



EXchange  
tensions

## Consolidation du COIN.

Tallinn, Estonie. (Résidence électronique)

# White Paper - Livre blanc.

version 1.0.0

Décembre 2020.

COINsolidation.org est une marque déposée de COINsolidation International, sous licence d'utilisation libre et commerciale. Conditions d'utilisation à l'adresse suivante :

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

COINsolidation International a fusionné avec [www.OpenQbit.com](http://www.OpenQbit.com) pour une coopération technologique basée sur la mécanique quantique (Quantum Security & Quantum Computing). Cette fusion permet l'utilisation, le partage et la réingénierie de la technologie développée par OpenQbit Inc. (Estonie, résidence électronique)

## Contenu

1. Introduction. ....	3
2. L'informatique quantique de sécurité.....	6
3. Création du dispositif "Hardware" d'un QRNG (Quantum Random Number Generator). ....	11
4. Qu'est-ce que la preuve de quantum (PQu) ?.....	17
5. Algorithme pour la création d'une adresse universelle consolidée (AUC) .....	20
6. Algorithme pour la double adresse consolidée (DAC) et (HAC).....	21
Projet et solution par COINsolidation. ....	23
7. Création de l'App CUA (Consolidated Universal Address) en 15 minutes. ....	24
8. Créez votre crypto-bourse Ethereum sur Android en seulement 15 minutes.....	28
9. Feuille de route COINsolidation. ....	31
10. COINsolidation Token (CUAG) - PLAN DE DISTRIBUTION DES ICOS. ....	32
11. Caractéristiques générales du jeton de consolidation COIN :.....	33
12. Concepts de base appliqués dans les plateformes Blockchain. ....	34
13. Qu'est-ce que la programmation en blockly ? .....	37
14. Annexe "Code pour l'algorithme CUA". ....	37
15. Conditions. ....	37

## 1. Introduction.

Actuellement, les fusions sont à jour, que ce soit pour un bien économique, technologique ou commercial.

Nous présentons le premier modèle de crypto-fusion ou crypto-tokens qui offre une sauvegarde entre deux cryptomonades, jetons ou un mélange de ceux-ci, basé sur un algorithme pour créer une adresse consolidée qui est utilisée et générée dans l'environnement COINsolidation.

Nous avons créé trois types d'adresses consolidées.

L'adresse universelle consolidée (**CUA**) est utilisée pour consolider et créer un nouveau jeton (actif) à utiliser par l'utilisateur. La combinaison peut être de trois types : Cryptocurrency-Cryptocurrency, Cryptocurrency-Token ou Token-Token. Dans le cas du CUA, il est formé par une relation Token-Token.

Le **HAC** (Hibric Address Consolidated) est utilisé lorsque nous devons consolider une adresse concernant une devise et/ou un jeton de cryptologie et une adresse normale pour le transfert d'actifs.

Le **DAC** (Dual Address Consolidated) est utilisé pour gérer et consolider deux adresses normales provenant de la même chaîne de blocs ou de deux technologies différentes.

Commençons par examiner les avantages du CUA.

Une adresse CUA se compose de l'adresse du jeton de consolidation COIN (adresse statique) et d'un jeton supplémentaire connu sous le nom de "Colored Coin" (adresse variable). Dans ce cas, nous pouvons voir que les adresses CUA seront toujours formées par les adresses d'une sorte de combinaison d'actifs (Cryptosolides ou jetons).

Dans notre cas, lorsque nous consolidons le jeton COINsolidation et un jeton OAP, nous le connaissons sous le nom de "CUA genesis" ou **CUAG (Consolidated Universal Address Genesis)**.

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

Le jeton "Colored coin" est basé sur et créé par la *chaîne de blocs Bitcoin* et utilise la norme Open Asset Protocol (**OAP**).

Commençons par examiner le potentiel et les avantages de la consolidation des adresses.

- I. Pour les utilisateurs qui créent un CUA, il sera possible de créer un token (**OAP**) qui peut être personnalisé par l'utilisateur qui a créé le CUA, l'utilisateur aura la possibilité d'avoir son propre token ou crypto actif afin qu'il puisse l'utiliser dans la création, le soutien ou l'expansion de son (ses) entreprise(s), d'une manière simple et facile il aura un atout dans le monde des crypto-tokens.
- II. Les entreprises qui créent un CUA peuvent disposer d'un jeton (**OAP**) qu'elles peuvent utiliser pour créer de la valeur dans leur chaîne d'approvisionnement ou utiliser l'actif dans des transactions de liquidité basées sur le soutien économique de leurs actifs et passifs d'entreprise.
- III. Pour les cryptomonades et les jetons existants, en créant une CUA, ils pourront utiliser votre adresse qui identifie votre actif et en la consolidant avec le jeton (**OAP**), ils pourront accroître leur demande en offrant à leurs investisseurs actuels et futurs leur propre jeton pour leurs utilisateurs.

Exemple de **CUAG**, nous avons les adresses respectives de deux Blockchain différents :

Adresse Bitcoin- Token - (OAP).

**akXma4vqxvmEqnVAKSM953wYsnjNBhN3GM7**

Adresse Ethereum - Token COINsolidation - (ERC20).

**0x8390f8abb8fd8ad3bf8457db59f2ed75e015d303**

En appliquant un algorithme pour consolider les adresses précédentes, nous obtenons l'adresse CUA.

**cua50d0615d303k8X3m9a04fv8qaxbvb8Efqn8VaAdK3SbMf985435w7Ydsbn5j9NfB2heNd37G5Me70**

\* Pour plus de détails sur l'algorithme, voir la section 7 - "Algorithme pour la création d'une adresse universelle consolidée".

Nous avons donc une direction unique qui représente deux technologies différentes provenant de deux directions différentes consolidées dans une seule direction.

Nous en tenons compte dans le domaine de la rentabilité et de l'expansion financière de manière simple et directe en investissant dans l'un des jetons qui intègre notre CUA ; immédiatement, vous obtiendrez un jeton basé sur la chaîne de blocs Bitcoin (OAP).

Examinons maintenant deux lignes directrices que nous avons également proposées dans COINsolidation pour le monde des cryptomontages et/ou des jetons.

COINsolidation token est le projet qui vise à consolider les adresses et à disposer d'un support immédiat pour obtenir un jeton personnalisé à utiliser dans la croissance de chaque

utilisateur dans le monde des cryptomontages. Le projet est né en 2018 avec un groupe d'ingénieurs et de financiers intéressés par la fusion des secteurs financier et technologique, pour profiter des fonds d'investissement et occuper des technologies innovantes comme l'informatique quantique pour donner de la sécurité aux actifs, ainsi que l'objectif d'utiliser des outils qui pourraient être accessibles à tous.

Après une évaluation de plusieurs possibilités de développement, nous avons choisi l'option de la méthodologie de programmation visuelle Blockly. Cette méthodologie est basée sur l'utilisation d'extensions ou de modules (programmes en langage de programmation java) avec des fonctionnalités simples mais puissantes pour étendre l'activité de crypto à toute personne, pour cela nous devons couvrir les points suivants :

- ✓ a.- Retour sur investissement financier immédiat pour les utilisateurs, les investisseurs et les actifs en pouvant créer un actif non tangible (jetons personnels) à l'usage exclusif du créateur et de l'utilisateur du (CUA)
- ✓ b.- Nous utilisons les avantages de la jonction de deux chaînes de blocs au choix de l'utilisateur pour accroître les investissements actuels et futurs sur le marché de la cryptoactivité en utilisant le (CUA).
- ✓ c.- Faciliter la gestion d'adresses distinctes en les regroupant en (DAC).
- ✓ d.- Créer et utiliser une sécurité basée sur l'informatique quantique.

Le défi a commencé dans la création de la technologie des extensions suffisamment modulaires en fonctionnalité et en "taille" ; cette dernière a été le défi de l'équipe de développement de COINsolidation depuis les extensions qui sont utilisées dans la méthodologie Blockly et les systèmes de ce type (AppInventor, AppyBuilder, Thunkable, Kondular, etc) sont généralement des extensions (programmes) créées qui ne dépassent pas 100k - 300k octets, avec les restrictions qui ont dans leur taille la tâche de créer des extensions pour l'utilisation dans la Blockchain actuelle ont été pratiquement impossible en raison des bibliothèques qui sont utilisées dans leur création dépassent entre 10MB et 35MB ces tailles pour les outils actuels Les systèmes Blockly ne sont pas fonctionnels pour les utiliser.

L'équipe a dû créer, adapter et minimiser la méthodologie de programmation et les bibliothèques afin d'obtenir les extensions avec la fonctionnalité, la sécurité et la taille optimales.

Après presque deux ans de développement et de tests, nous avons terminé la première chaîne de blocs "bêta" utilisant des extensions pour Blockly, y compris l'algorithme de consensus "Proof of Quantum" utilisant la sécurité quantique pour l'échange de cryomoney.

Actuellement, nous disposons d'une chaîne de blocs propriétaire qui a été mise en service pour des tests "bêta" et d'ici la fin 2021, nous mettrons en service la version de production pour la distribution d'informations. Actuellement, notre jeton de consolidation COIN est basé sur les chaînes de blocs Ethereum et Bitcoin, cette dernière permettant la création de jetons personnalisés pour les utilisateurs.

## 2. L'informatique quantique de sécurité.

### Comment fonctionne l'informatique quantique ? <sup>(2)</sup>

La transformation numérique entraîne des changements dans le monde plus rapides que jamais. Croyez-vous que l'ère numérique est sur le point de prendre fin ? La **culture numérique** a déjà été identifiée comme un domaine dans lequel il est urgent de disposer de connaissances ouvertes et de possibilités accessibles d'apprentissage de la technologie afin de combler les lacunes en matière de développement social et économique. L'apprentissage des concepts clés de l'ère numérique deviendra encore plus crucial avec l'arrivée imminente d'une autre nouvelle vague technologique capable de transformer les modèles existants avec une rapidité et une puissance étonnantes : les **technologies quantiques**.

Dans cet article, nous comparons les concepts de base de l'informatique traditionnelle et de l'informatique quantique ; et nous commençons également à explorer leur application dans d'autres domaines connexes.

Que sont les technologies quantiques ?

Tout au long de l'histoire, les êtres humains ont développé la technologie à mesure qu'ils ont compris comment la nature fonctionne grâce à la science. Entre 1900 et 1930, l'étude de certains phénomènes physiques encore mal compris a donné naissance à une nouvelle théorie physique, la **mécanique quantique**. Cette théorie décrit et explique le fonctionnement du monde microscopique, l'habitat naturel des molécules, des atomes ou des électrons. Grâce à cette théorie, non seulement il a été possible d'expliquer ces phénomènes, mais il a également été possible de comprendre que la réalité subatomique fonctionne d'une manière totalement contre-intuitive, presque magique, et que dans le monde microscopique se produisent des événements qui ne se produisent pas dans le monde macroscopique.


Ces **propriétés quantiques** comprennent la superposition quantique, l'intrication quantique et la téléportation quantique.

- La **superposition quantique** décrit comment une particule peut se trouver dans différents états en même temps.
- L'**intrication quantique** décrit comment deux particules aussi éloignées l'une de l'autre que souhaité peuvent être corrélées de telle manière que, lors de leur interaction avec l'une d'entre elles, l'autre en soit consciente.
- La **téléportation** quantique utilise l'enchevêtrement quantique pour envoyer des informations d'un endroit à un autre dans l'espace sans avoir à le traverser.

Les technologies quantiques sont basées sur ces propriétés quantiques de nature subatomique.

Dans ce cas, la compréhension du monde microscopique par la mécanique quantique nous permet aujourd'hui d'inventer et de concevoir des technologies capables d'améliorer la vie des gens. Il existe de nombreuses et très différentes technologies qui utilisent les phénomènes quantiques et certaines d'entre elles, comme les lasers ou l'imagerie par résonance magnétique (IRM), sont présentes depuis plus d'un demi-siècle. Cependant, nous assistons actuellement à une révolution technologique dans des domaines tels que l'informatique quantique, l'information quantique, la simulation quantique, l'optique quantique, la métrologie quantique, les horloges ou les capteurs quantiques.

Qu'est-ce que l'informatique quantique ? Tout d'abord, il faut comprendre l'informatique classique.



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

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

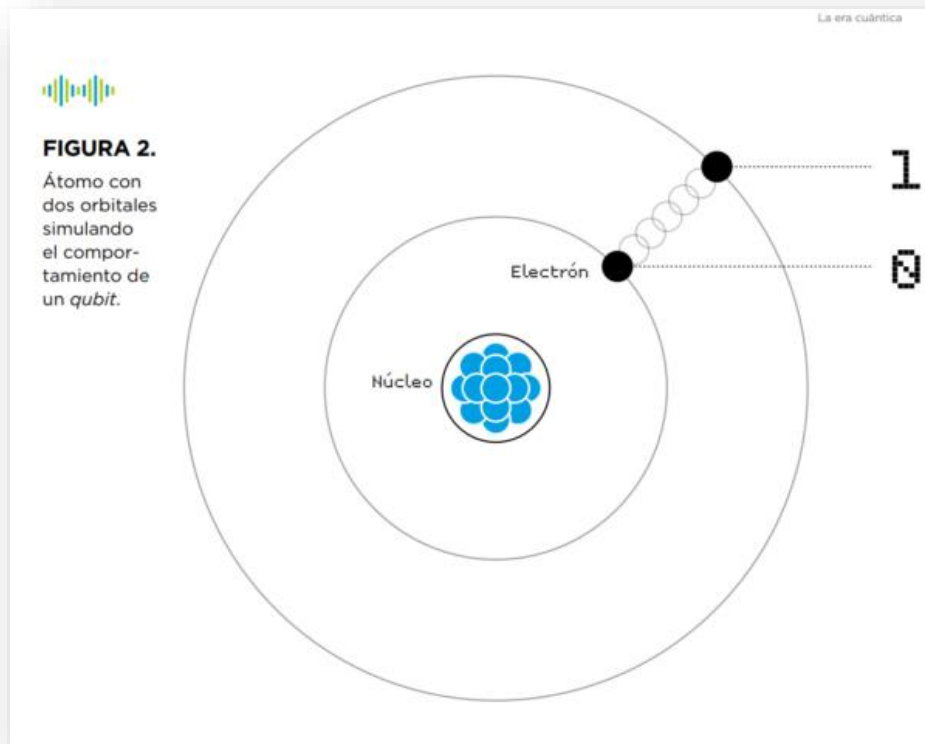
Pour comprendre le fonctionnement des ordinateurs quantiques, il convient d'abord d'expliquer comment fonctionnent les ordinateurs que nous utilisons tous les jours, que nous appellerons dans ce document ordinateurs numériques ou classiques. Ceux-ci, comme le

reste des appareils électroniques tels que les tablettes ou les téléphones portables, utilisent les bits comme unités fondamentales de la mémoire. Cela signifie que les programmes et les applications sont encodés en bits, c'est-à-dire en langage binaire de zéros et de uns. Chaque fois que nous interagissons avec l'un de ces dispositifs, par exemple en appuyant sur une touche du clavier, des chaînes de zéros et de uns sont créées, détruites et/ou modifiées à l'intérieur de l'ordinateur.

La question intéressante est de savoir ce que sont ces zéros et ces uns physiquement à l'intérieur de l'ordinateur. Les états zéro et un correspondent à un courant électrique qui circule, ou non, à travers des parties microscopiques appelées transistors, qui agissent comme des interrupteurs. Lorsqu'aucun courant ne circule, le transistor est "off" et correspond au bit 0, et lorsqu'il circule, il est "on" et correspond au bit 1.

Plus simplement, c'est comme si les bits 0 et 1 correspondaient à des trous, de sorte qu'un trou vide est un bit 0 et un trou occupé par un électron est un bit 1. C'est pourquoi ces appareils sont appelés électroniques. À titre d'exemple, la figure 1 montre l'écriture binaire de certains caractères. Maintenant que nous avons une idée du fonctionnement des ordinateurs d'aujourd'hui, essayons de comprendre comment fonctionnent les quanta.

### Des bits aux qubits





L'unité fondamentale de l'information dans l'informatique quantique est le bit ou qubit quantique. Les qubits sont, par définition, des systèmes quantiques à deux niveaux - nous en verrons des exemples ici - qui, comme les bits, peuvent être au niveau bas, qui correspond à un état de faible excitation ou d'énergie défini comme 0, ou au niveau haut, qui correspond à un état d'excitation plus élevé ou défini comme 1. Cependant, et c'est là que réside la différence fondamentale avec le calcul classique, les qubits peuvent également se trouver dans n'importe lequel des états intermédiaires infinis entre 0 et 1, comme un état qui est la moitié de 0 et la moitié de 1, ou les trois quarts de 0 et un quart de 1.

Les algorithmes quantiques, un calcul exponentiellement plus puissant et plus efficace

L'objectif des ordinateurs quantiques est de tirer parti de ces propriétés quantiques des *qubits*, en tant que systèmes quantiques, afin d'exécuter des algorithmes quantiques qui utilisent le chevauchement et l'entrelacement pour fournir une puissance de traitement beaucoup plus importante que les classiques. Il est important de souligner que le véritable changement de paradigme ne consiste pas à faire la même chose que les ordinateurs numériques ou classiques - les actuels - mais plus rapidement, comme on peut le lire dans de nombreux articles, mais que les algorithmes quantiques permettent d'effectuer certaines opérations d'une manière totalement différente qui, dans de nombreux cas, s'avère plus efficace - c'est-à-dire en beaucoup moins de temps ou en utilisant beaucoup moins de ressources informatiques -.

Voyons un exemple concret de ce que cela implique. Imaginons que nous soyons à Bogota et que nous voulions connaître le meilleur itinéraire pour se rendre à Lima parmi un million de possibilités pour y arriver ( $N=1.000.000$ ). Afin d'utiliser les ordinateurs pour trouver la voie optimale, nous devons numériser 1 000 000 d'options, ce qui implique de les traduire en langage binaire pour l'ordinateur classique et en *qubits pour l'ordinateur* quantique. Alors qu'un ordinateur classique devrait analyser un par un tous les chemins jusqu'à trouver celui qui lui convient, un ordinateur quantique tire parti du processus connu sous le nom de parallélisme quantique qui lui permet de considérer tous les chemins en même temps. Cela implique que, alors que l'ordinateur classique a besoin de l'ordre de  $N/2$  étapes ou itérations, c'est-à-dire 500 000 tentatives, l'ordinateur quantique trouvera le chemin optimal après seulement des opérations  $\sqrt{N}$  sur le registre, soit 1 000 tentatives.

Dans le cas précédent, l'avantage est quadratique, mais dans d'autres cas, il est même exponentiel, ce qui signifie qu'avec  $n$  *qubits*, nous pouvons obtenir une capacité de calcul équivalente à  $2^n$  bits. Pour illustrer ce point, il est courant de compter qu'avec environ 270 qubits, nous pourrions avoir plus d'états de base dans un ordinateur quantique - plus de chaînes de caractères différentes et simultanées - que le nombre d'atomes dans l'univers, qui est estimé à environ  $10^{28}$ . Un autre exemple est qu'on estime qu'avec un ordinateur

quantique de 2000 à 2500 *qubits*, nous pourrions casser pratiquement toute la cryptographie utilisée aujourd'hui (la cryptographie dite à clé publique).

Pourquoi est-il important de connaître la technologie quantique ?

Nous sommes dans un moment de transformation numérique où les différentes technologies émergentes telles que la chaîne de blocs, l'intelligence artificielle, les drones, l'Internet des objets, la réalité virtuelle, les imprimantes 5G, 3D, les robots ou les véhicules autonomes sont de plus en plus présentes dans de multiples domaines et secteurs. Ces technologies, appelées à améliorer la qualité de vie de l'être humain en accélérant le développement et en générant un impact social, progressent aujourd'hui de manière parallèle. Il est rare de voir des entreprises développer des produits qui exploitent des combinaisons de deux ou plusieurs de ces technologies, comme la chaîne de blocs et l'IdO ou les drones et l'intelligence artificielle. Bien qu'elles soient destinées à converger, générant ainsi un impact exponentiellement plus important, le stade initial de développement dans lequel elles se trouvent et la rareté des développeurs et des personnes ayant un profil technique font que la convergence est encore une tâche en suspens.

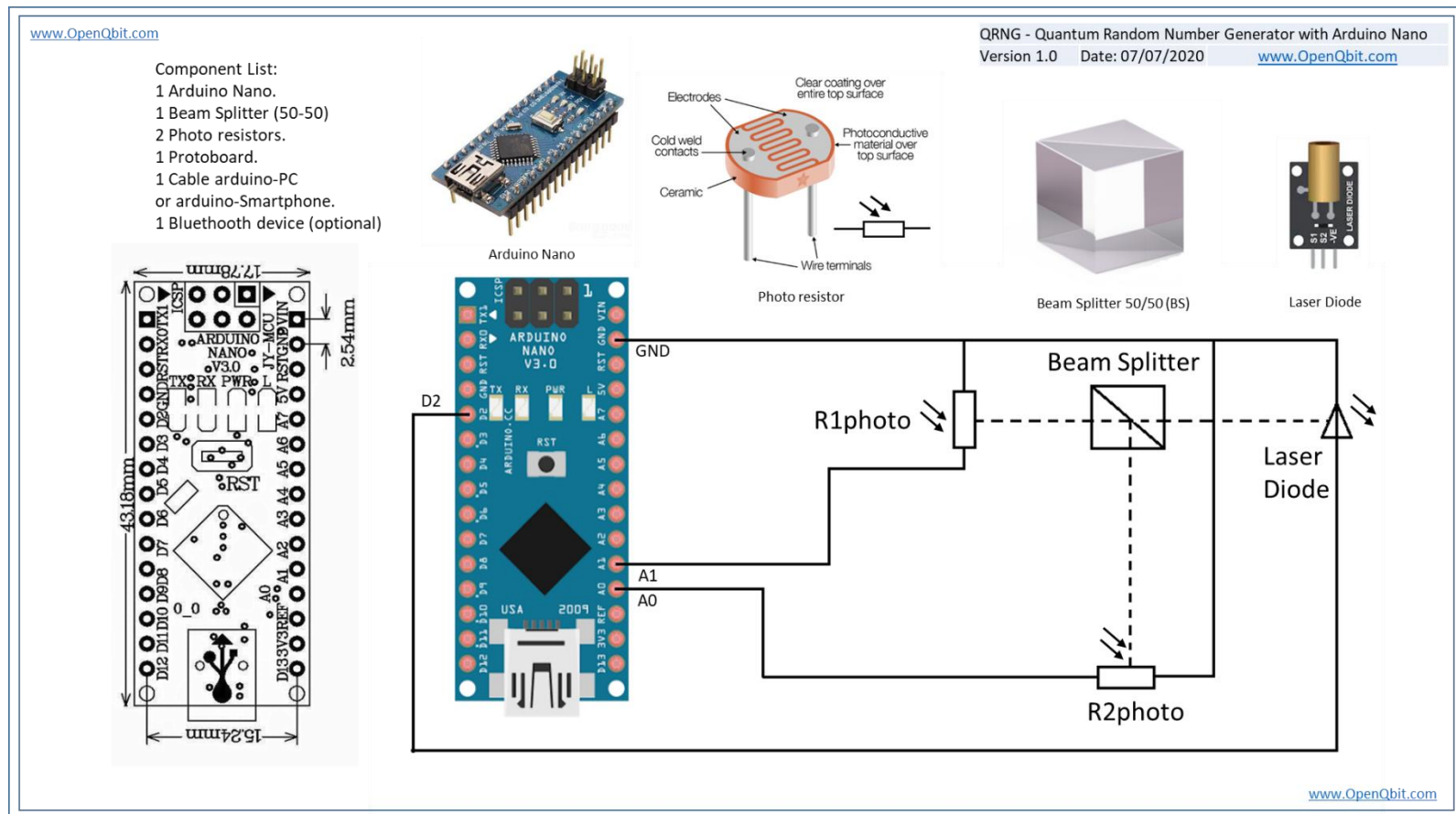
En raison de leur potentiel perturbateur, les technologies quantiques devraient non seulement converger avec toutes ces nouvelles technologies, mais aussi avoir une influence transversale sur pratiquement toutes d'entre elles. L'informatique quantique menacera l'authentification, l'échange et le stockage sécurisé des données, ce qui aura un impact majeur sur les technologies où la cryptographie joue un rôle plus pertinent, comme la cybersécurité ou la chaîne de blocage, et un impact négatif mineur, mais qui doit également être pris en compte dans des technologies telles que la 5G, l'IdO ou les drones.

**Vous voulez pratiquer l'informatique quantique ?**

Des dizaines de simulateurs d'ordinateurs quantiques sont déjà disponibles sur le net avec différents langages de programmation déjà utilisés tels que C, C++, Java, Matlab, Maxima, Python ou Octave. De plus, de nouveaux langages comme Q#, lancé par Microsoft. Vous pouvez explorer et jouer avec une machine quantique virtuelle grâce à des plateformes telles qu'IBM et Rigetti.

### 3. Création du dispositif "Hardware" d'un QRNG (Quantum Random Number Generator).

Nous allons maintenant créer un dispositif "matériel" physique pour générer des nombres aléatoires quantiques (QRNG) avec des composants peu coûteux qui peuvent être facilement assemblés à la maison et coûtent environ 35 dollars américains.



## 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);
}
```

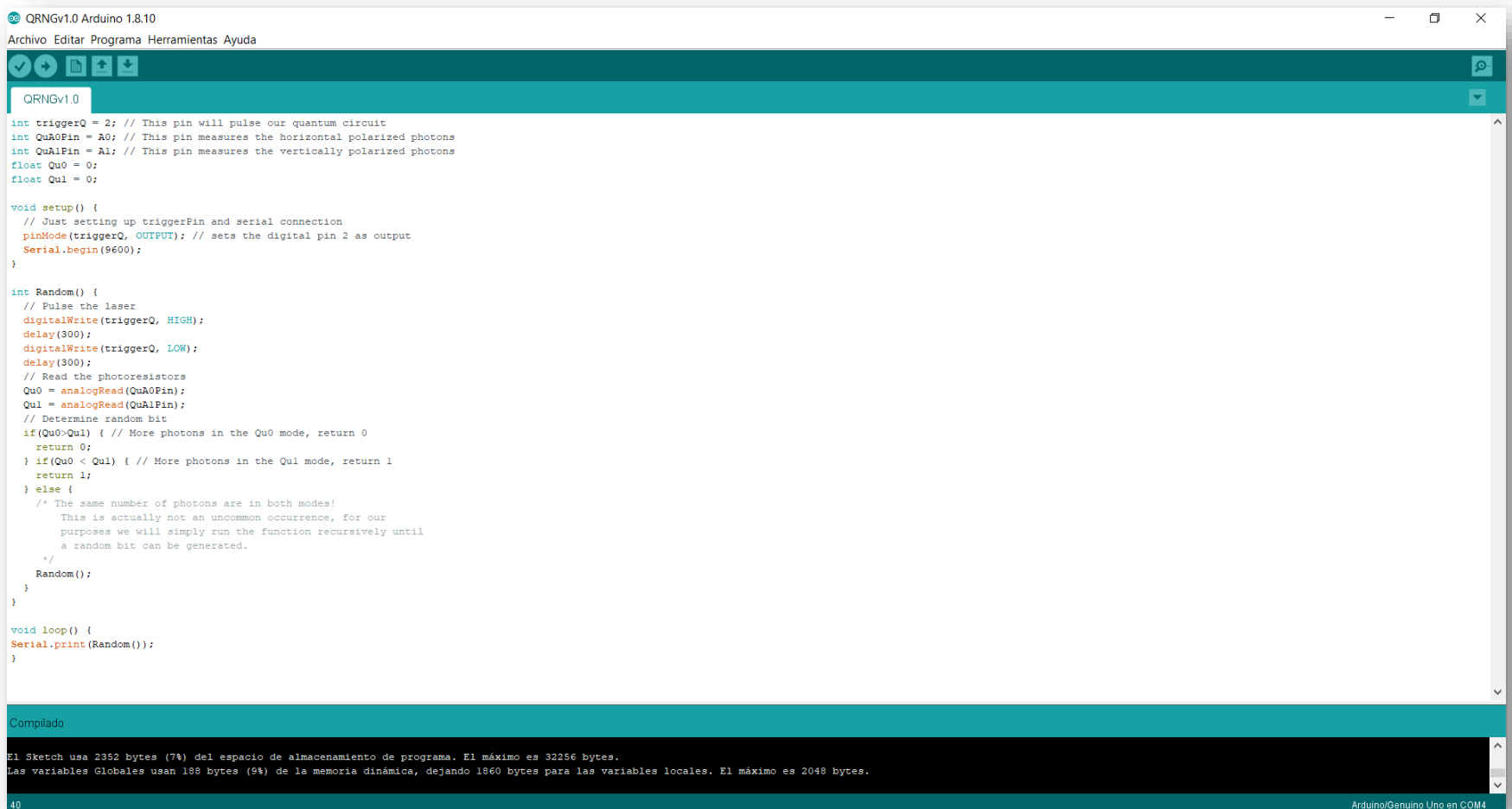
```
int Random() {
  // Pulse the laser
  digitalWrite(triggerQ, HIGH);
  delay(300);
  digitalWrite(triggerQ, LOW);
  delay(300);
  // Read the photoresistors
  Qu0 = analogRead(QuA0Pin);
  Qu1 = analogRead(QuA1Pin);
  // Determine random bit
  if(Qu0>Qu1) { // More photons in the Qu0 mode, return 0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

Compilation du programme QRNGv10.ino et téléchargement sur arduino nano....



```
QRNGv1.0 Arduino 1.8.10
Archivo Editor Programa Herramientas Ayuda

QRNGv1.0

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());
}
```

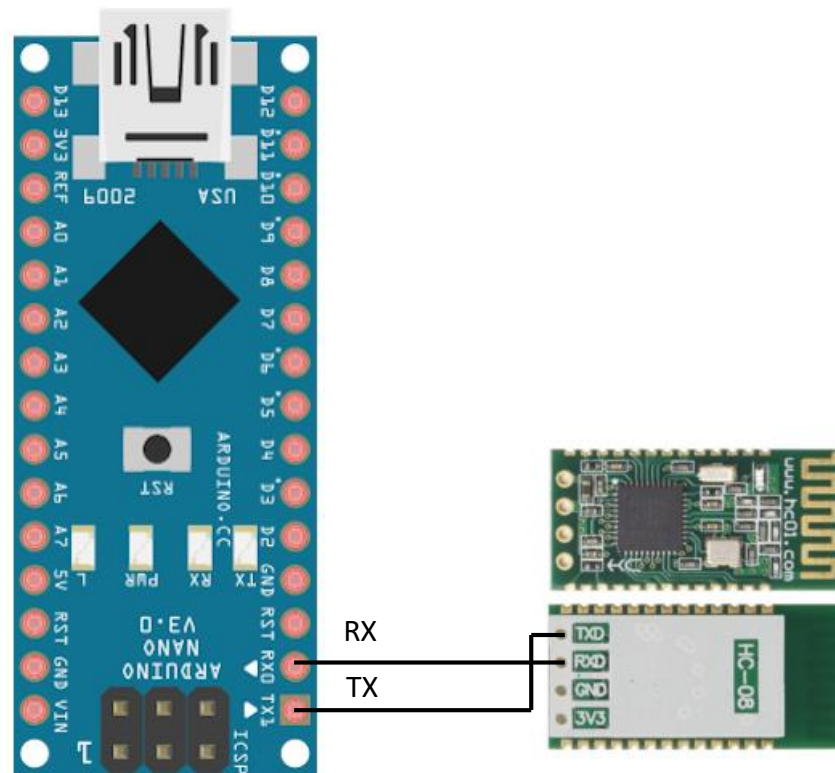
Compilado

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.

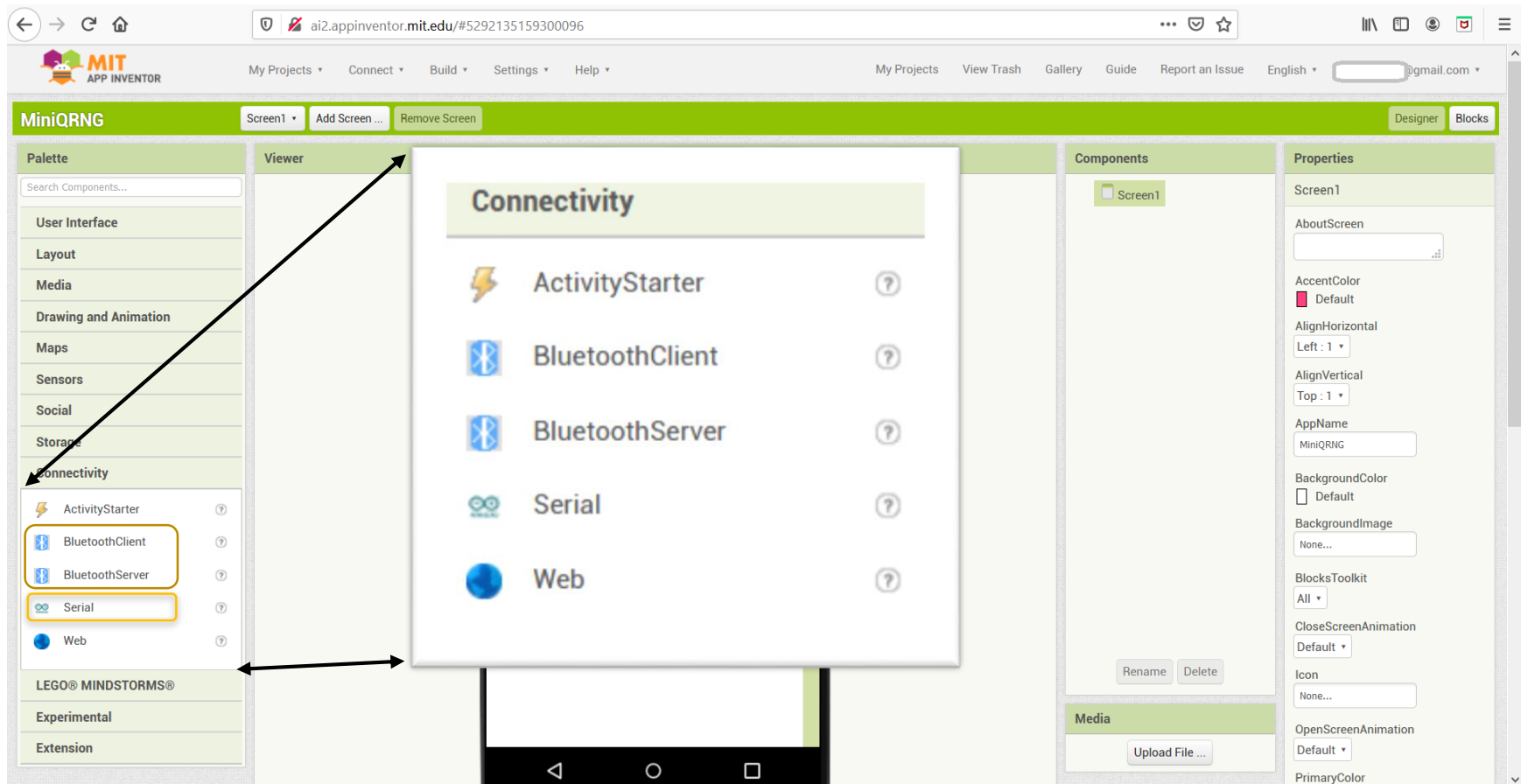
40 Arduino/Genuino Uno en COM4

Il y a deux façons de communiquer avec le nano difficile, l'une via le port série et l'autre via une connexion Bluetooth.

Pour la connexion bluetooth est très simple, il suffit d'acheter le module HC-08 ou un module similaire et de le connecter comme suit :



Les composants série ou Bluetooth suivants peuvent être utilisés pour connecter App Inventor à Arduino :





Maintenant compilé et chargé le programme QRNGv10.ino ne manque que de communiquer avec la nano ardue pour sauvegarder les données (nombres aléatoires quantiques) celles-ci seront en format binaire, cependant, les données obtenues peuvent être facilement passées à un autre format tel que l'hexadécimal ou le décimal en fonction de l'exigence finale.

Enfin, pour voir un exemple du fonctionnement de la connexion série ou Bluetooth, voici quelques liens de référence.

Rappelez-vous que tout est par la programmation Blockly à être testé avec App Inventor qui a déjà des blocs pour la communication avec arduino série ou d'autres systèmes de type blockly peut être par le biais de Bluetooth similaire en ligne.

<http://kio4.com/appinventor/9A0BluetoothRXTX.htm>

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

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

Examiner l'ensemble du projet de conception et d'utilisation des extensions QRNG (Quantum Random Number Generator). Consultez le manuel d'utilisation à l'adresse suivante

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





#### 4. Qu'est-ce que la preuve de quantum (PQu) ?

PoQu. - "Proof of Quantum" est un algorithme de consensus développé pour la Mini BlocklyChain et le COINsolidation, ce test est une variante du Proof of Work (PoW) qui fonctionne comme suit.

Le Test of Quantum (PoQu) au démarrage est exécuté avec le même algorithme que le "Test of Work" (PoW) qui consiste à mettre le processeur de l'appareil (PC, serveur, tablette ou téléphone portable) au travail pour obtenir une chaîne de caractères qui est un puzzle mathématique appelé "hash".

N'oubliez pas qu'un "hachage" est un algorithme ou un processus mathématique qui, lors de l'introduction d'une phrase ou d'un type d'information numérique tel qu'un fichier texte, un programme, une image, une vidéo, un son ou tout autre type d'information numérique, nous donne comme résultat un caractère alphanumérique qui représente la signature numérique qui le représente de manière unique et non répétitive des données, l'algorithme de hachage est unidirectionnel, ce qui signifie que lorsque vous saisissez une donnée pour obtenir sa signature "hachage", son processus inverse ne peut pas être effectué, ayant une signature "hachage" nous ne pouvons pas savoir quelles informations ont été obtenues ; cette propriété nous donne un avantage de sécurité pour traiter les informations que nous envoyons sur Internet. Comment cela fonctionne-t-il ? Imaginez que vous envoyez n'importe quel type d'information par des canaux non sécurisés et que vous l'accompagnez de son "hachage source" respectif, le récepteur, lorsqu'il reçoit l'information, peut obtenir le "hachage" de l'information reçue ; nous l'appellerons "hachage destination" et nous le vérifierons avec le "hachage source" ; si les deux "hachages" sont identiques, nous pouvons confirmer que l'information n'a pas été altérée dans le canal qui a été envoyé, c'est juste un exemple où ce type de processus de sécurité de l'information est actuellement utilisé.

Actuellement, il existe différents types d'algorithmes ou de processus de hachage qui diffèrent par leur niveau de sécurité. Les plus utilisés ou les plus connus sont : MD5, SHA256 et SHA512.

Exemple de SHA256 :

Nous avons une chaîne ou une phrase comme suit : "Mini BlocklyChain" est modulaire.

Si nous appliquons un hachage de type SHA256 à la chaîne précédente, cela nous donnera le hachage suivant.



**f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db8**

La chaîne alphanumérique ci-dessus est la signature qui représente la phrase dans l'exemple ci-dessus

Par exemple, nous pouvons utiliser le site sur Internet :

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

Dans le cas de l'algorithme "Test Work" (PoW), il fonctionne en utilisant la puissance de calcul pour obtenir un hachage prédéfini.

Imaginons que nous ayons le précédent "hash" que nous avons pris de la chaîne "Mini BlocklyChain is modular".

**f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db8**

A ce "hachage" à son début on met le paramètre de difficulté qui est simplement de mettre des zéros "0" au début, c'est-à-dire que si on dit que la difficulté est de 4 il aura "0000" + "hachage" à cela on l'appellera "hachage de semence".

**0000 f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db8**

Maintenant, en tenant compte du fait que nous connaissons les informations d'entrée qui constituent la chaîne : "La Mini BlocklyChain est modulaire", nous ajoutons à la fin de la chaîne un nombre commençant à zéro "0" et nous en retirons son hachage, nous l'appellerons "hachage nonce" :

**f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db80**

On a du haschisch nonce :

**7529f3ad273fc8a9eff12183f8d6f886821900750bb6b59c1504924dfd85a7c8**

Ensuite, nous effectuons une comparaison du nouveau "hash nonce" avec le "hash seed" ; s'ils sont égaux, le nœud qui trouve le premier l'égalité gagnera l'exécution du traitement de la transaction en cours. Comme nous pouvons le voir, ce processus est basé sur la probabilité et la puissance de calcul du dispositif, ce qui donne au test de "Preuve du travail" une équité consensuelle pour tous les nœuds.

Si le "hachage de semence" ne coïncide pas avec le "hachage nonce", la difficulté est augmentée de un et le "hachage nonce" est à nouveau supprimé, le nombre qui est augmenté est appelé le nombre "nonce", il est comparé avec le "hachage de semence" jusqu'à ce qu'ils coïncident ou soient identiques.



Comme on peut le voir, le nombre "nonce" ou augmentation est celui qui permettra d'obtenir le "hash" de l'égalité.

Basé sur l'algorithme du "Test of Work" (PoW), l'algorithme du test quantique (PoQu) est basé sur l'obtention du nombre "nonce" comme le fait le PoW et l'utilisation d'un niveau de difficulté minimum allant de 1 à 5, cela ne sert qu'à obtenir le droit pour l'appareil mobile d'être candidat à l'obtention d'un consensus.

Le test quantique (PoQu), est activé lorsque le téléphone mobile a terminé le PoW minimum et gagne le passage pour obtenir un nombre de probabilité dans le système QRNG.

Le QRNG (Quantum Random Number Generator) est un générateur de nombres aléatoires quantiques, ce système est basé sur la génération de vrais nombres aléatoires basés sur la mécanique quantique est le système le plus sûr aujourd'hui pour générer de tels nombres. Pour plus de détails, voir "Sécurité du calcul quantique" dans l'index 3.

COINsolidation peut mettre en œuvre les deux types de concession minimum PoW et PoQu.

Le test PoQu est basé sur l'obtention du numéro "nonce". Ce numéro dans le test PoQu est connu sous le nom de "Magic Number" et avec cela le système Peer to Peer confirmera si le numéro est correct et ensuite un numéro aléatoire sera obtenu avec le pool de serveurs QRNG de COINsolidation. Ce nombre aléatoire sera enregistré dans tous les nœuds, une liste sera créée contenant **((Node Addition) / 2) + 1** et de cette liste sera choisi celui qui a le plus grand pourcentage de probabilité d'être le candidat gagnant du consensus (PoQu) et il exécutera la file d'attente des transactions en cours.

L'algorithme PoQu utilise également les tests du **NIST** (National Institute of Standards and Technology) pour nous assurer que les nombres aléatoires dans le QRNG sont vraiment des nombres aléatoires.

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

Dans COINsolidation, nous avons mis en place un bloc pour PoW et un bloc pour PoQu. Ces blocs utilisent un type de hachage : SHA256 pour une utilisation gratuite, pour une utilisation commerciale nous avons un SHA512 et d'autres hachages selon les besoins.

Pour plus de détails sur le concept de HASH, voir :

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

REMARQUE : le Test of Work (PoW) utilisé dans les téléphones mobiles ne peut utiliser qu'une difficulté maximale de 5 car le traitement mathématique de ces appareils n'est pas dédié comme les serveurs ou les PC. Nous utilisons uniquement l'algorithme PoW pour obtenir la possibilité d'obtenir votre laissez-passer ou la permission d'entrer dans le système



du générateur quantique de nombres aléatoires (QRNG) et avec lui pour exécuter l'algorithme du générateur quantique de nombres aléatoires (PoQu). Voir l'utilisation de (PoQu) dans Mini BlockyChain :

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

## 5. Algorithme pour la création d'une adresse universelle consolidée (AUC)

Nos adresses CUA (Consolidated Universal Address) sont créées à l'aide de l'algorithme suivant :

Étape 1 - Les identificateurs sont retirés des adresses respectives, ce sont les caractères alphanumériques qui identifient l'adresse à partir de laquelle les chaînes de blocs ont été créées.

Adresse Bitcoin- Token - (OAP).

**akXma4vqxvmEqnVAKSM953wYsnjNBhN3GM7**

Adresse Ethereum - Token COINsolidation - (ERC20).

**0x9d08c0ac0f2fdf078c883db6fa617b15776e4b41**

Étape 2 - Le SHA512(address String-Text) de chaque adresse sans son identifiant initial est obtenu en enlevant le "a" de A1 et le "0x" de A2 et en prenant les deux derniers caractères de chaque opération de hachage symbolisée par un "U". Numéros de vérification.

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

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

Étape 3 - Les caractères de chaque adresse sont concaténés un par un à partir de l'adresse qui comporte le moins de caractères qui la composent ; si le nombre de caractères est identique, la concaténation peut commencer à partir de n'importe quelle adresse.

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].

Concaténation des adresses :

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

**\*\*Les derniers caractères qui ne peuvent pas être concaténés sont placés au début de la chaîne.**



6e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

Étape 4 - Le nombre de caractères qui pourraient être concaténés à l'étape 3 est ajouté au début de la chaîne résultant de l'étape 3.

66e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

Étape 5 - Les deux paires de vérificateurs de l'étape 2 de chaque direction sont concaténées au début de la chaîne résultant de l'étape 3 dans le même ordre A1 + A2.

50fb66e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

Étape 6 - L'identification **CUAG** (Consolidated Universal Address Genesis) est intégrée au début de l'adresse créée à l'étape 5.

**cua**50fb66e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

\*\*Dans le cas de la consolidation des adresses de Bitcoin et Ethereum, elle donnera une adresse composée de **82** caractères hexadécimaux.

## 6. Algorithme pour la double adresse consolidée (DAC) et (HAC).

La création d'un DAC est la même que celle du CUA, la différence est que dans les DAC, ils sont utilisés pour consolider les adresses normales pour recevoir des transactions, ces adresses ne représentent pas de cryptomonedra ou de jeton.

Étape 1 - Les identificateurs sont retirés des adresses respectives, ce sont les caractères alphanumériques qui identifient l'adresse à partir de laquelle les chaînes de blocs ont été créées.

**18gYNA9c2G9X8HZ8QxWLpLXZauAxFnsJbe** (adresse Bitcoin)

**0x5d2Acdb34c279Aa6d1e94a77F7b18aB938BFb2bB** (Dirección Ethereum)

Étape 2 - Le SHA512(address String-Text) de chaque adresse sans son identifiant initial est obtenu en enlevant le **"1"** de A1 et le **"0x"** de A2 et en prenant les deux derniers caractères de chaque opération de hachage symbolisée par un "U". Numéros de vérification.

U(SHA512(**8gYNA9c2G9X8HZ8QxWLpLXZauAxFnsJbe**)) = **48**

U(SHA512(**5d2Acdb34c279Aa6d1e94a77F7b18aB938BFb2bB**)) = **f3**



Étape 3 - Les caractères de chaque adresse sont concaténés un par un à partir de l'adresse qui comporte le moins de caractères qui la composent ; si le nombre de caractères est identique, la concaténation peut commencer à partir de n'importe quelle adresse.

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].

Concaténation des adresses :

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

**\*\*Les derniers caractères qui ne peuvent pas être concaténés sont placés au début de la chaîne.**

**8BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3**

Étape 4 - Le nombre de caractères qui pourraient être concaténés à l'étape 3 est ajouté au début de la chaîne résultant de l'étape 3.

**78BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3**

Étape 5 - Les deux paires de vérificateurs de l'étape 2 de chaque direction sont concaténées au début de la chaîne résultant de l'étape 3 dans le même ordre A1 + A2.

**48f378BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3**

Étape 6 - L'identification **DAC** (Dual Address Consolidated) est intégrée au début de l'adresse créée à l'étape 5.

**dac48f378BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3**

**\*\*Dans le cas de la consolidation des adresses de Bitcoin et Ethereum, elle donnera une adresse composée de **81** caractères hexadécimaux.**

Dans le cas du HAC (Hibric Address Consolidated) est appliqué dans le précédent ce qui varie sont les adresses qui sont utilisées, dans ce cas nous utiliserons une adresse qui représente un actif (Cryptomonedra ou token) et une adresse standard normale de transferts d'actifs de quelque type de blockchain.

**NOTE** : La taille des adresses CUA, HAC et DAC peut varier dans chaque cas en fonction des adresses qui les composent



## Projet et solution par COINsolidation.

Actuellement, il existe différents types de Blockchain orientés vers des actifs de caractéristiques différentes, ce qui conduit à avoir un nombre infini de types d'adresses d'usage quotidien doivent garder un contrôle serré pour éviter de faire des erreurs de transfert.

D'autre part, le monde de la cryptographie et des jetons est limité aux experts financiers ou, dans leur cas, aux experts en technologie des chaînes de blocs, de sorte qu'il est difficile pour le citoyen moyen de s'aventurer dans la création de sa propre cryptographie ou de son propre jeton.

Nous avons résolu les deux problèmes précédents dans COINsolidation en faisant les points et/ou outils suivants que nous avons créés.

Pour le point de contrôle des adresses de différentes chaînes de blocs, nous avons créé un algorithme où il consolide (joint) deux ou plusieurs adresses dans leurs différentes combinaisons donnant comme résultat une seule adresse de type CUA, HAC et/ou DAC.

Avec cette solution, au lieu d'envoyer deux adresses de la même chaîne de blocs ou de chaînes différentes, une seule adresse consolidée sera utilisée.

Pour le second problème, nous avons utilisé la méthodologie de programmation appelée Blockly, un outil visuel qui ne nécessite pas de grandes connaissances en programmation et qui permet à toute personne ou entreprise moyenne de créer ses propres applications sans avoir à investir des équipes de développement coûteuses, du temps et de l'argent.

Nous avons créé les extensions (modules) pour qu'il suffise de les installer et de les utiliser pour créer des applications mobiles, en 15 minutes. Vous pouvez par exemple échanger votre propre crypto-monnaie ou développer votre propre monnaie (token) en quelques minutes. Tout cela en utilisant une sécurité des données de pointe appelée PQC (Cryptographie post-quantique).

Il suffit d'installer les extensions sur n'importe quel outil gratuit tel que Appinventor, AppyBuidr, Thunkable, Kondular ou autres et en quelques minutes vous pouvez entrer dans le monde des cryptomonies et de la création de jetons, le tout dans la paume de votre main.

Enfin, la consolidation du COIN crée l'utilisation d'une sécurité quantique à faible coût (logiciel et matériel) qui peut être utilisée pour protéger les données informatiques à domicile. Actuellement, les technologies basées sur l'informatique quantique et la sécurité ont un coût élevé que seules les entreprises ayant un niveau financier élevé peuvent créer et utiliser. Cependant, dans le cadre de COINsolidation, nous pensons que les nouvelles

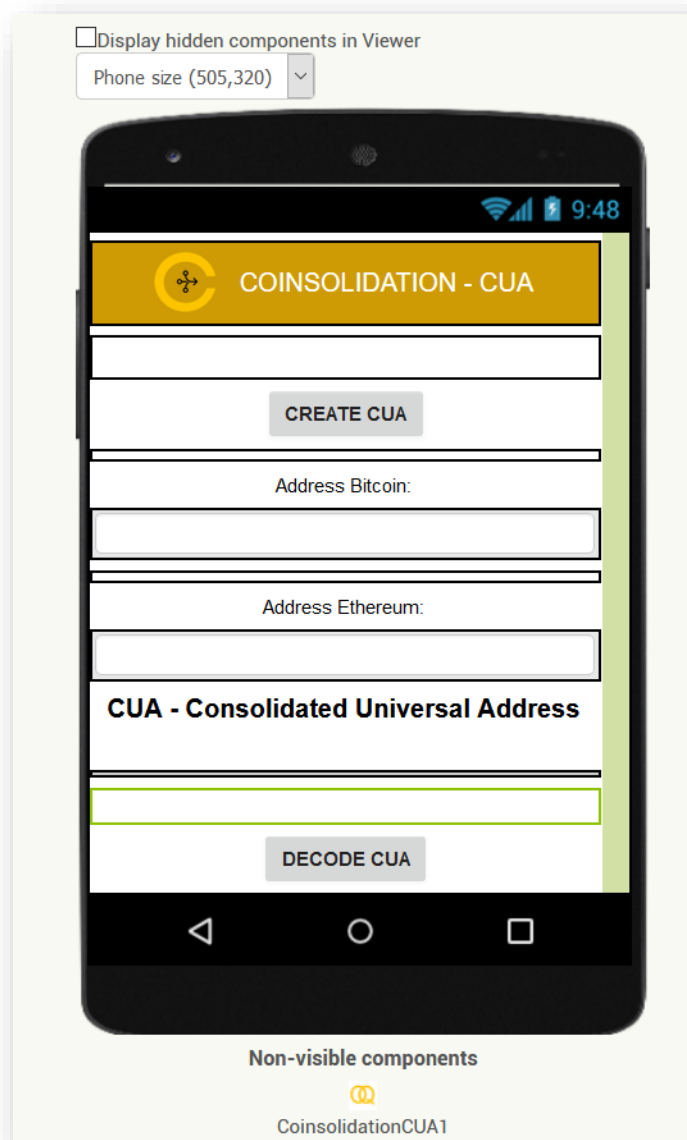


technologies devraient être accessibles à tous, que l'équité d'utilisation de la chaîne de blocage et de l'informatique quantique devrait être pour tous, nous créons des logiciels libres (cryptomonique) et du matériel à faible coût (sécurité quantique).

## 7. Création de l'App CUA (Consolidated Universal Address) en 15 minutes.

\*Application pour les pièces Bitcoin et Ethereum (BTC-ETH)

Conception d'un écran 5 minutes dans <https://appinventor.mit.edu/>







Utilisation de l'extension **CoinsolidatioCUA.AIX** (5 minutes).

The image shows a Scratch script with 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**



Nous créons l'application dans **Menu > Build > App** (fournir le code QR pour .apk) - (5 minutes).

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**

0

0

Show Warnings

CUA Progress Bar

35%

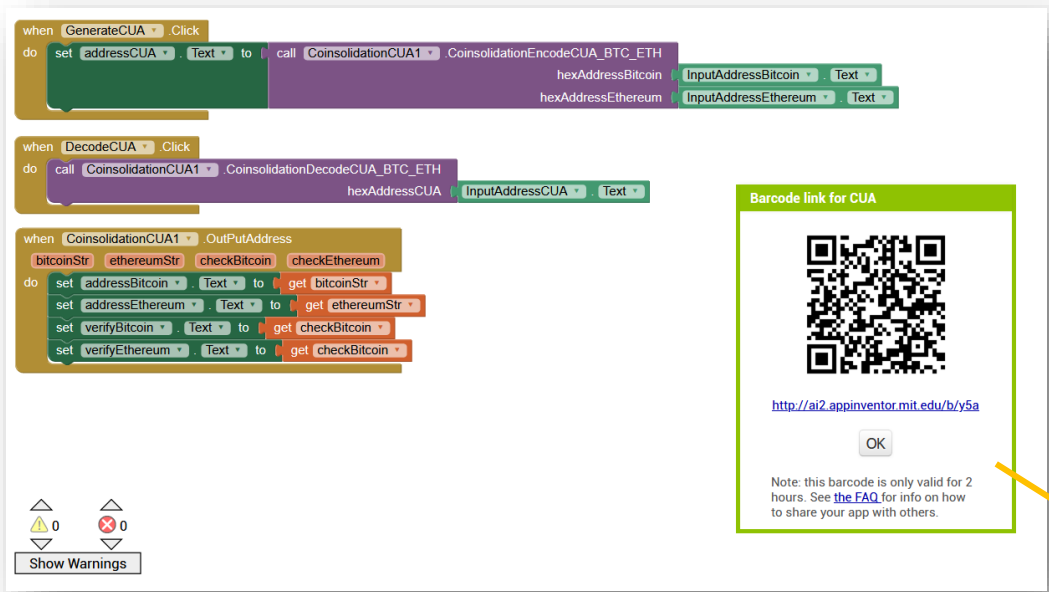
Compiling part 2 (please wait)

+

-

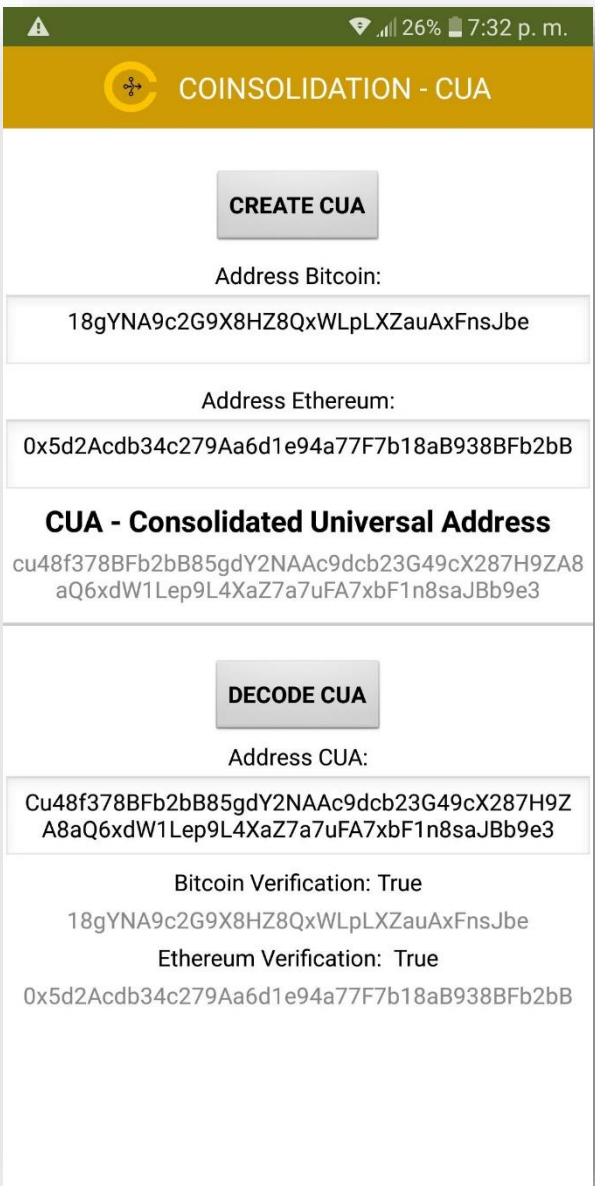


Nous avons installé l'application sur le téléphone portable à partir du QR en utilisant l'application Android d'AppInventor (MIT AI2 Companion) - <https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3>



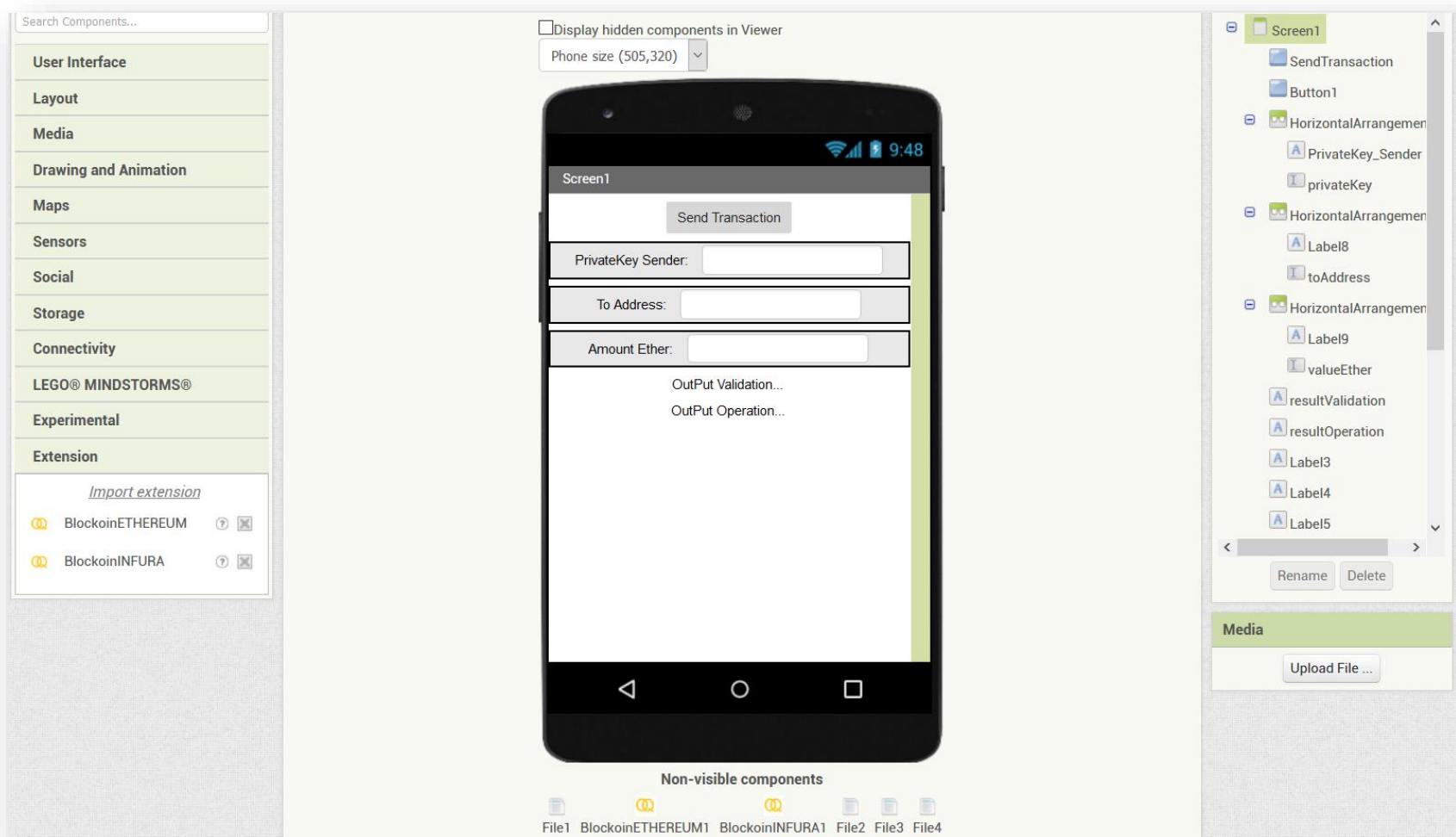
**NOTE :** Le fichier APK de l'application prêt à être installé se trouve dans le répertoire suivant : <https://github.com/COINsolidation/App>

Pour revoir le code Java pour la génération de l'extension CUA et pour mettre en œuvre un algorithme consolidé de génération d'adresses universelles, consultez l'annexe "Code pour l'algorithme CUA" ou le lien du code : <https://github.com/COINsolidation/source>



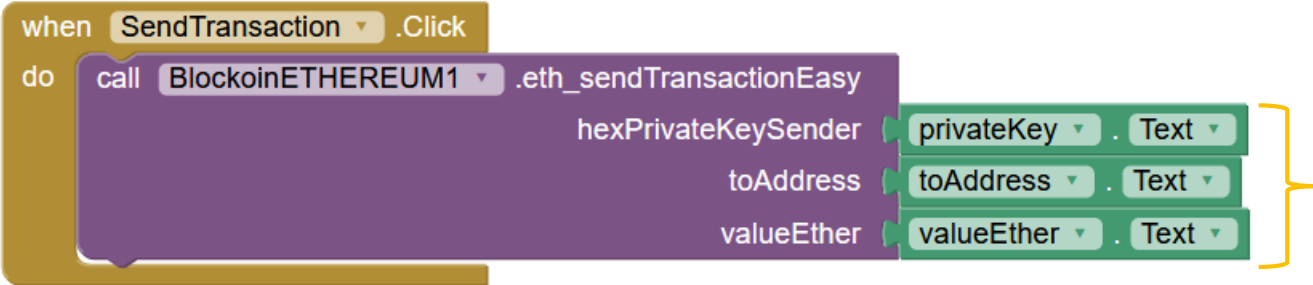


**8. Créez votre crypto-bourse Ethereum sur Android en seulement 15 minutes.**  
Design in App Inventor (écran). - 5 minutes.





Blocs de fonctions (eth\_SendTransactionEasy) et événements (OutPutSendTransactionEasy) - 5 minutes

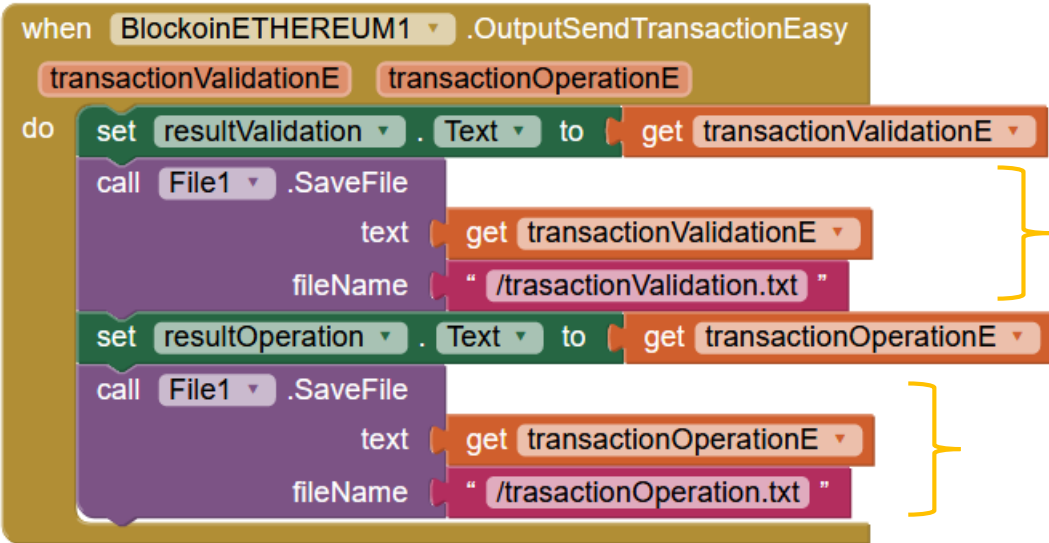


Données d'entrée :

**PrivateKey** : Clé primaire de l'adresse de l'expéditeur.

**toAddress** : Adresse hexadécimale du destinataire.

**valeurEther** : Indiquez la quantité d'éther qui sera envoyée.



Enregistrez les résultats dans des fichiers texte :

Fichier de fonction1 : Fichier **trasactionValidation.txt**

Enregistrez les résultats dans des fichiers texte :

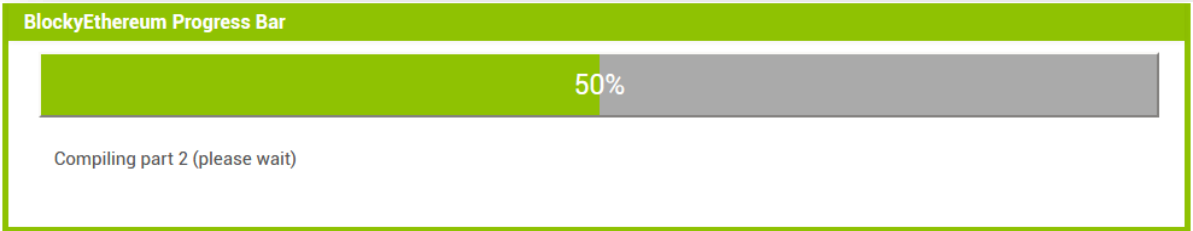
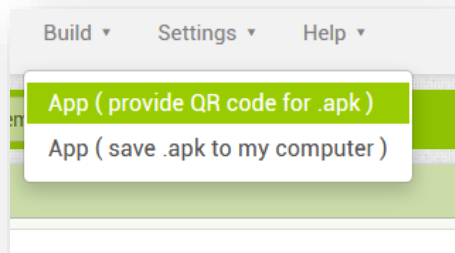
Fonction File2 : Fichier **trasactionValidation.txt**

\*\*Pour plus de détails, voir le guide d'utilisation de l'extension Ethereum Exchange (EEE) dans le référentiel : <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>



Nous compilons, générons le fichier APK pour l'installer sur l'appareil Android. - 5 minutes



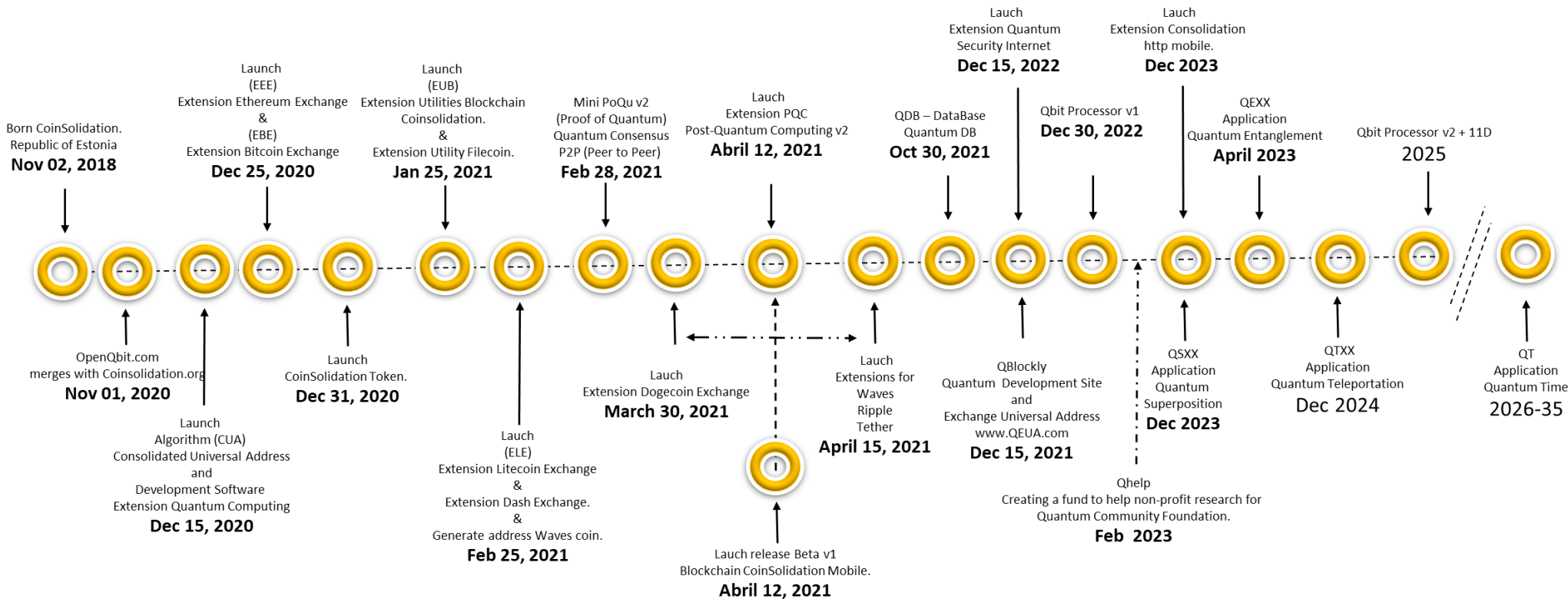
REMARQUE : Lorsque la transaction est exécutée, il faut environ 6 à 8 secondes pour relâcher le bouton "Envoyer la transaction". En raison du temps de connexion avec le réseau Ethereum.

Pour plus de détails sur l'extension de l'EEE - (extension de l'échange d'éthereum). Voir le manuel d'utilisation de l'EEE dans le lien :  
<https://github.com/COINsolidation/UserGuide>



9. Feuille de route COINsolidation.

ROADMAP



\*OpenQbit.com fusionne avec COINsolidation.org (Nov 01, 2020) / OpenQbit est spécialisé dans l'informatique quantique et la sécurité quantique.  
\*La version 1 du processeur quantique utilisera des portes de logique quantique de base pour une utilisation domestique.



EXchange  
tensions

10.COINsolidation Token (CUAG) - PLAN DE DISTRIBUTION DES ICOS.

L'OIC se divise en trois étapes :

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	ADRESSE DE L'OIC
0xa646c054394f85257E18D56Cf5c6b5E603447470	ADRESSE DE L'OPÉRATION DE CO-SOLIDATION





## 11. Caractéristiques générales du jeton de consolidation COIN :

Créé par : Lugu Samaya.

Nom : COINsolidation

Symbole : CUAG - (Consolidated Universal Address Genesis).

Type : NFT

Total des jetons créés : 40.000.000.000,00

Nombre de décimales : 18

Pays de lancement : Estonie

Site officiel : [www.COINsolidation.org](http://www.COINsolidation.org)

Société : COINsolidation International.

Date de lancement : 30 décembre 2020

Algorithme de consensus : PQu (Proof of Quantum)

Algorithme d'adresse : Adresse universelle consolidée (AUC).

Sécurité utilisée : PQC (Post-Quantum Cryptography) basée sur l'informatique quantique.

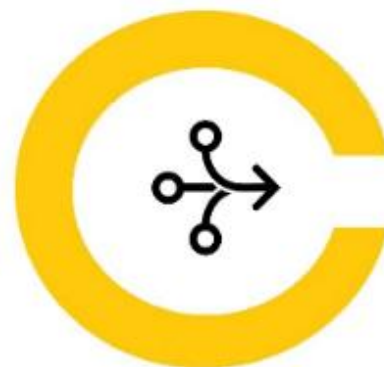
Proposition technologique : extension des systèmes Blockly à l'utilisation de cryptomonades et mise en œuvre de la sécurité quantique.

Partenariats ou accords technologiques (fusion) :

Entreprise : OpenQbit Inc.

Industrie : Informatique quantique et PQC (Cryptographie post-quantique).

Site officiel : [www.OpenQbit.com](http://www.OpenQbit.com)





## 12. Concepts de base appliqués dans les plateformes Blockchain.

### Qu'est-ce qu'une chaîne de blocage ?

La chaîne de blocs est généralement associée aux bitcoins et autres devises cryptées, mais ce ne sont que la partie visible de l'iceberg, car elle n'est pas seulement utilisée pour la monnaie numérique, mais peut être utilisée pour toute information pouvant avoir une valeur pour les utilisateurs et/ou les entreprises. Cette technologie, dont les origines remontent à 1991, lorsque Stuart Haber et W. Scott Stornetta ont décrit les premiers travaux sur une chaîne de blocs cryptographiquement sécurisés, n'a pas été remarquée avant 2008, date à laquelle elle est devenue populaire avec l'arrivée des bitcoins. Mais actuellement, son utilisation est demandée dans d'autres applications commerciales et devrait se développer à moyen terme sur plusieurs marchés, comme les institutions financières ou l'Internet des objets, entre autres secteurs.

La blockchain, mieux connue sous le terme de blockchain, est un enregistrement unique, convenu d'un commun accord, réparti sur plusieurs nœuds (appareils électroniques tels que les PC, les smartphones, les tablettes, etc. Dans le cas des crypto-monnaies, on peut considérer qu'il s'agit du livre comptable où chacune des transactions est enregistrée.

Son fonctionnement peut être complexe à comprendre si nous entrons dans les détails internes de sa mise en œuvre, mais l'idée de base est simple à suivre.

Il est stocké dans chaque bloc :

- 1.- un certain nombre d'enregistrements ou de transactions valables,
- 2.- des informations concernant ce bloc,
- 3.- son lien avec le bloc précédent et le bloc suivant grâce au hachage de chaque bloc –un code unique qui serait comme l'empreinte digitale du bloc.

Par conséquent, **chaque bloc** a une **place spécifique et immuable dans la chaîne**, car chaque bloc contient des informations provenant du hachage du bloc précédent. La chaîne entière est stockée sur chaque nœud du réseau qui constitue la chaîne de blocs, de sorte qu'**une copie exacte de la chaîne est stockée sur tous les participants du réseau**.

### Qu'est-ce qu'une adresse ou un compte au sein de la plateforme Ethereum ?

Il s'agit d'une chaîne de 42 caractères dans la plateforme Ethereum représentant un nombre en base hexadécimale, où les avoirs définis dans l'Ethereum seront déposés ou envoyés.



Dans d'autres plateformes de chaînes de blocs, le nombre de caractères du compte ou de l'adresse peut être différent, par exemple :

**0x5d2Acdb34c279Aa6d1e94a77F7b18aB938BFb2bB**

### **Qu'est-ce qu'une kryptomonie ?**

Il s'agit d'une monnaie numérique ou virtuelle conçue pour fonctionner comme un moyen d'échange. Elle utilise la cryptographie (sécurité numérique) pour sécuriser et vérifier les transactions, ainsi que pour contrôler la création de nouvelles unités d'une cryptomoney particulière.

### **Qu'est-ce qu'un jeton ?**

Les jetons sont des actifs numériques qui peuvent être utilisés dans le cadre d'un écosystème de projet donné.

La principale distinction entre les jetons et les crypto-monnaies est que les premiers nécessitent une autre plate-forme de chaîne de blocs (pas la leur) pour fonctionner. Ethereum est la plateforme la plus courante pour la création de jetons, principalement en raison de sa fonction de contrat intelligent. Les jetons créés sur la chaîne de blocs Ethereum sont généralement connus sous le nom de jetons ERC-20, bien qu'il existe d'autres types de jetons plus spécialisés comme le jeton ERC-721 utilisé principalement pour les biens de collection (cartes, utilisation dans les jeux vidéo, œuvres d'art, etc.).

### **Qu'est-ce qu'un échange ?**

Un échange de crypto-monnaie est le point de rencontre où les échanges de crypto-monnaie ont lieu en échange de monnaie fiduciaire ou d'autres crypto-monnaie. Dans ces maisons d'échange en ligne, le prix du marché est généré, ce qui marque la valeur des cryptomonies en fonction de l'offre et de la demande.

### **Qu'est-ce que le taux de change ?**

Ce sont les taux de la valeur d'un éther ou d'une autre monnaie cryptée dans la monnaie en circulation dans chaque pays. Par exemple, le jour de la création de ce manuel, un éther a une valeur en dollars américains de 430,94

### **Qu'est-ce qu'une transaction ?**

Il s'agit de l'exécution ou du transfert d'un certain type de bien non corporel auquel on peut attribuer une valeur préétablie dans le cadre du système Ethereum et qui peut ensuite être transformé en une valeur corporelle pour une entreprise ou une personne.

### **Qu'est-ce que txHash ?**



Il s'agit d'un nombre hexadécimal qui permet de suivre le résultat en détail de chaque transaction.

### Quels sont les types de transactions ?

Vous en avez deux types, l'un est la transaction "hors ligne" que cela crée sans qu'il soit nécessaire d'avoir une connexion au réseau principal d'Ethereum ; elle peut être stockée jusqu'à ce que vous choisissiez de vous connecter au réseau d'Ethereum et de libérer la transaction, ce qui présente l'avantage de la sécurité car toute la transaction est traitée hors ligne ce qui évite toute anomalie qui pourrait se trouver dans la connexion au réseau. L'autre transaction est celle "en ligne" qui doit toujours être connectée à l'internet avec les avantages et les inconvénients qu'elle comporte en matière de sécurité.

### Qu'est-ce qu'une adresse Blockchain ?

Une adresse ou un compte est composé de trois parties, l'adresse, la clé publique et la clé privée, ces deux clés sont une chaîne de chiffres et de caractères au format hexadécimal qui sont utilisés pour envoyer et recevoir (actif) ou éther (monnaie numérique).

La clé primaire ne doit jamais être partagée avec quiconque car c'est elle qui autorise la libération du solde (signe les transactions) détenu sur le compte.

La clé publique est connue de tout le monde et est partagée avec tout le monde car elle est la référence pour confirmer que la transaction est correcte à la fois en termes de valeur et de destinataire.

Exemples de composantes de la gestion du réseau Ethereum :

```
{  
  "private" :  
    "429a043ea6393b358d3542ff2aab9338b9c0ed928e35ec0aed630b93adb14a1c",  
  "public" :  
    "049b4b7e72701a09d3ee09165bba460f2549494a9d9fd7a95aaac57c2827eac162fd9e105b2461cd6594ca8ca6a8daf10fe982f918be1b0060c87db9cfbcd289a8",  
  "address" : "88ab6dcecc3603c7042f4334fc06db8e8d7062d5"  
}
```



### 13. Qu'est-ce que la programmation en blockly ?

**Blockly** est une **méthodologie de programmation visuelle** composée d'un ensemble simple de commandes que nous pouvons combiner comme s'il s'agissait des pièces d'un puzzle. C'est un outil très utile pour ceux qui veulent **apprendre à programmer** de manière intuitive et simple ou pour ceux qui savent déjà programmer et qui veulent voir le potentiel de ce type de programmation. Il est basé sur le langage JavaScript et a été développé par la société Google et le MIT.

Blockly est une forme de programmation où il n'est pas nécessaire d'avoir des connaissances dans un quelconque langage informatique, c'est parce qu'il s'agit simplement de joindre des blocs graphiques comme si nous jouions au lego ou à un puzzle, il suffit d'avoir un peu de logique et c'est tout !

Tout le monde peut créer des programmes pour les téléphones mobiles (smartphones) sans se frotter aux langages de programmation difficiles à comprendre, il suffit d'assembler des blocs de manière graphique, de façon simple, facile et rapide à créer.

### 14. Annexe "Code pour l'algorithme CUA".

Référence à Github : <https://github.com/coinsolidation/source>

### 15. Conditions.

Conditions d'utilisation voir sur le site [www.coinsolidation.org](http://www.coinsolidation.org) ou <https://github.com/coinsolidation/Terms>

Soutien à l'utilisation commerciale.

[support@coinsolidation.org](mailto:support@coinsolidation.org)

Utilisation commerciale de la chaîne de vente en bloc.

[sales@coinsolidation.org](mailto:sales@coinsolidation.org)

Informations juridiques et questions ou préoccupations concernant les licences

[legal@coinsolidation.org](mailto:legal@coinsolidation.org)

Les réseaux sociaux :

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

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