken (eth_SendTransactionEasy) en event (OutPutSendTransactionEasy) - 5 minuten

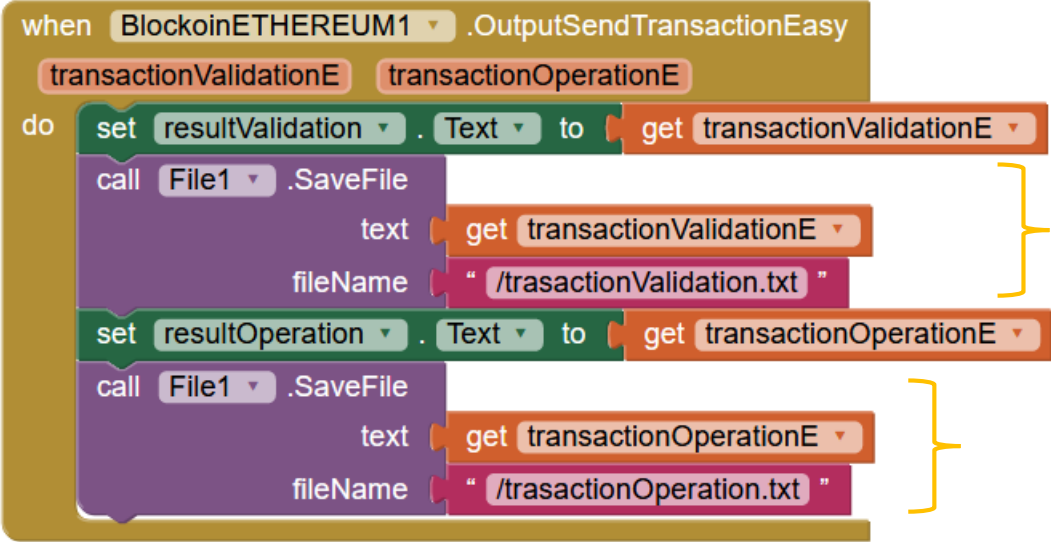


Voer de gegevens in:

PrivateKey: Primaire sleutel tot het adres van de afzender.

toAddress: Hexadecimaal adres van de ontvanger.

waardeEther: Geef de hoeveelheid Ether die wordt verzonden.



Sla de resultaten op in tekstbestanden:
Functie File1: File **trasactionValidation.txt**

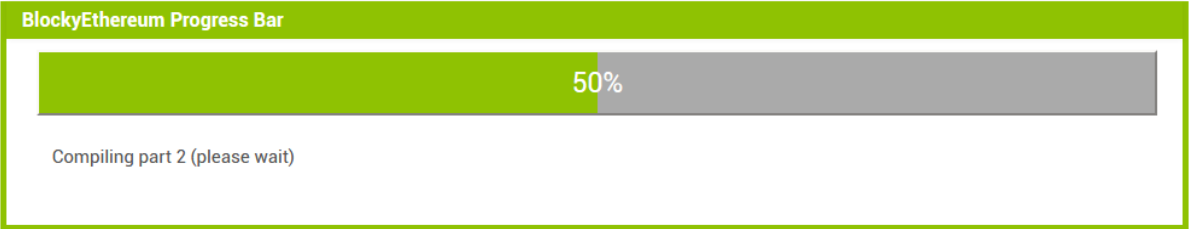
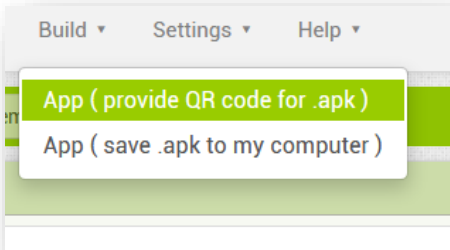
Sla de resultaten op in tekstbestanden:
File2-functie: Dossier **trasactionValidation.txt**

**Meer details zie Gebruikershandleiding Ethereumuitwisseling (EEE) Uitbreiding in de 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>



We compileren, genereren APK-bestand om het te installeren op het Android-toestel. - 5 minuten



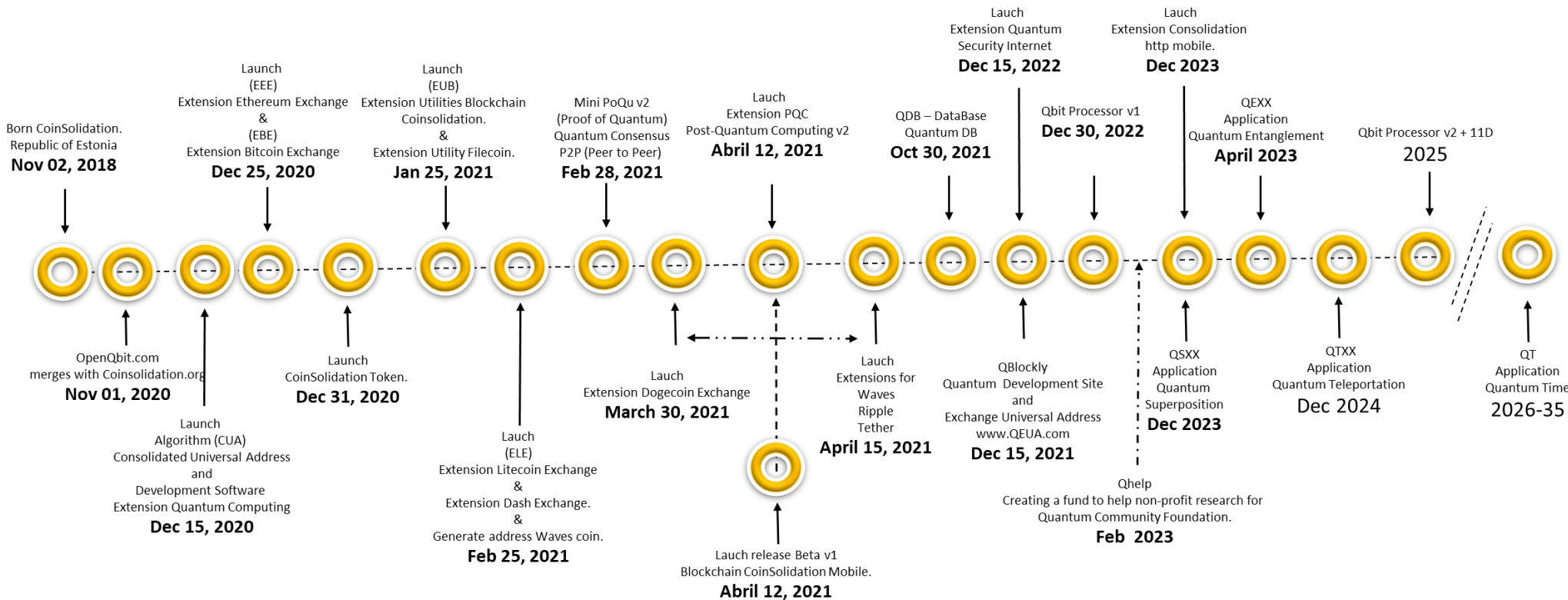
OPMERKING: Wanneer de transactie wordt uitgevoerd, duurt het ongeveer 6 tot 8 seconden om de knop "Transactie verzenden" los te laten. Vanwege de verbindingstijd met het Ethereum-netwerk.

Voor meer details over de uitbreiding van de EER - (uitbreiding van de Ethereumbeurs). Zie de EEE gebruikershandleiding in de link:
<https://github.com/COINsolidation/UserGuide>



9. Stappenplan COINsolidatie.

ROADMAP



*OpenQbit.com fuseert met COINsolidation.org (01 nov 2020) / OpenQbit is gespecialiseerd in Quantum Computing en Security Quantum.

*Quantum processor versie 1 zal gebruik maken van basis kwantum logische poorten voor thuisgebruik.



EXchange
tensions

10.COINsolidatie Token (CUAG) - ICO DISTRIBUTION PLAN.

De ICO is verdeeld in drie fasen:

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 ADDRESS
0xa646c054394f85257E18D56Cf5c6b5E603447470	COINsolidation WERKINGSADRES



11. Algemene kenmerken van de COINsolidatiepenning:

Gemaakt door: Lugu Samaya.

Naam: COINsolidatie

Symbool: CUAG - (Consolidated Universal Address Genesis).

Type: NFT

Totaal aantal gecreëerde penningen: 40.000.000.000,00

Aantal decimalen: 18

Lanceringsland: Estland

Officiële website: www.COINsolidation.org

Bedrijf: COINsolidation International.

Lanceringsdatum: 30 december 2020

Consensusalgoritme: PQu (Bewijs van Quantum)

Adresalgoritme: Geconsolideerd universeel adres (CUA).

Gebruikte beveiliging: PQC (Post-Quantum Cryptografie) gebaseerd op quantum computing.

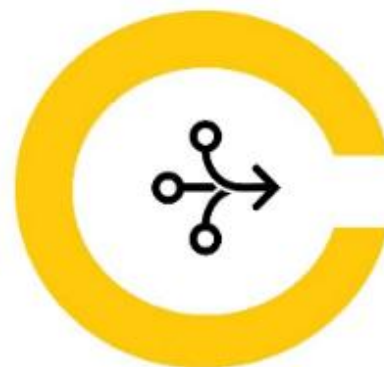
Technologisch voorstel: Uitbreidingen voor Blockly-systemen om cryptomonaden te gebruiken en implementatie van kwantumbeveiliging.

Technologiepartnerschappen of -overeenkomsten (fusie):

Bedrijf: OpenQbit Inc.

Industrie: Quantum computing en PQC (Post-Quantum Cryptografie).

Officiële website: www.OpenQbit.com





12. Basisconcepten toegepast in Blockchain-platforms.

Wat is een blokketen?

De blokketen wordt over het algemeen geassocieerd met Bitcoin en andere crypto valuta, maar deze zijn slechts het topje van de ijsberg, omdat het niet alleen wordt gebruikt voor digitaal geld, maar kan worden gebruikt voor alle informatie die een waarde kan hebben voor gebruikers en/of bedrijven. Deze technologie, die haar oorsprong vindt in 1991, toen Stuart Haber en W. Scott Stornetta het eerste werk aan een keten van cryptografisch beveiligde blokken beschreef, werd pas in 2008 opgemerkt, toen het populair werd met de komst van de bitcoin. Maar op dit moment wordt het gebruik ervan in andere commerciële toepassingen gevraagd en de verwachting is dat het op middellange termijn zal groeien in verschillende markten, zoals financiële instellingen of het internet van de dingen aan het internet van de dingen, naast andere sectoren.

De blokketen, beter bekend onder de term blokketen, is een enkel, overeengekomen record dat verdeeld is over verschillende knooppunten (elektronische apparaten zoals pc's, smartphones, tablets, etc.) in een netwerk. In het geval van crypto-valuta's kunnen we het zien als het boekje waarin elk van de transacties wordt geregistreerd.

De werking ervan kan complex zijn om te begrijpen als we ingaan op de interne details van de uitvoering ervan, maar het basisidee is eenvoudig te volgen.

Het wordt opgeslagen in elk blok:

- 1.- een aantal geldige records of transacties,
- 2.- informatie over dat blok,
- 3.- het verband met het vorige blok en het volgende blok door de hash van elk blok –unieke code die zou zijn als de vingerafdruk van het blok.

Daarom heeft **elk blok** een **specifieke en onbeweeglijke plaats binnen de keten**, aangezien elk blok informatie bevat uit de hash van het vorige blok. De hele keten wordt opgeslagen op elk netwerkknoppunt dat de blokketen vormt, dus **een exacte kopie van de keten wordt op alle deelnemers aan het netwerk opgeslagen**.

Wat is een adres of account binnen het blokketen-ethereumplatform?

Het is een reeks van 42 karakters in het Ethereum-platform die een getal in hexadecimale basis vertegenwoordigt, waar de in het Ethereum gedefinieerde activa zullen worden



gedeponeerd of verzonden. In andere blokkenplatforms kan het aantal karakters van de account of het adres bijvoorbeeld verschillen:

0x5d2Acdb34c279Aa6d1e94a77F7b18aB938BFb2bB

Wat is een kryptomoney?

Het is een digitale of virtuele valuta die is ontworpen om te functioneren als een medium voor uitwisseling. Het maakt gebruik van cryptografie (digitale beveiliging) om transacties te beveiligen en te verifiëren, en om de creatie van nieuwe eenheden van een bepaalde kryptomoney te controleren.

Wat is een penning?

Munten zijn digitale middelen die binnen een bepaald projectecosysteem kunnen worden gebruikt.

Het belangrijkste onderscheid tussen penningen en crypto-currencies is dat de eerste een ander blokkenplatform (niet hun eigen) nodig hebben om te kunnen werken. Ethereum is het meest voorkomende platform voor het maken van penningen, voornamelijk vanwege de slimme contractfunctie. De penningen die op de blokken Ethereum worden gemaakt, zijn over het algemeen bekend als ERC-20 penningen, hoewel er andere, meer gespecialiseerde soorten penningen zijn, zoals de ERC-721 penningen die voornamelijk worden gebruikt voor verzamelobjecten (kaarten, gebruik in videospelletjes, kunstwerken, enz.).

Wat is een beurs?

Een crypto-valutawissel is het ontmoetingspunt waar het wisselen van crypto-valuta plaatsvindt in ruil voor fiatgeld of andere crypto-valuta. In deze online beurshuizen wordt de marktprijs gegenereerd die de waarde van de cryptomonen aangeeft op basis van vraag en aanbod.

Wat is Exchange Rates?

Dit zijn de koersen van de waarde van een Ether of andere crypto-valuta in de circulerende valuta van elk land. Bijvoorbeeld, op de dag van de creatie van deze handleiding heeft een Ether een waarde in Amerikaanse dollars van \$430,94

Wat is een transactie?

Het is de uitvoering of overdracht van een bepaald type niet-materieel actief dat binnen het ethersysteem een vooraf vastgestelde waarde kan krijgen en dat later kan worden omgezet in een materiële waarde voor een bedrijf of persoon.



Wat is txHash?

Het is een hexadecimaal getal dat helpt om het resultaat van elke transactie in detail te volgen.

Welke soorten transacties zijn er?

Je hebt twee soorten, een is de transactie "offline" dit creëert zonder de noodzaak om verbinding te hebben met het hoofdnetwerk van Ethereum kan worden opgeslagen totdat u ervoor kiest om verbinding te maken met het netwerk van Ethereum en laat de transactie, hebben het voordeel van de veiligheid, omdat de hele transactie wordt verwerkt offline die voorkomt dat elke anomalie die zou kunnen worden in de netwerkverbinding. De andere transactie is de "online" transactie die altijd verbonden moet zijn met het internet met de veiligheidsvoordelen en -nadelen die het met zich meebrengt.

Wat is een Blockchain adres?

Een adres of account bestaat uit drie delen, het adres, de openbare sleutel en de privésleutel, deze twee sleutels zijn een reeks getallen en karakters in hexadecimaal formaat die worden gebruikt om te verzenden en te ontvangen (actief) of ether (digitale valuta).

De primaire sleutel mag nooit met iemand worden gedeeld, omdat het de sleutel is die de vrijgave van het saldo (de transacties) op de rekening toestaat.

De publieke sleutel is bekend bij het hele publiek en wordt met iedereen gedeeld, omdat het de referentie is om te bevestigen dat de transactie correct is, zowel in termen van waarde als naar wie hij wordt gestuurd.

Voorbeelden van componenten voor het beheer van het ethereumnetwerk:

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



13. Wat is Blockly-programmering?

Blockly is een **visuele programmeermethode** die bestaat uit een eenvoudige set commando's die we kunnen combineren alsof het de stukjes van een puzzel zijn. Het is een zeer nuttig hulpmiddel voor degenen die willen **leren programmeren op** een intuïtieve en eenvoudige manier of voor degenen die al weten hoe ze moeten programmeren en het potentieel van dit soort programmering willen zien. Het is gebaseerd op de JavaScript-taal en is ontwikkeld door het bedrijf Google en MIT.

Blockly is een vorm van programmeren waarbij je geen enkele achtergrond in welke computertaal dan ook nodig hebt, dit is omdat het gewoon het aansluiten van grafische blokken is alsof we lego of een puzzel spelen, je moet gewoon wat logica hebben en dat is het!

Iedereen kan programma's voor mobiele telefoons (smartphones) maken zonder te knoeien met die programmeertalen die moeilijk te begrijpen zijn, gewoon blokken op een grafische manier in elkaar zetten op een eenvoudige, gemakkelijke en snelle manier om te maken.

14. Bijlage "Code voor CUA-algoritme".

Verwijzing naar Github: <https://github.com/coinsolidation/source>

15. Voorwaarden.

Gebruiksvoorwaarden zie op de site www.coinsolidation.org of <https://github.com/coinsolidation/Terms>.

Ondersteuning bij commercieel gebruik.
support@coinsolidation.org

Verkoopblokken zakelijk gebruik.
sales@coinsolidation.org

Juridische informatie en vragen of problemen met betrekking tot vergunningen
legal@coinsolidation.org

Sociale netwerken:

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

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