

Patrones de diseño en Wallets para privacy coin en dispositivos móviles.

El caso de estudio de Zcash

Mg. Francisco Gindre fgindre@lifa.info.unlp.edu.ar

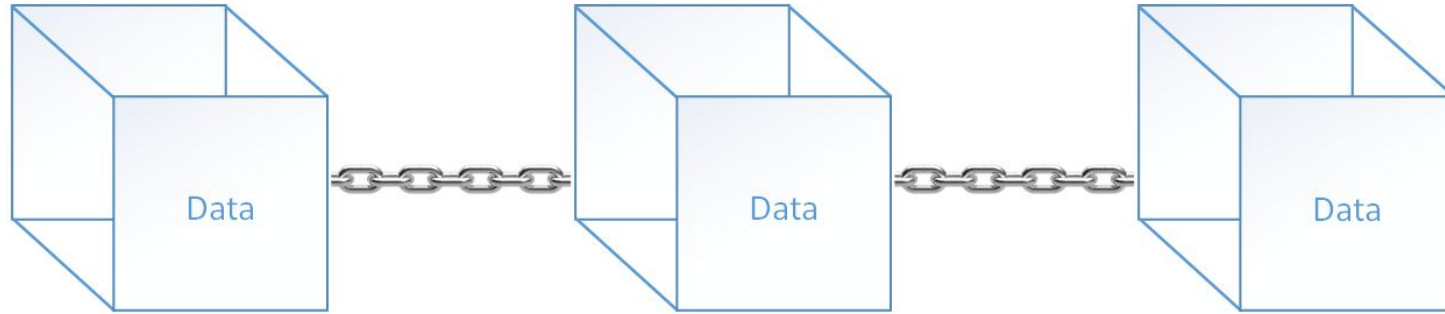
Dr. Matías Urbieto

Facultad de Informática - Universidad Nacional de La Plata

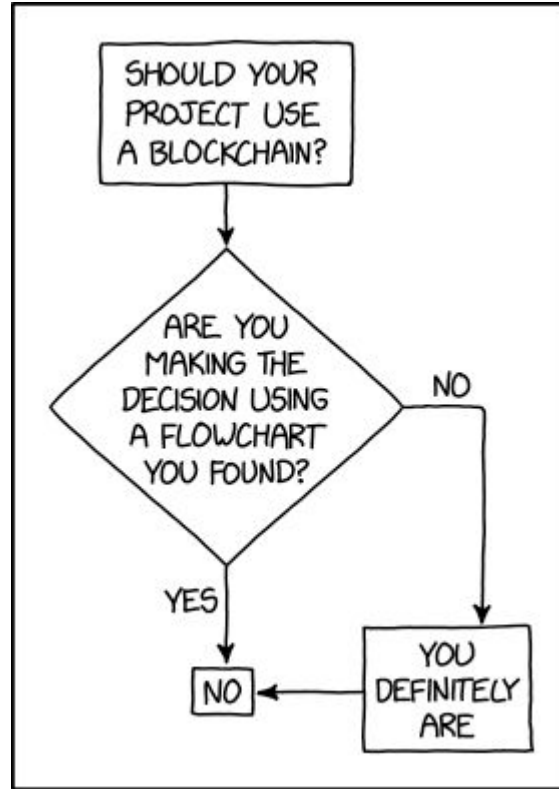
Introducción

Blockchain

Blockchain:



Regla de Oro - ¿Mi proyecto necesita blockchain?



<https://imgs.xkcd.com/comics/blockchain.png>

Blockchain

También es un Patrón

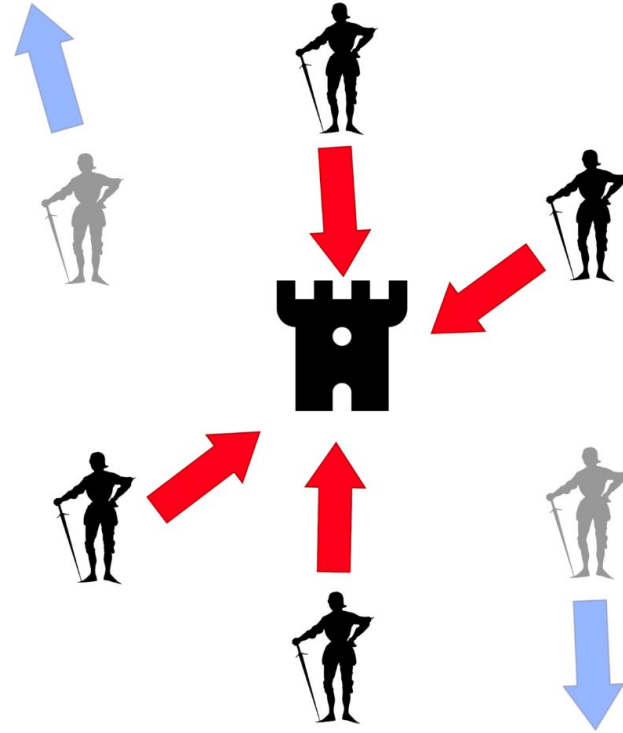
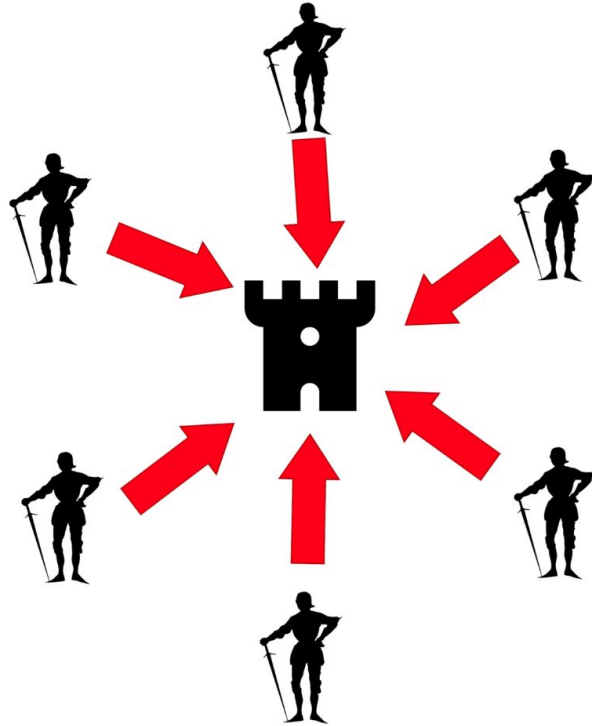
Problemas que soluciona en una red de pares

- Confirmación de información en base a un protocolo de consenso
- Inmutabilidad sobre la información acordada en base al consenso
- Resistencia a censura: evitar que los buenos actores sean privados de operar en la red
- Resistencia a la captura: para poder romper el consenso requiere lograr el 51% de los nodos de la red. (Sybill resistance)
- Tolerancia al problema de los generales bizantinos (BFT Resistance)

https://en.wikipedia.org/wiki/Byzantine_fault

https://en.wikipedia.org/wiki/Sybil_attack

Byzantine Fault



¿Cual es concepto principal?

Consenso y Coso

Blockchain

Como Efectivo Electrónico

Bitcoin - peer to peer electronic cash system

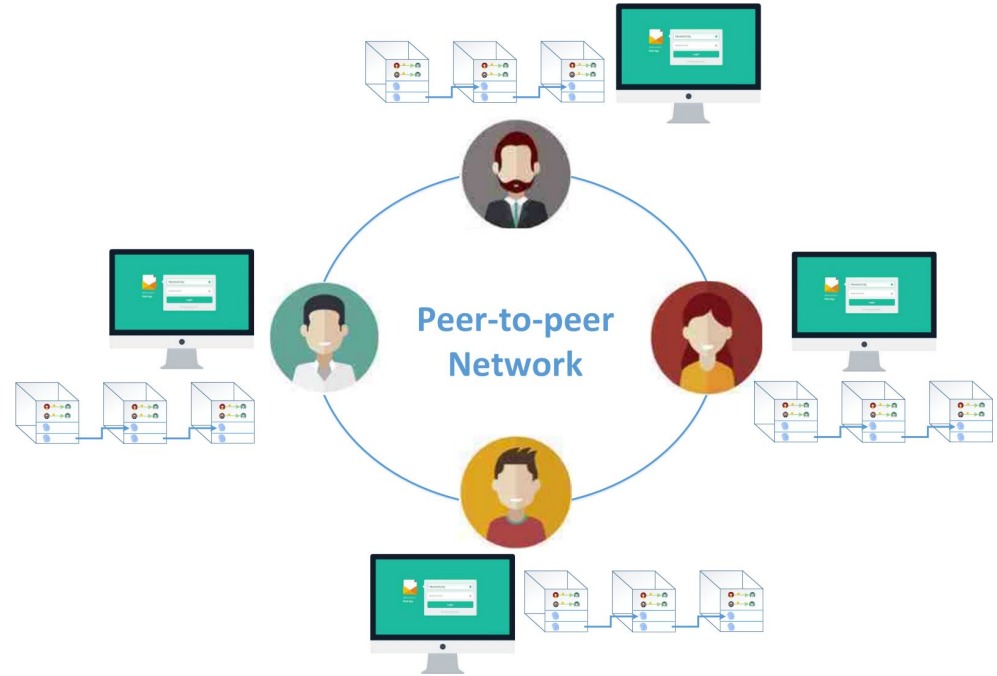
- Asiento contable acordado mediante un protocolo de consenso
- Inmutabilidad sobre la información acordada (blockchain)
- Resistencia a censura: evitar que los buenos actores sean privados de operar en la red
- Resistencia a la captura: para poder romper el consenso requiere lograr el 51% de los nodos de la red. (Sybill resistance)
- Tolerancia al problema de los generales bizantinos (BFT Resistance)
- Resistente a la falsificación (Double-Spend, Finney Attack)

https://en.wikipedia.org/wiki/Byzantine_fault

https://en.wikipedia.org/wiki/Sybil_attack

<https://bitcoin.stackexchange.com/questions/4942/what-is-a-finney-attack>

Sin autoridad central

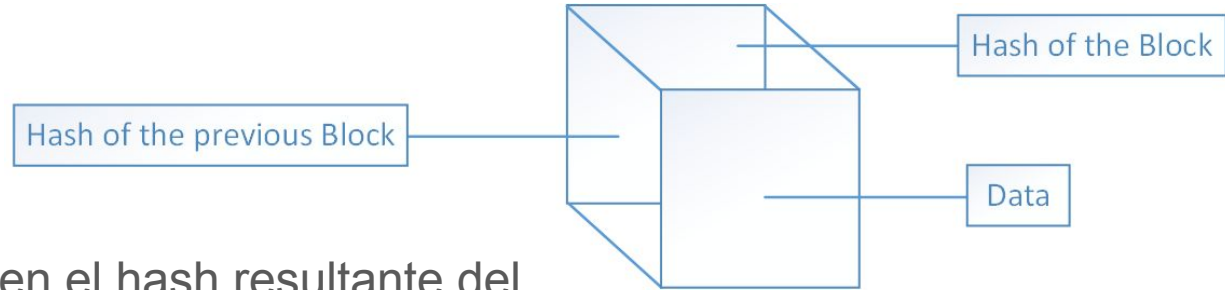


+ Algoritmo de consenso

Si todos disponemos de la misma copia de la cadena de bloques y actuamos acorde al consenso, no se requiere una autoridad coordinadora.

Asiento contable (ledger) disponible al público

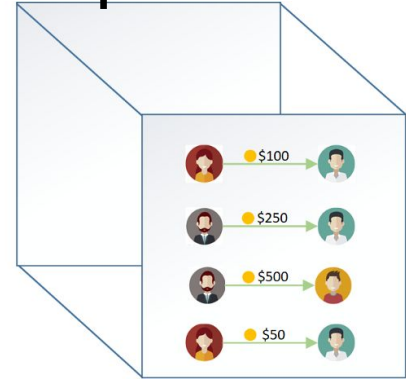
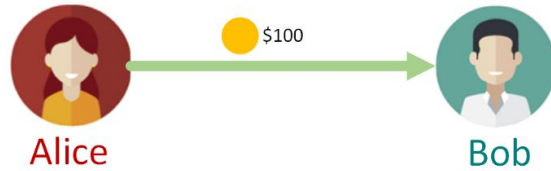
Cada bloque esta atado al anterior mediante el Cálculo de su Hash



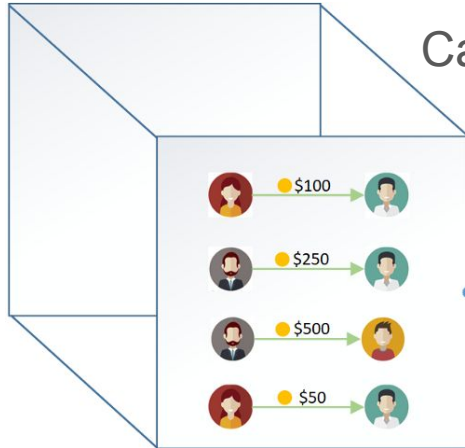
Que también está incluido en el hash resultante del Bloque actual.

Cada bloque contiene los datos de las transacciones que lo conforman

Asiento contable (ledger) disponible al público



Cada bloque contiene los datos
de las transacciones que lo conforman



Hash

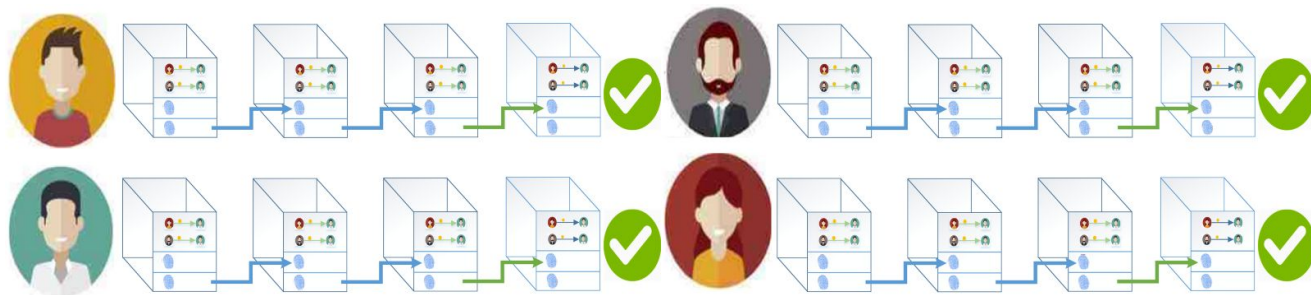


64ac1a4e8097ee9de2a0b
97deb635f0d3e3dfc5c84a
84edd99d4f0e2a1d95761

Consenso sobre nuevos bloques

Cada vez que se crea un nuevo bloque todos los integrantes de la red de pares deben verificar que sea válido y agregarlo a su copia de la cadena de bloques.

De esa forma se garantiza que la única forma de alterar la cadena es tener la mayoría del poder de cómputo (51% attack). Se considera que en redes consolidadas, tales ataques son es anti-económicos.



Potestad Criptográfica en una blockchain

La identidad de los actores se basa
En poder firmar digitalmente los
Valore que se guardan en la cadena
De bloques.



PRIVATE KEY

6831728990636725551934513790552817929570764
7578558684440512287097919467220420

La clave pública es como una
Casilla de correo que se dispone públicamente y a la que cualquiera que la
conozca puede enviarle mensajes.



PUBLIC KEY

044e554e13e016a83a958197cf3b8622b9afc5b9ea04
bdf37e1ef20a2dabcfa7d180ba760ec74408abadd246
8bc5415d67305dd679d4bd1610c72f0aff57dc1ab3

La clave privada es la llave de esa casilla postal la cual debería ser celosamente
custodiada por su propietario. Ella da acceso a todos los mensajes existentes.

Potestad Criptográfica en una blockchain

La identidad de los actores se basa

En poder firma

Valore que se s

De bloques.

La clave públic

Casilla de corre

conozca puede

La clave privada

custodiada por

Quien tenga la clave
privada, tiene la potestad
de los fondos asociados a
ella.

52817929570764
9467220420

522b9afc5b9ea04
c74408abadd246
72f0aff57dc1ab3

que la

elosamente
xistentes.

Potestad Criptográfica en una blockchain

Una forma común de generar claves privadas es mediante códigos Mnemónicos tal como lo estipula BIP-39. Utilizando palabras de un diccionario predefinido

burst mechanic draw sign wolf easy priority supply render regular nature crunch

Que se transforman en Bytes de forma determinística, para generar las claves privadas y luego derivar las Públicas y las direcciones.

```
9cef595f241bab37b49999c2ebb381ed6f43dd842b996635cd6dc6a1b8ee92eee753eebaa4730a2259e18c4cc5fa5666bb3e8e76bb40cc65bdc938dafd2c6fe1
```

<https://iancoleman.io/bip39/>

Bitcoin

Transacción: Alice le pide a Bob
unas pizzas. Paga con BTC

“Bob, mandame dos de muzza. pago BTC 100% barrani”

“Alice, te querés mword” - Bob, crypto millonario, ex-pizzero

Summary

USD

BTC

Hash a1075db55d416d3ca199f55b6084e2115b9345e16c5cf302fc80... 

2010-05-22 15:16

1XPTgDRhN8RFnzniWCddobD9iKZatrvH4	150.00000000 BTC 
1XPTgDRhN8RFnzniWCddobD9iKZatrvH4	250.00000000 BTC 
1XPTgDRhN8RFnzniWCddobD9iKZatrvH4	150.00000000 BTC 
1XPTgDRhN8RFnzniWCddobD9iKZatrvH4	80.00000000 BTC 
1XPTgDRhN8RFnzniWCddobD9iKZatrvH4	0.01000000 BTC 
1XPTgDRhN8RFnzniWCddobD9iKZatrvH4	0.01000000 BTC 
1XPTgDRhN8RFnzniWCddobD9iKZatrvH4	0.01000000 BTC 
1XPTgDRhN8RFnzniWCddobD9iKZatrvH4	0.01000000 BTC 
1XPTgDRhN8RFnzniWCddobD9iKZatrvH4	0.01000000 BTC 
1XPTgDRhN8RFnzniWCddobD9iKZatrvH4	0.01000000 BTC 



17SkEw2md5avVNyYgj6RiXuQKNwkXaxF... 10000.00000000 BTC 

[Load more inputs... \(121 remaining\)](#)

Fee 0.99000000 BTC
(4191.363 sat/B - 1047.841 sat/WU - 23620 bytes)

10000.00000000 BTC

Alice envía fondos a Bob

Alice crea debe tomar uno de los Inputs (fondos disponibles) que ha recibido anteriormente y crear una nueva transacción



Para ello debe firmarla con su clave privada que es la que tiene potestad sobre los fondos (Spend authority)

Alice envía Fondos a Bob

Una transacción es como una registro en un asiento contable. Hay entradas de fondos, y salidas.

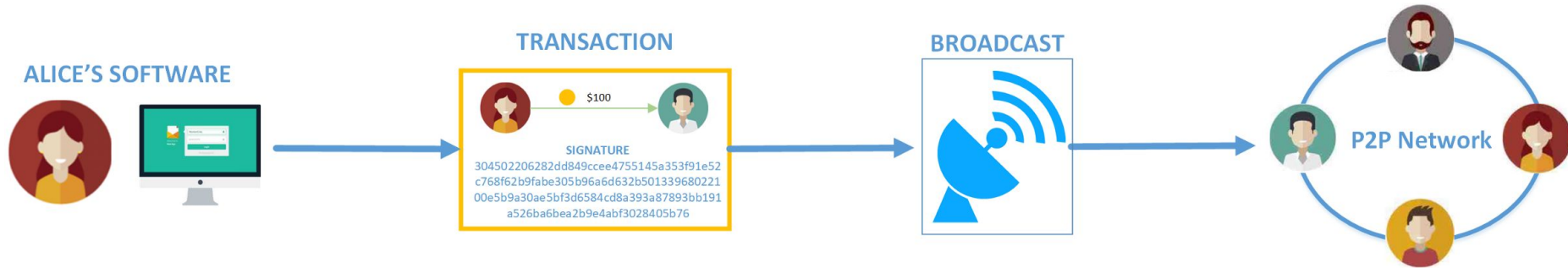
Donde, $\text{entradas} > \text{salidas}$

que toda diferencia es considerada como la “comisión del minero”.

El “cambio” también es una salida, y el destinatario es el remitente.

Transaction as Double-Entry Bookkeeping			
Inputs	Value	Outputs	Value
Input 1	0.10 BTC	Output 1	0.10 BTC
Input 2	0.20 BTC	Output 2	0.20 BTC
Input 3	0.10 BTC	Output 3	0.20 BTC
Input 4	0.15 BTC		
Total Inputs:	0.55 BTC	Total Outputs:	0.50 BTC
	<i>Inputs</i>		<i>0.55 BTC</i>
-	<u><i>Outputs</i></u>		<u><i>0.50 BTC</i></u>
	<i>Difference</i>		<i>0.05 BTC (implied transaction fee)</i>

Transmitir la transacción a la red de pares



Verificar la transacción enviada por Alice

ALICE'S MESSAGE SIGNATURE

304502206282dd849ccee4755145a353f91e52c768f62b9fab
e305b96a6d632b50133968022100e5b9a30ae5bf3d6584cd8
a393a87893bb191a526ba6bea2b9e4abf3028405b76



ALICE'S PUBLIC KEY

044e554e13e016a83a958197cf3b8622b9afc5b9ea04
bdf37e1ef20a2dabcfa7d180ba760ec74408abadd246
8bc5415d67305dd679d4bd1610c72f0aff57dc1ab3

Verify

TRANSACTION VERIFIED



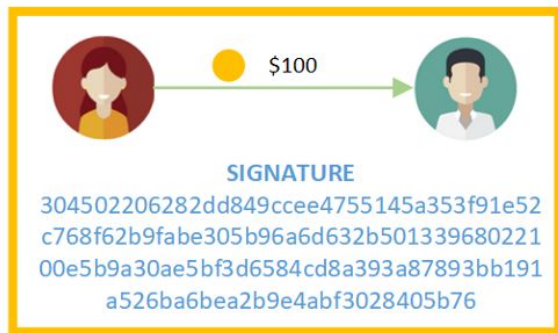
SIGNATURE

304502206282dd849ccee4755145a353f91e52
c768f62b9fabe305b96a6d632b501339680221
00e5b9a30ae5bf3d6584cd8a393a87893bb191
a526ba6bea2b9e4abf3028405b76



Incluir la transacción en un bloque

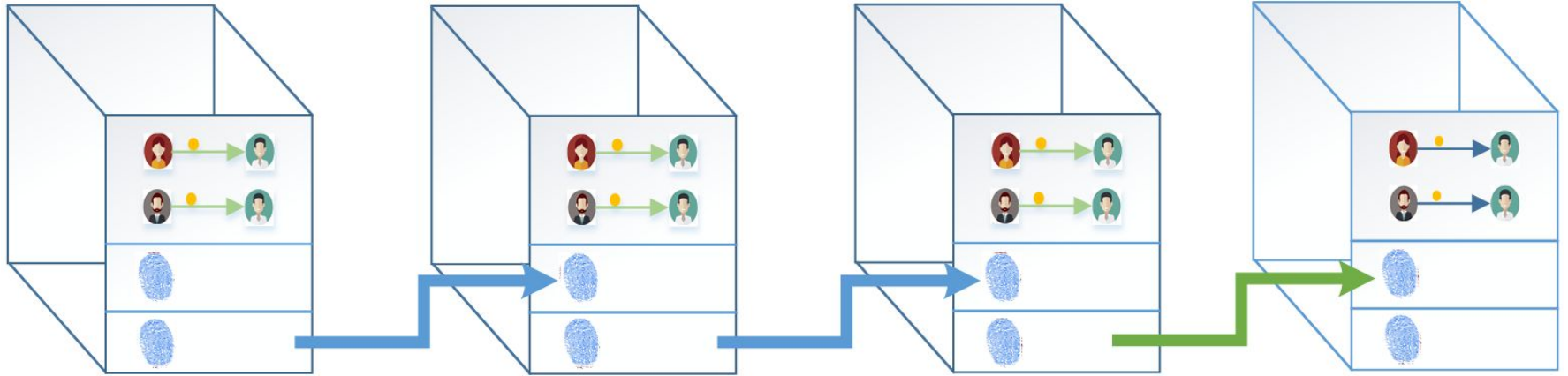
VERIFIED TRANSACTION



Include



Añadir el bloque al extremo de la cadena



Cómo construir una
aplicación móvil que
interactúa con una
blockchain

¿Qué es una wallet?

En las blockchains, no existe una entidad “Billetera”.

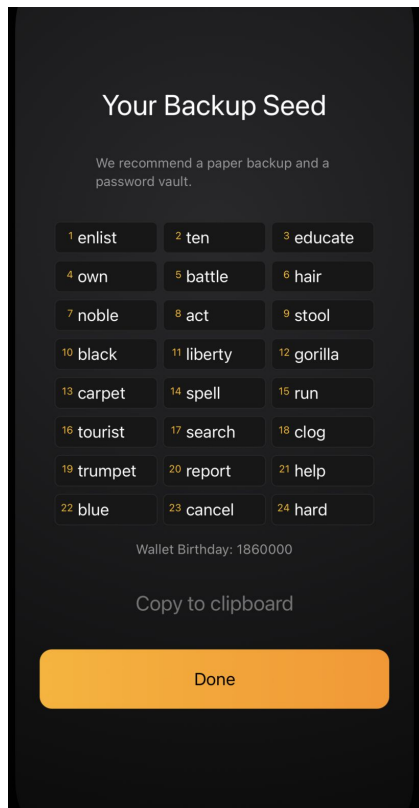
Una wallet es una representación de los datos de una blockchain desde el punto de vista de un conjunto de claves privadas.

Elementos como el balance y las transacciones, son subproductos de un proceso denominado sincronización que consiste en tomar claves públicas y recorrer la cadena de bloques recolectando los elementos de interés para las mismas

Requerimientos de una wallet

- Custodia soberana de las claves del usuario.
 - Not your Keys, not your coins
 - Gestión de claves privadas
 - Claves Mnemónicas
- Utilizar criptomonedas con privacidad en el asiento contable (Privacy Coins)
 - Zcash
 - Monero
- Desencriptación en el propio dispositivo
- Enviar fondos
- Recibir fondos
- Historial de transacciones
- Estar al día con la blockchain

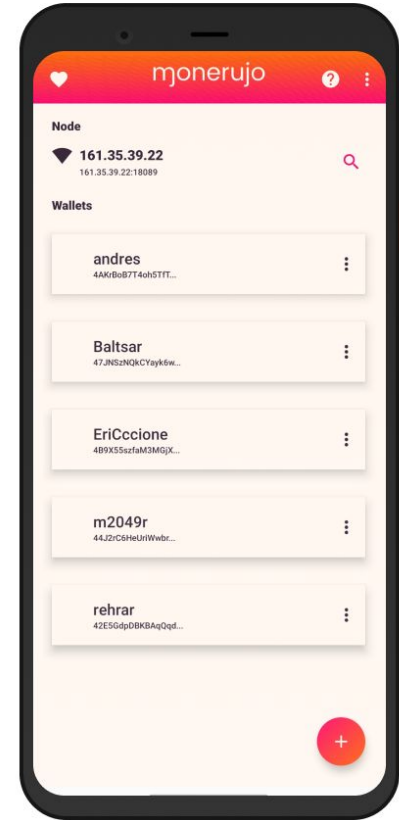
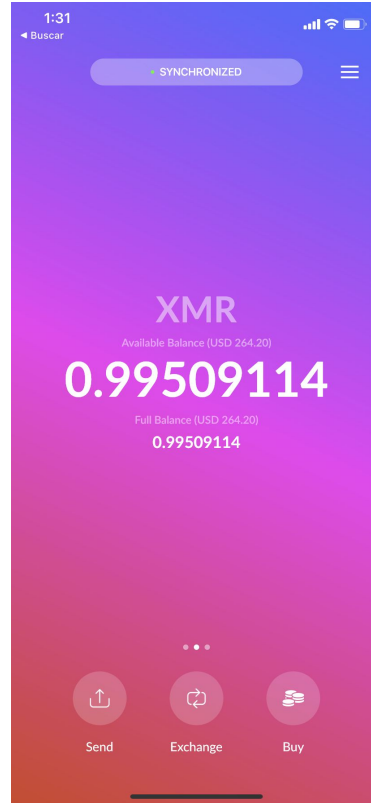
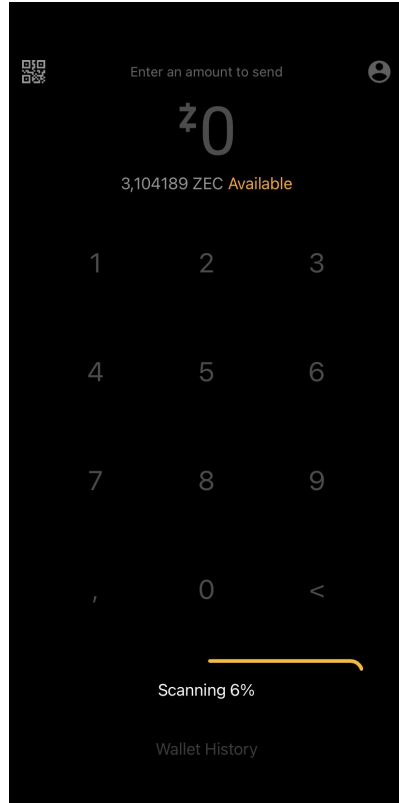
Gestión soberana de claves privadas



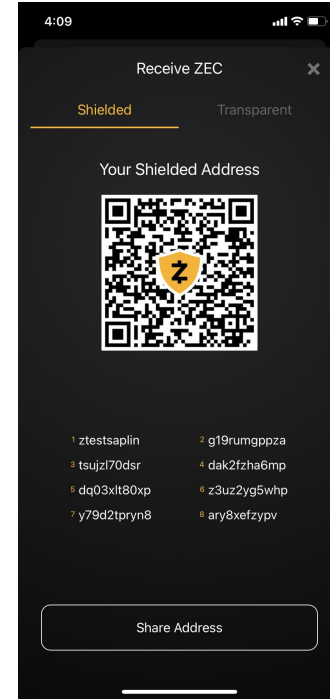
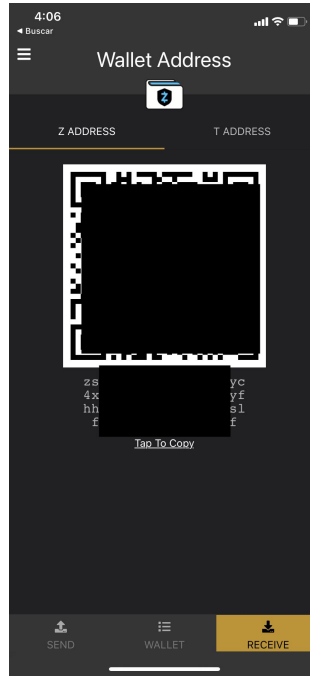
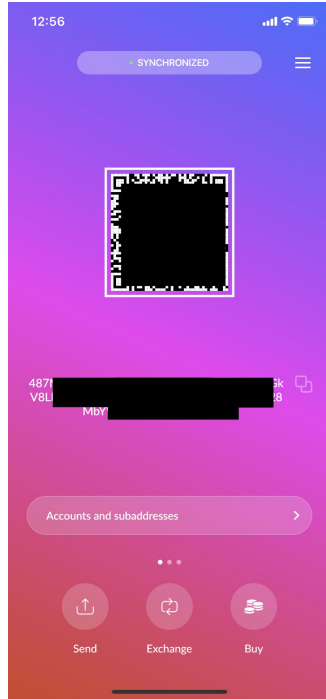
Para experimentar con frases mnemónicas

<https://iancoleman.io/bip39/>

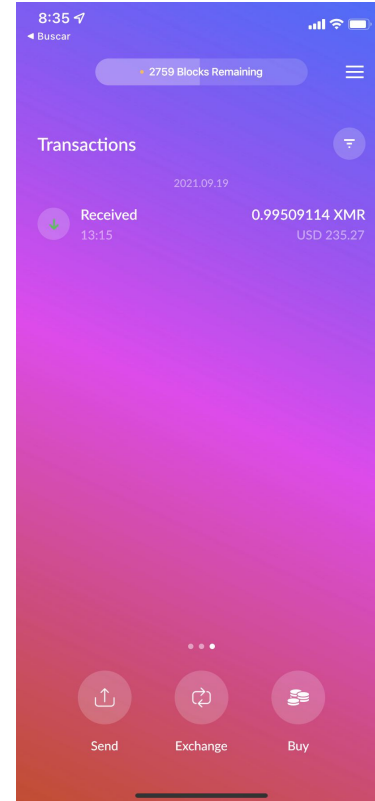
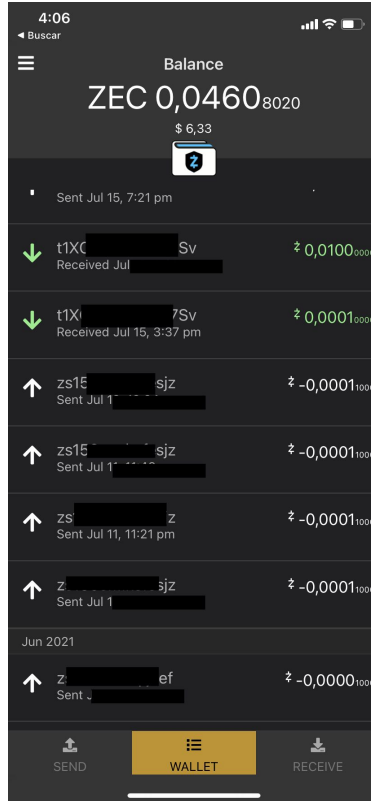
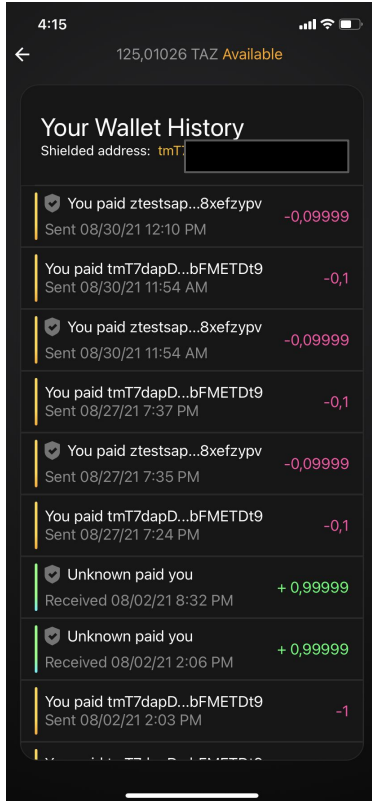
Sincronizar




Recibir fondos




Ver historial



Enviar Fondos



Enter an amount to send



₺0,0001

3,104189 ZEC Available

1

2

3

4

5

6

7

8

9


,

0

<

Send

Wallet History




₺0,0001

from your shielded wallet

Send

To: t1XC6FF



This is your Auto Shielding address

On Clipboard

Your Auto Shielding Address (+)

t1XC6FF


LAST USED


Unknown

zs1wq6m5



Transaction Details





4pm

Total Spent

₺0,0001

0,000010 network fee

from your shielded wallet

to t1X

Pending confirmation

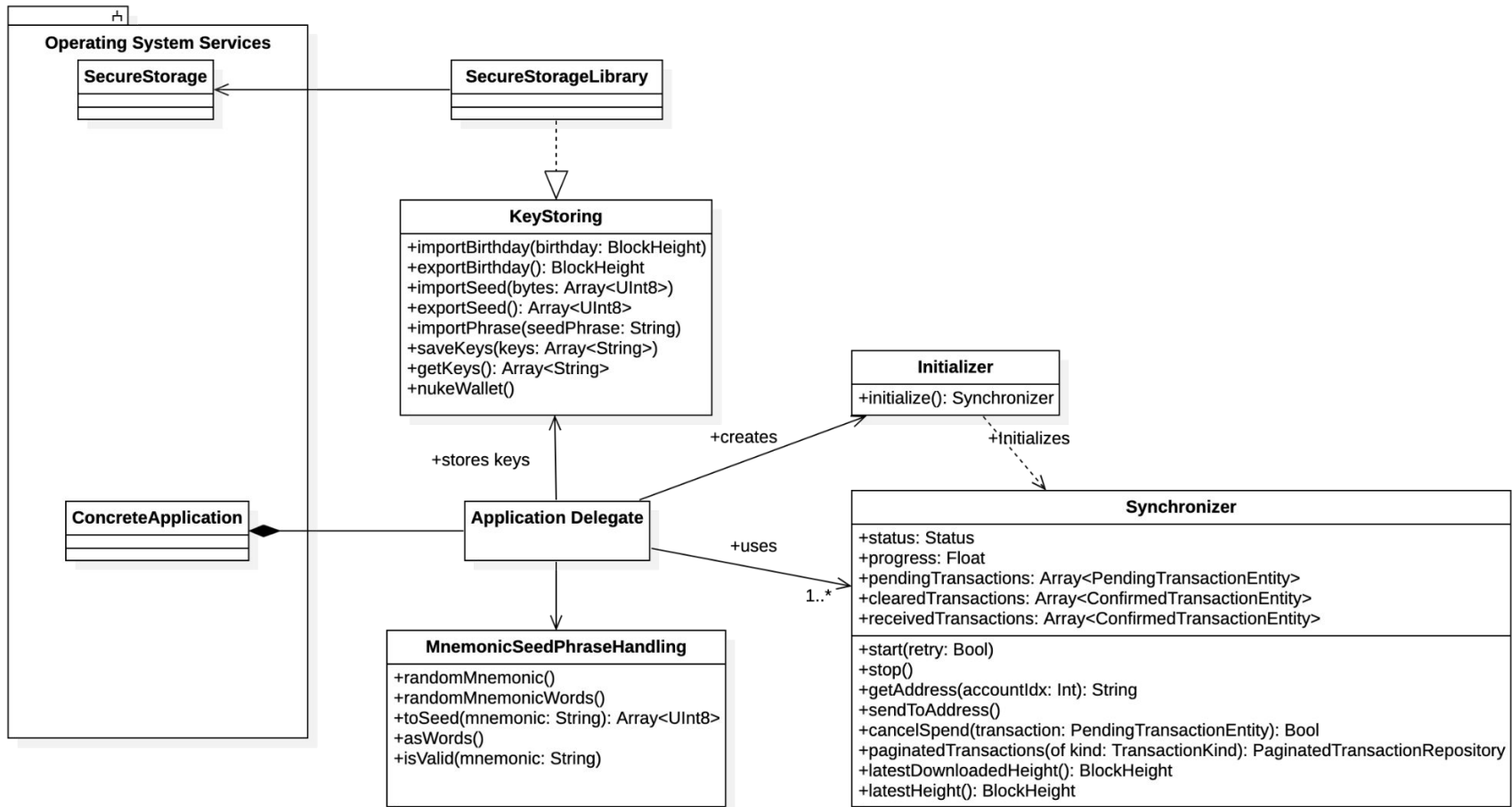
Total Spent

₺0,00011

Arquitectura y patrones propuestos

Ingeniería inversa / Análisis de Requerimientos

- En los repositorios
 - Buscar historias de usuario, feature requests, etc
 - Documentación
 - Análisis del Código Fuente
- En las aplicaciones
 - Contrastar con lo encontrado en los repositorios
 - Analizar casos de uso presentes en la Interfaz Gráfica
 - Comparar la funcionalidad entre todas las Apps
- En la documentación oficial de Zcash y Monero
 - Análisis del protocolo
 - Documentación referente a wallets
 - Buscar requerimientos no funcionales



Frases de Recuperación

MnemonicSeedHandling:
manejo de frases Mnemónicas

Identidad y Claves

history	salad	panther	clog	chapter	trumpet
random	service	notice	bottom	rival	pool
task	middle	major	venture	cousin	notice
hub	apart	tube	pear	hospital	cable

Seed Bytes

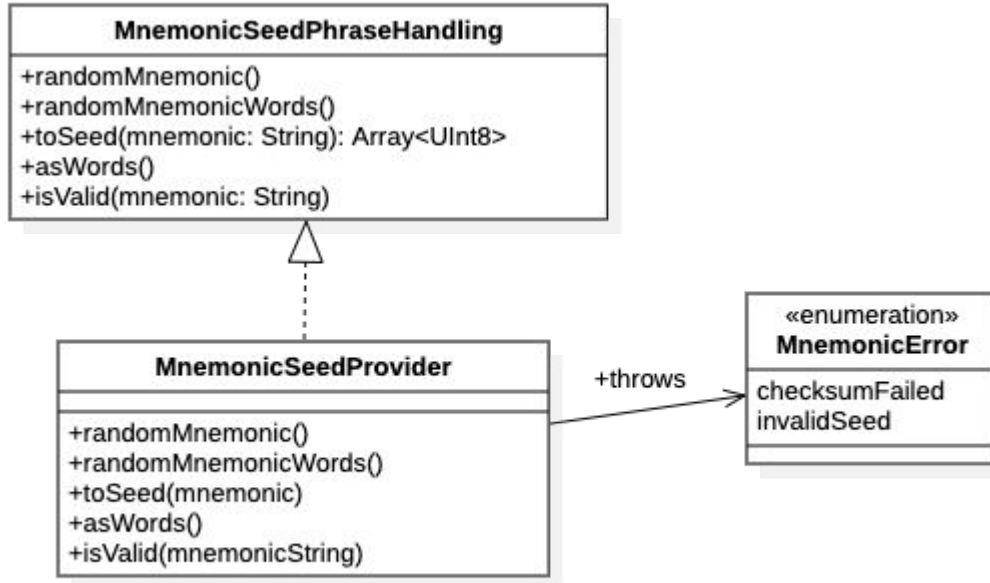


Direcciones

Derivación

50e76d494f6d551b9d2b39a
a0ebbc911f661b6f7bd440b2
159cc67bd21bdad31a145c8
ef943cf0077334bcecb4244a
91483cf83fd6ff4202bc583ffc
1f0af4d3

MnemonicPhraseHandling: Manejo de frases semilla



MnemonicPhraseHandling: Participantes

MnemonicPhraseHandling: la interfaz propuesta que condensa los requerimientos enumerados anteriormente.

MnemonicSeedProvider: es la implementación de la interfaz. Puede implementarla directamente o delegar esta implementación a una librería en cuyo caso actuaría como adaptador entre la interfaz propuesta y la existente en la librería utilizada.

MnemonicError: representa los errores posibles para estos requerimientos. Estos pueden nuclearse en dos errores primarios. Uno es **checksumFailed**, que refiere a la comprobación del checksum resultante de convertir la frase provista a bytes y verificar que esta frase, cuyas palabras corresponden al diccionario utilizado, sea íntegra en base al estándar utilizado. El otro es **InvalidSeed** refiere a que la frase propuesta es inválida en términos del diccionario propuesto.

MnemonicPhraseHandling: Usos

- **Una técnica:** La frase semilla
- **Distintos estándares:** BIP-39 es el más utilizado pero algunas monedas implementan el propio (ej: monero)
- Disponer de una forma de abstraer estas particularidades es una forma de evitar errores en la generación de claves privadas y la posible pérdida de fondos.

MnemonicPhraseHandling: Consecuencias

- Adapter stack: se debe tener precaución al hacer un wrapper de una librería para evitar este anti-patrón



MnemonicPhraseHandling: Consecuencias

- **Coexistencia de estándares:**
Si una wallet multi-monedas soporta activos que utilizan distintos tipos de claves, puede representar un problema no tenerlos separados adecuadamente.



MnemonicPhraseHandling: Detalles de Implementación

- **Fail fast:** no devolver opcionales. Solo resultados positivos, excepciones y/o errores.
- **No booleans allowed:** la no validación de una frase semilla es un error grave que puede significar o desencadenar una pérdida de fondos
- **Cuidado con la localización:** no mezclar diccionarios
- **Performance:** las operaciones con frases semillas pueden ser ineficientes.

MnemonicPhraseHandling: Detalles de Implementación

- **Don't DIY:** no lo hagas tu mismo. Utiliza sólo código fuente confiable y auditado.
 - La simpleza de la propuesta BIP-39 puede ser tentadora para implementar una librería propia.
 - Una vulnerabilidad en este punto pone en peligro los fondos de los usuarios.
- **Temas relacionados:** Patrones Proxy y Adapter del GoF

Manejo de claves de usuario

KeyStoring: Almacenamiento de Claves

Identidad y Claves

history	salad	panther	clog	chapter	trumpet
random	service	notice	bottom	rival	pool
task	middle	major	venture	cousin	notice
hub	apart	tube	pear	hospital	cable

Seed Bytes



Direcciones

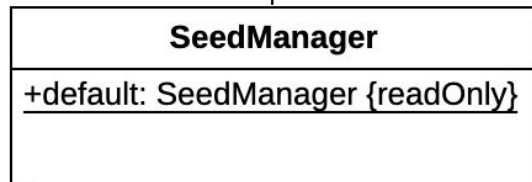
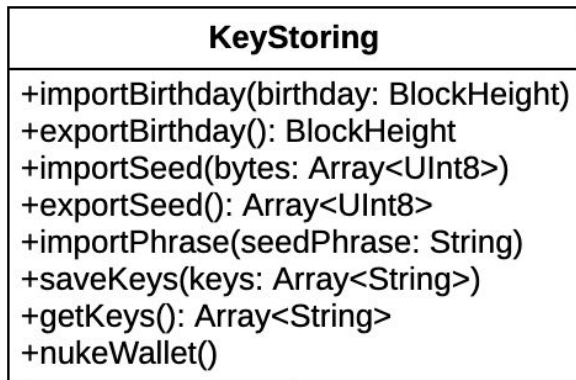
Derivación

50e76d494f6d551b9d2b39a
a0ebbc911f661b6f7bd440b2
159cc67bd21bdad31a145c8
ef943cf0077334bcecb4244a
91483cf83fd6ff4202bc583ffc
1f0af4d3

Identidad y Claves

Quien posee las claves privadas,
Controla los fondos

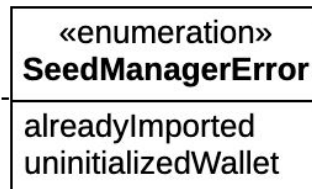
KeyStoring: almacenar claves y datos sensibles



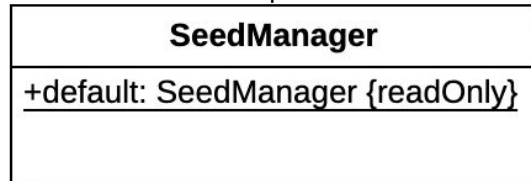
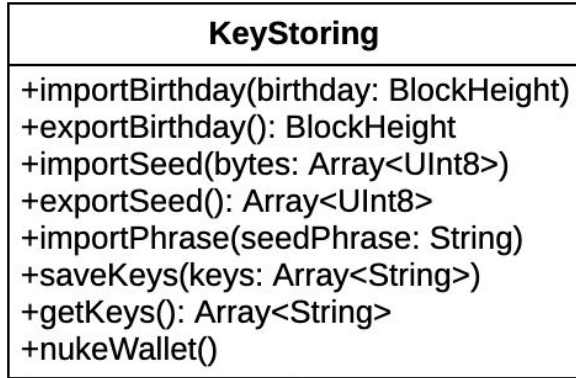
Participantes:

SeedManager: implementación concreta

SeedManagerError: transformación de los errores concretos de la implementación al dominio de la aplicación.



KeyStoring: almacenar claves y datos sensibles



throws



Colaboraciones:

Esta interfaz se utiliza en conjunto con
MnemonicSeedHandling

Detalles de implementación

- ¿Qué sucede con ellas cuando el usuario elimina la aplicación?
- ¿son las claves eliminadas automáticamente por el sistema o permanecen en él hasta que el usuario las elimina manualmente?
- ¿están incluidas en las copias de respaldo del sistema?
- ¿qué sucede al reinstalar la aplicación en caso de que haya claves pre-existentes?

Inicializar el ambiente de una wallet

Initializer: inicialización.

Zcash SDK

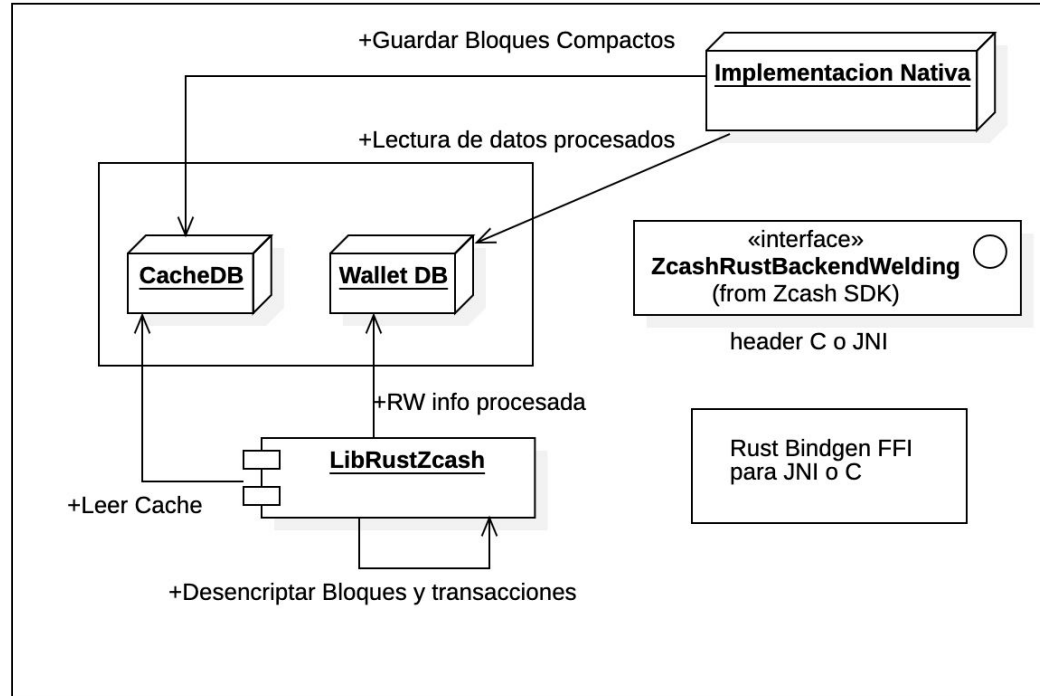
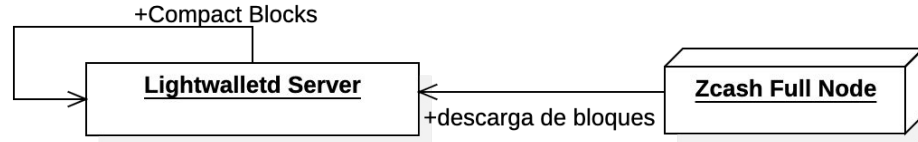
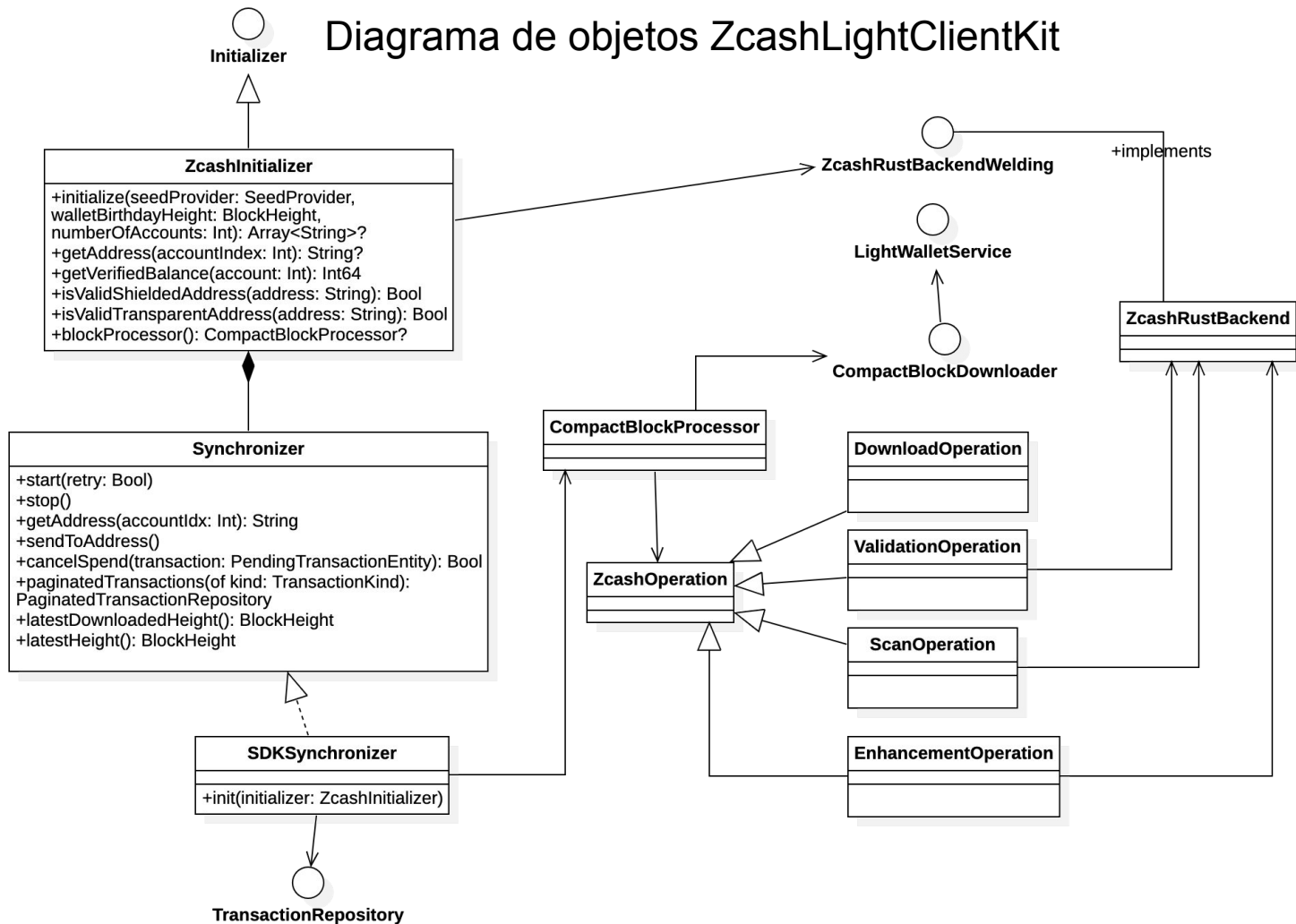
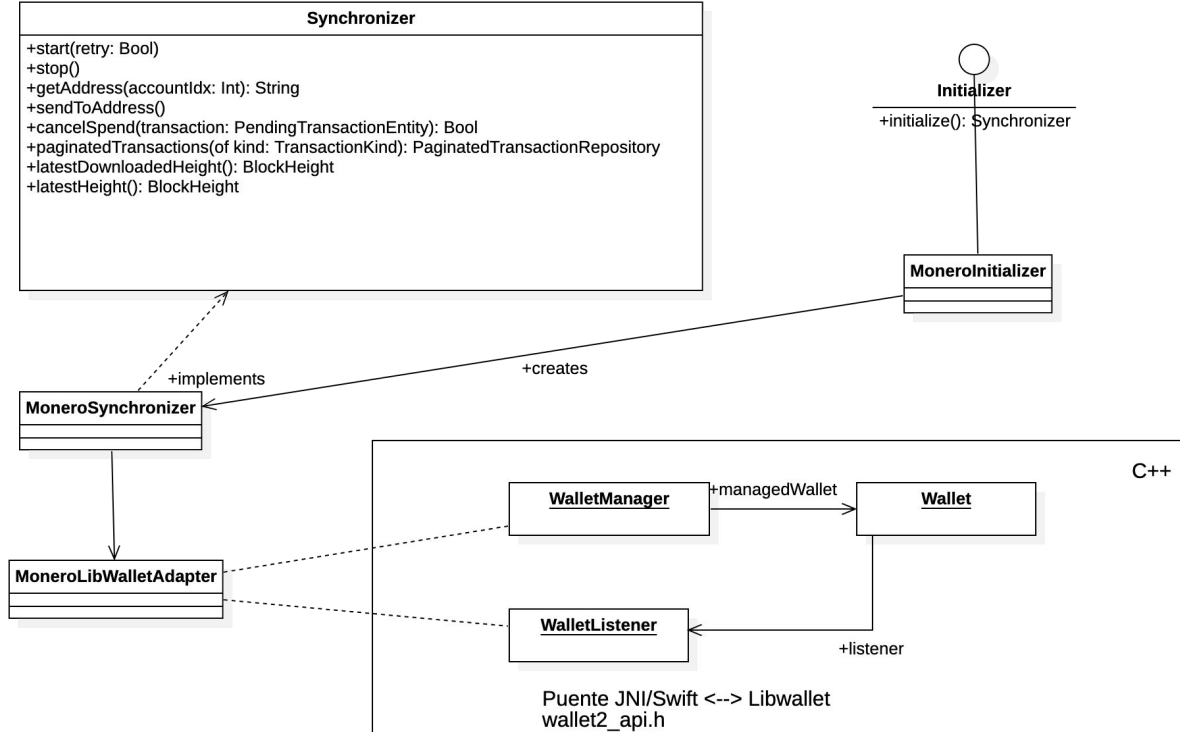


Diagrama de objetos ZcashLightClientKit



Monero SDK (propuesta)



Initializer: Intención

Proveer un mecanismo para encapsular y abstraer la complejidad de inicializar los componentes requeridos para poder sincronizar una cadena de bloques y controlar el grafo de objetos y entidades derivados de éstos

Initializer: Colaboración

Clases Façade que proveen recursos de interfaces foráneas (FFI), Synchronizer

Initializer: Participantes

Recursos del sistema: el inicializador recibe referencias a los recursos del sistema que se requieren para poder sincronizar una cadena de bloques.

Recursos de la aplicación cliente: referencias a los recursos de la aplicación que se requieren para poder sincronizar una cadena de bloques, como por ejemplo archivos de bases de datos, parámetros, referencias a clases de FFI.

Recursos provistos por el usuario: puede requerir claves o datos provistos por o relacionados al usuario o sus claves.

Objetos inicializados: Distintas clases que utilicen recursos nucleados en el inicializador pueden utilizarlo como parámetro o como constructor.

Initializer: Consecuencias

Objeto Dios (God Object):

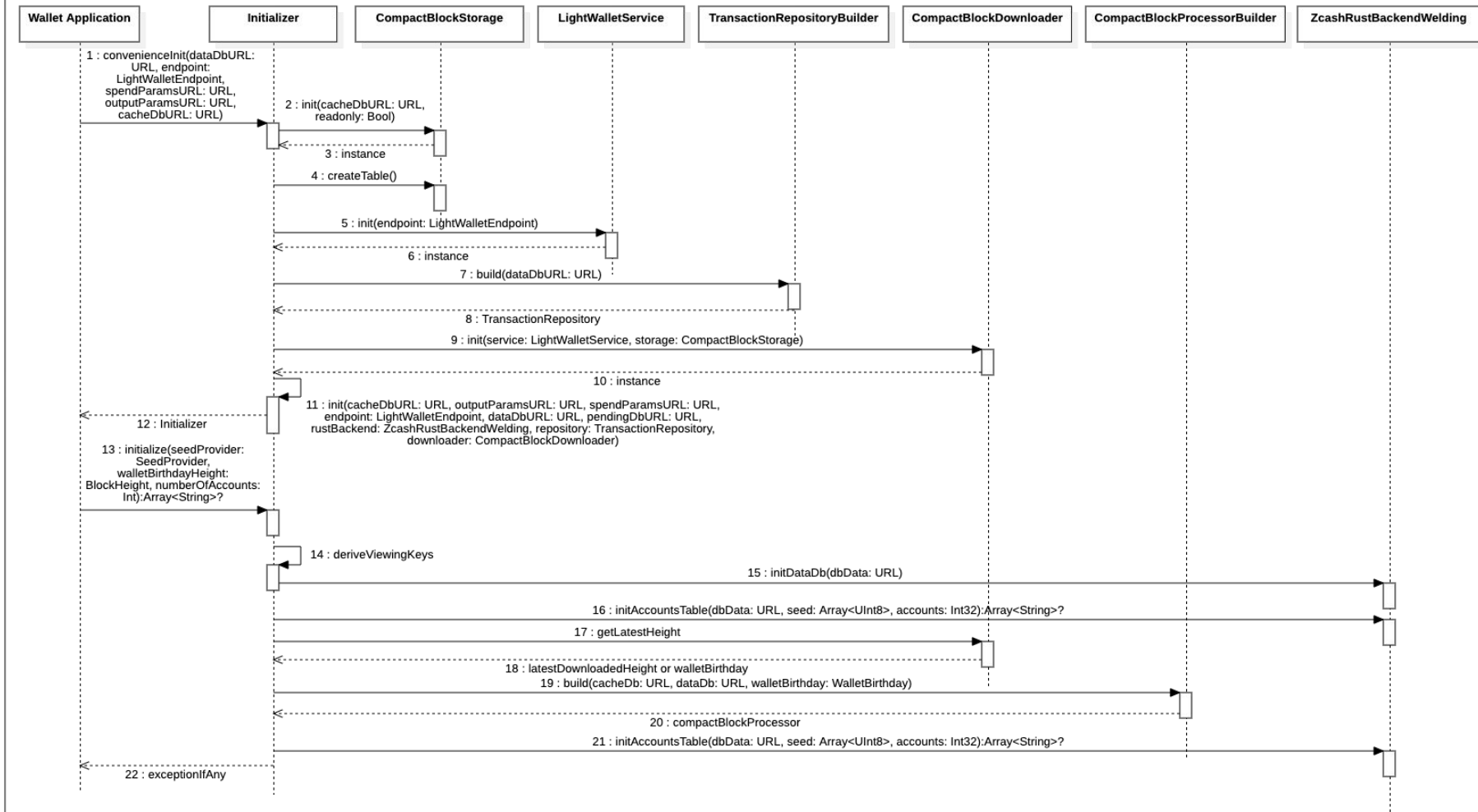
- el inicializador puede terminar siendo un lugar donde toda la lógica compleja del dominio se concentre por “conveniencia” y “practicidad”.
- Puede terminar siendo la “zona demilitarizada” donde todo desacuerdo del equipo de desarrollo se vuelca aquí.

Initializer: Usos

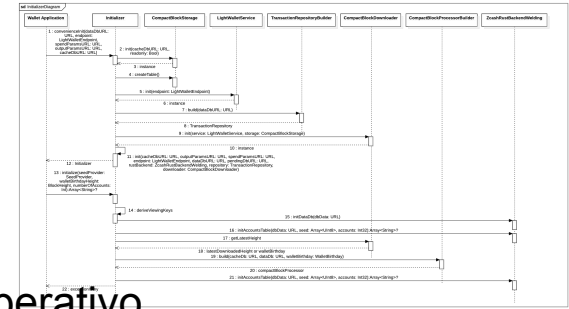
- La noción de una clase inicializadora se encuentra presente en varios kits de desarrollo y frameworks. En el caso de las criptomonedas, uno de los más prominentes podría ser la instancia Web3
- El de SDK de Zcash utiliza este concepto para concentrar toda el comportamiento creacional en esta instancia.

Initializer: Detalles de implementación

sd InitializerDiagram



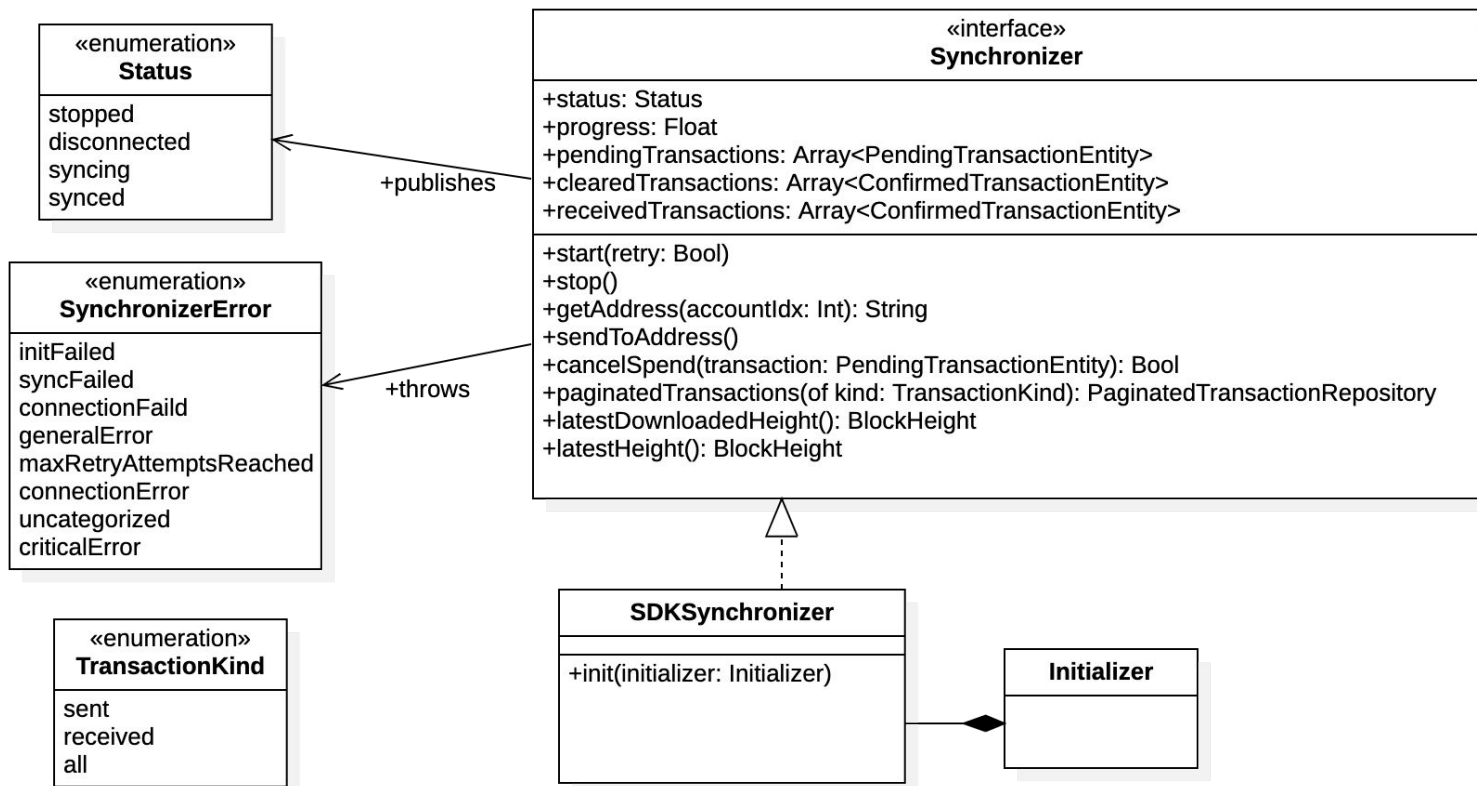
Initializer: Detalles de implementación



- “Es todo un tema”
- Puede involucrar diversos recursos de todo el sistema operativo
 - Operaciones de Entrada/Salida
 - Hilos de Ejecución
 - Distintos ambientes de programación (FFI en C++, Rust, etc)
- Posee todos los ingredientes para una catástrofe
- Esperar errores
 - Pueden no ser recuperables.
 - Planificar su resolución
- Contemplar la inicialización en la experiencia de usuario

Funcionalidad núcleo de una wallet

Synchronizer: sincronización con cadena de bloques



Synchronizer: Intención

Operar en una cadena de bloques se resume en la acción de estar al día con los datos producidos ella desde el punto de vista de un conjunto de claves privadas y/o públicas.

Synchronizer: Motivación

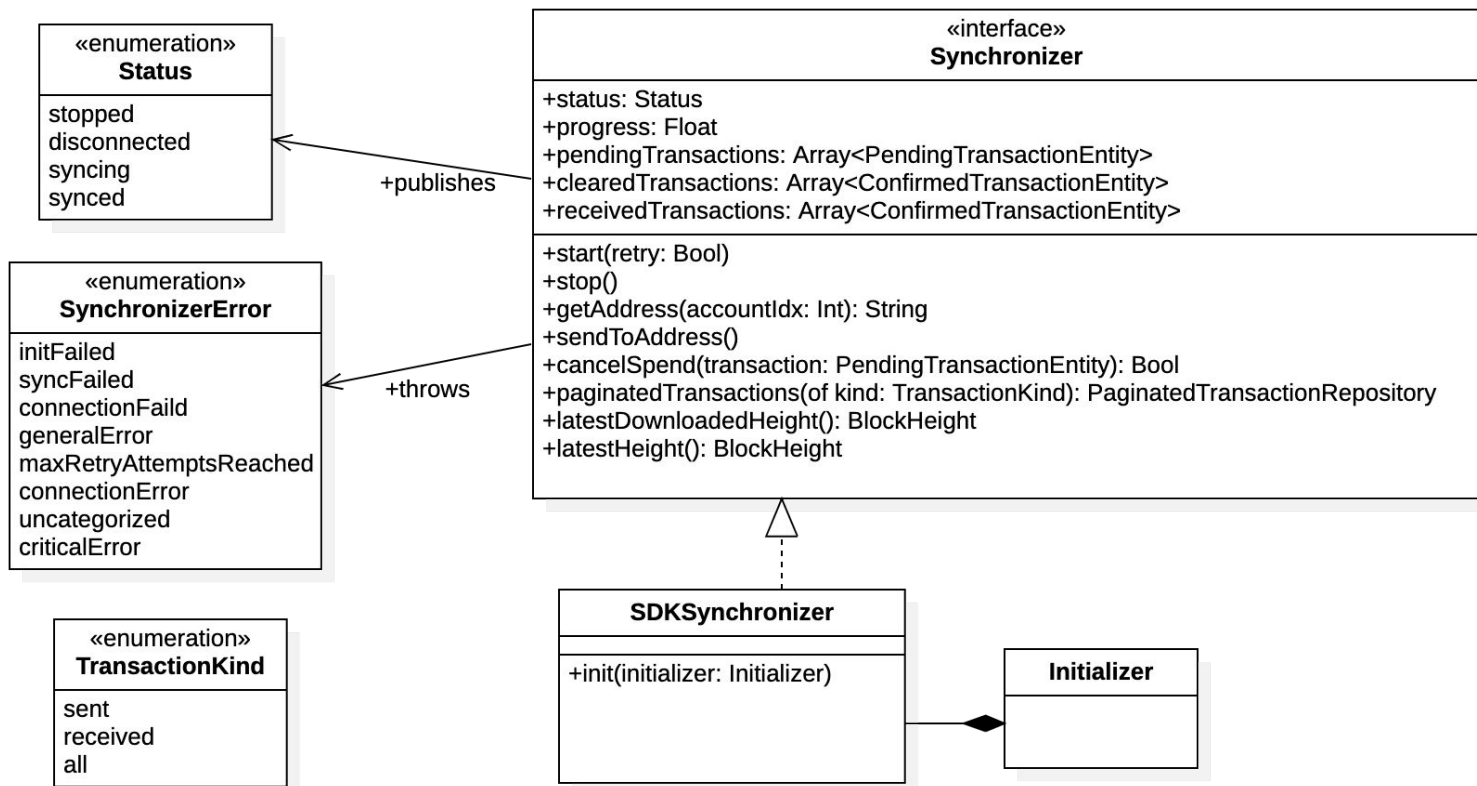
- Abstraer todos los requerimientos funcionales y no funcionales inherentes aun protocolo de consenso de una criptomoneda
- Proveer una interfaz (más) simple y condensada sobre las acciones principales para operar en una criptomoneda
- Sincronizar el la actividad central de una wallet.

Synchronizer: Aplicabilidad

- Todas las wallets sincronizan: local o remotamente.
- Los detalles de implementación le son indistintos al usuario

Synchronizer: participantes

- **SDKSynchronizer**: Instancia concreta. la implementación de la interfaz. Esta tiene adicionalmente la responsabilidad de conocer el ambiente de ejecución y adaptarse al ciclo del vida del mismo.
- **Status**: Expresa el estado del sincronizador. Éste tiene una gran influencia sobre la interfaz de usuario
- **SynchronizerError**: Condensa los posibles fallos que de una forma resumida para poder ser atrapados por el código del cliente y mostrados al usuario (o enmascarados) en una forma adecuada según el criterio del desarrollador
- **TransactionKind**: El tipo de transacciones retornadas por el sincronizador.



Synchronizer: colaboraciones

El sincronizador puede colaborar diversos componentes del sistema, su rol es primordialmente el de condensar los requerimientos para operar dentro de una blockchain de una forma concisa y clara de cara a una aplicación cliente.

Para su creación colabora con un **Initializer**.

Synchronizer: Consecuencias

Buy or Build

interfaz compacta y asertiva <> decisiones de diseño han sido tomadas por terceros

El precio a pagar por la “conveniencia” es generalmente la posibilidad de personalizar el comportamiento de una pieza de software.

Synchronizer: Implementación

- Está ligada tanto a la cadena de bloques que sincronizará como a la plataforma en la que se ejecutará
- Se deben administrar las particularidades de cada plataforma
- El ambiente de programación también influye considerablemente la implementación
 - Versión imperativa
 - Versión Funcional-Reactiva

Synchronizer: Usos

- ZcashSDK
- BitcoinKit SDK -> Clase BitcoinCore condensa una interfaz similar
- Wallet2_api.h en Monero, es una interfaz que tiene funcionalidades similares

Synchronizer: Temas Relacionados

El sincronizador es una interfaz que puede ser considerada una **Façade**. Para su creación se utilizan patrones *creacionales* como **Builder** o **Factory Method**

Funcionalidad núcleo de una wallet

Estrategia de Manejo de Errores

Manejo de Errores

- Pérdida de fondos
- Inconsistencia local con inutilización permanente
- Inconsistencia local con inutilización temporal
- Errores técnicos locales recuperables
- Errores técnicos en servidor

De mayor gravedad a menor.

\0

referencias

Mastering Bitcoin:

<https://github.com/bitcoinbook/bitcoinbook>

ZcashLightClientKit:

<https://github.com/zcash/ZcashLightClientKit/>

Zcash Android SDK:

<https://github.com/zcash/zcash-android-wallet-sdk>

Nighthawk Apps

<https://github.com/nighthawk-apps/zcash-ios-wallet>

<https://github.com/nighthawk-apps/nighthawk-wallet-android>

ECC Reference Wallets:

<https://github.com/zcash/zcash-android-wallet>

<https://github.com/zcash/zcash-ios-wallet>