

**RSK**  
SMARTER BITCOIN

# Contratos Inteligentes en Bitcoin

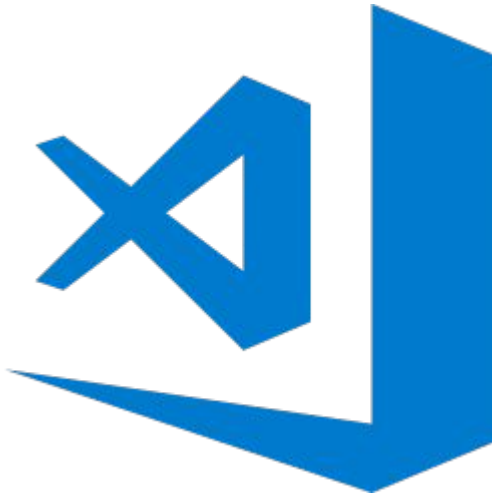
Sebastián Pérez  
RSK Community Tech Evangelist  
sebastian@rsk.co - @sebaleoperez

@RSKSMART  
#SMARTERBITCOIN



# Requerimientos

# Herramientas



1. VISUAL STUDIO CODE
2. PLUGIN "solidity" de Juan Blanco
3. INSTALAR NODE JS
4. INSTALAR TRUFFLE CON NPM
5. INSTALAR JDK DE JAVA



Y lo más  
importante

@RSKSMART  
#SMARTERBITCOIN

Smart contracts powered by the Bitcoin Network.



# Introducción

@RSKSMART  
#SMARTERBITCOIN

Smart contracts powered by the Bitcoin Network.

# Hace unos años...



@RSKSMART  
#SMARTERBITCOIN



Smart contracts powered by the Bitcoin Network.

# Internet del Conocimiento



@RSKSMART  
#SMARTERBITCOIN



Smart contracts powered by the Bitcoin Network.

# Internet del Conocimiento



- ✓ Acceso democrático
- ✓ Comunicación instantánea
- ✓ Bajo costo
- ✗ Info digital es fácil de copiar





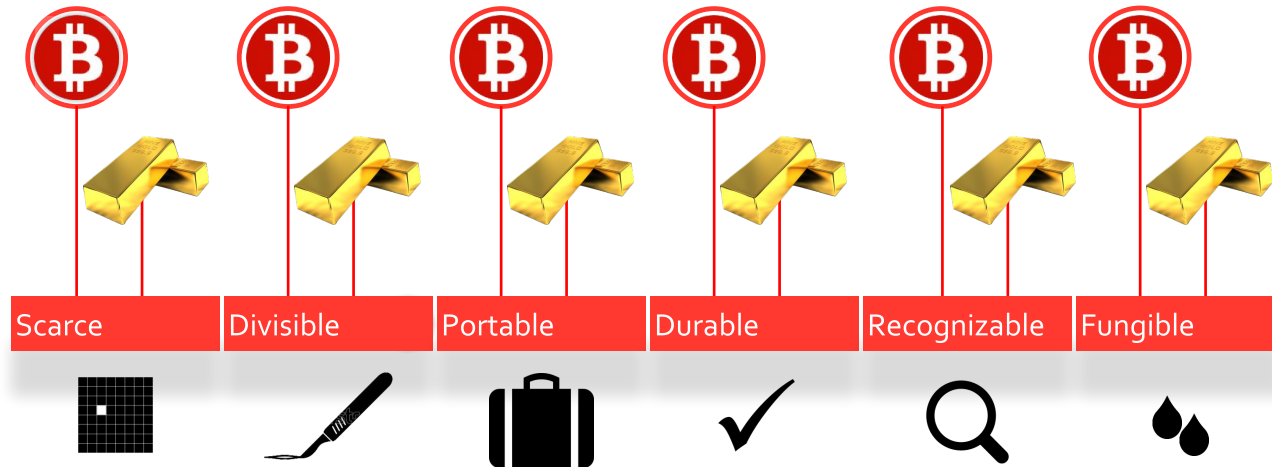
# Internet del Valor

- ✓ Acceso democrático
- ✓ Intercambio de valor
- ✓ Bajo costo
- ✓ Activos digitales únicos



Smart contracts powered by the Bitcoin Network.

# Bitcoin como ejemplo



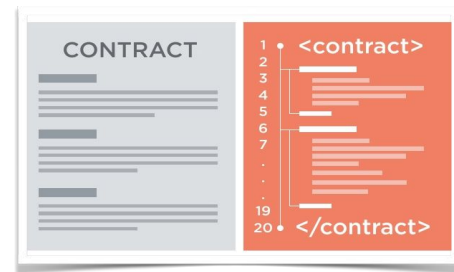
# Bitcoin: Comparativa



	Oro	Dinero (USD)	Bitcoin
Fungible	High	High	High
Portability	Moderate	High	High
Durable	High	Moderate	High
Highly divisible	Moderate	Moderate	High
Secure (not counterfeited)	Moderate	Moderate	High
Easily transactable	Low	High	High
Scarce	Moderate	Low	High
Sovereign (Gov issued)	Low	High	Low
Decentralized	Low	Low	High

# Contratos inteligentes

- 1 Auto-verificable: contiene todas las condiciones del acuerdo.
- 2 Auto-ejecutable: contiene todas las cláusulas del acuerdo.
- 3 Resistente a modificaciones: asegurado por la inmutabilidad de la blockckhain.
- 4 Conciliación automática: info no repudiable y/o disputable.





## Misión 2

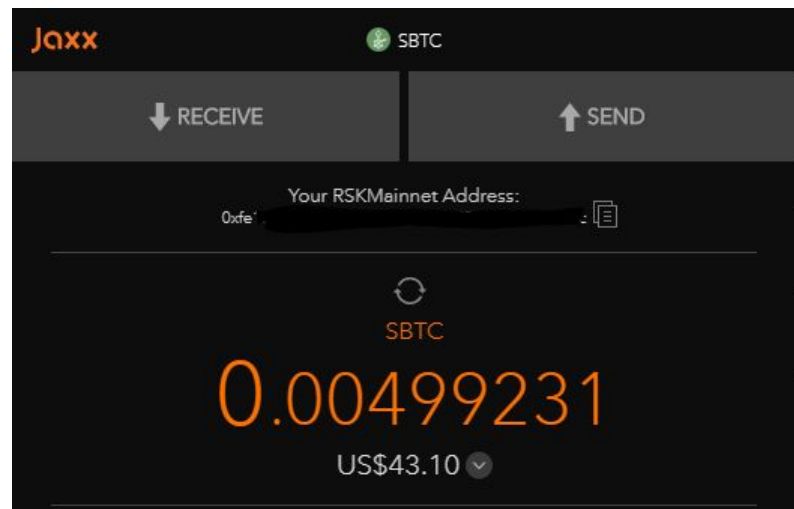
@RSKSMART  
#SMARTERBITCOIN

Smart contracts powered by the Bitcoin Network.

# Descarga tu propia Billetera !



# Jaxx





# RSK: Contratos inteligentes sobre Bitcoin

# INTERNET DEL VALOR



@RSKSMART  
#SMARTERBITCOIN

Smart contracts powered by the Bitcoin Network.



# INTERNET DEL VALOR



@RSKSMART  
#SMARTERBITCOIN

Smart contracts powered by the Bitcoin Network.

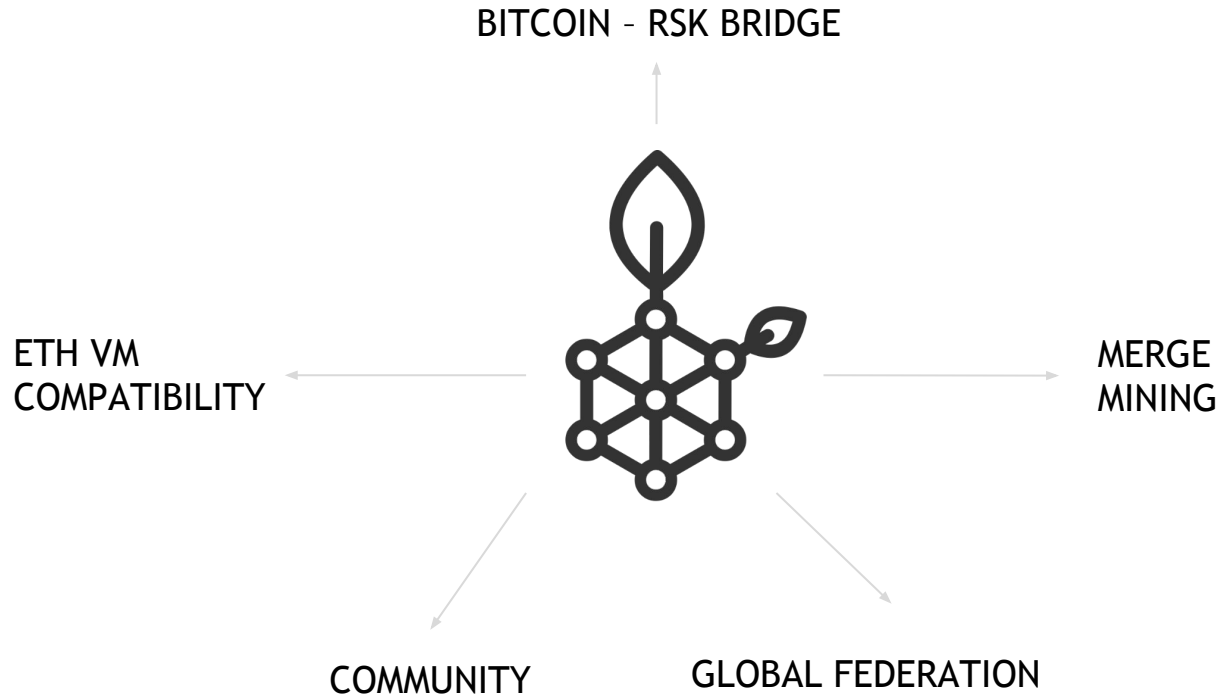


## ¿Por qué Bitcoin?

- 1 Bitcoin es la red blockchain descentralizada más segura y madura que existe.
- 2 RSK agrega la **funcionalidad** de contratos inteligentes a la red Bitcoin.
- 3 Bitcoin es la red con la comunidad más grande y el mayor network effect.



# Pilares



# ETH Compatible Virtual Machine



- TOTALMENTE COMPATIBLE CON LAS PLATAFORMAS LIDERES EN CONTRATOS INTELIGENTES
- BASADO EN LA VM DE ETHEREUM Y OPTIMIZADO CON MEJORAS EN LA SEGURIDAD
- NUEVOS OPCODES (E.G. DUPN, SWAPN)
- VERSIONADO DE CONTRATOS



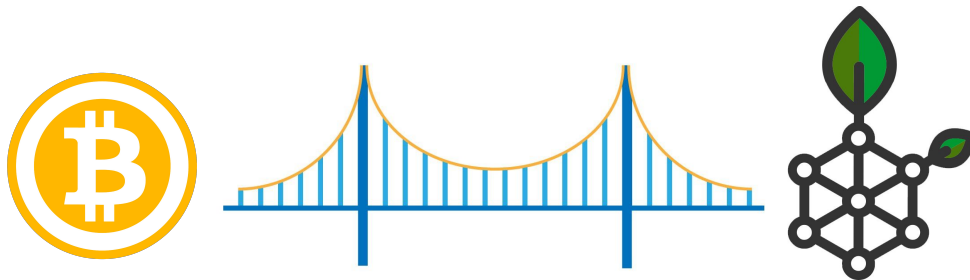
@RSKSMART  
#SMARTERBITCOIN

Smart contracts powered by the Bitcoin Network.

# Bitcoin-RSK Bridge



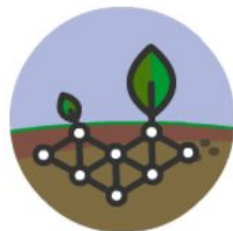
- CONTRATOS INTELIGENTES QUE PERMITEN A LA PLATAFORMA DE RSK A USAR BITCOIN COMO COMBUSTIBLE



# Merged Mining



- LA RED DE RSK ESTA ASEGURADA POR LOS MINEROS DE BITCOIN
- LOS MINEROS DE BITCOIN INCREMENTAN SUS INGRESOS SIN COSTO ADICIONAL
- RSK APROVECHA LA SEGURIDAD DE BITCOIN PARA CREAR LA PLATAFORMA DE CONTRATOS INTELIGENTES MAS SEGURA



# Global Federation



- MAS DE 20 DE LAS MAS RENOMBRADAS EMPRESAS DE BITCOIN DISTRIBUIDAS POR TODO EL MUNDO
- NIVEL EXTRA DE SEGURIDAD (E.G. CHECKPOINT SERVICES)
- SERVICIOS ADICIONALES (E.G. ORACLES)



# Community



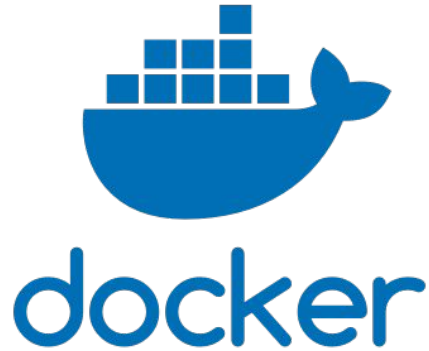
@RSKSMART  
#SMARTERBITCOIN





## Instalando un nodo

## Plataformas disponibles





# Correr un nodo de RSK usando Java

1. BAJAR E INSTALAR EL JDK DE JACA  
<https://java.com/download>
2. BAJAR LA ULTIMA VERSION JAR DEL NODO DE RSK  
<https://github.com/rsksmart/rskj/releases>
3. UBICAR EL JAR EN TU CARPETA DE TRABAJO
4. ABRIR UNA CONSOLA EN ESA CARPETA
5. EJECUTAR EL NODO CON ESTE COMANDO

```
java -cp rskj-core-0.5.3-ORCHID-all.jar -Dlogback.configurationFile=./logback.xml  
co.rsk.Start --regtest --reset
```



# Interactuar con el nodo de RSK

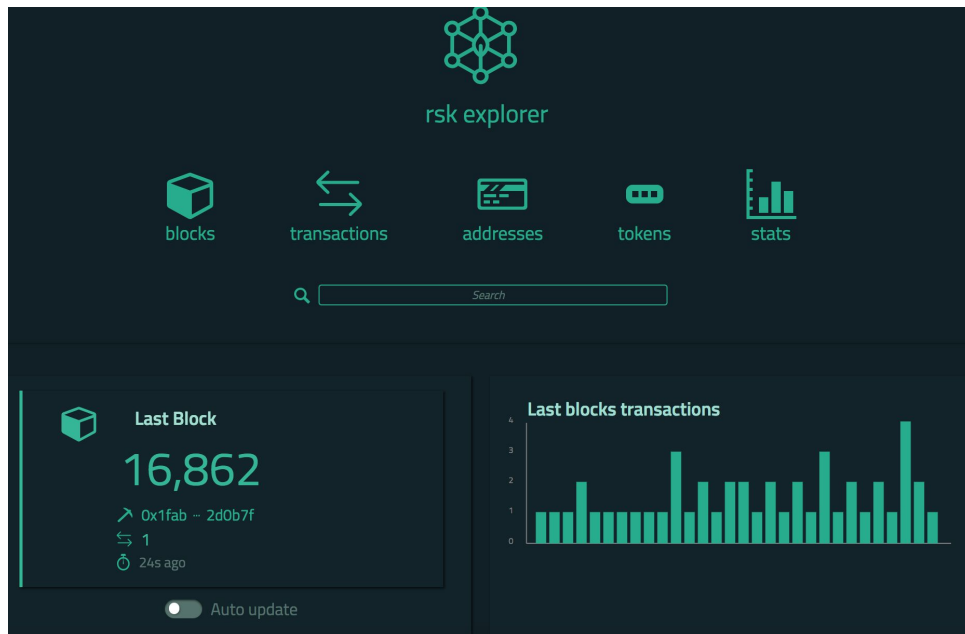
## 1. RUN TRUFFLE

```
truffle console
```

## 2. CHEQUEAR ULTIMO NUMERO DE BLOQUE

```
truffle(console)> web3.eth.blockNumber
```

# RSK Explorer



<https://explorer.testnet.rsk.co/>

<https://explorer.rsk.co/>

@RSKSMART  
#SMARTERBITCOIN

Smart contracts powered by the Bitcoin Network.

# RSK Network Status



<https://stats.testnet.rsk.co/>

<https://stats.rsk.co/>

@RSKSMART  
#SMARTERBITCOIN

Smart contracts powered by the Bitcoin Network.

# RSK Public Nodes



<https://public-node.testnet.rsk.co>

<https://public-node.rsk.co>



Vamos a decir hola !





# Crear el proyecto

- *truffle init*

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\temp\demo\1> truffle init
Downloading...
Unpacking...
Setting up...
Unbox successful. Sweet!

Commands:

  Compile:      truffle compile
  Migrate:      truffle migrate
  Test contracts: truffle test
PS C:\temp\demo\1> █
```



# Escribir contrato

```
pragma solidity ^0.4.4;

contract HelloWorld {
    string greeting;

    constructor() public {
        greeting = "Hello smart world!";
    }

    function getGreeting () public view returns (string) {
        return greeting;
    }

    function setGreeting(string _newGreeting) public returns (bool success) {
        greeting = _newGreeting;
        return true;
    }
}
```



# Actualizar Migrations

The screenshot shows the Visual Studio Code interface with the file explorer on the left and the editor on the right. The file explorer shows a project structure with folders 'contracts' and 'migrations', and files 'HelloWorld.sol', 'Migrations.sol', '1\_initial\_migration.js', 'truffle-config.js', and 'truffle.js'. The editor displays the content of '1\_initial\_migration.js'.

```
1 var Migrations = artifacts.require("./Migrations.sol");
2 var HelloWorld = artifacts.require("./HelloWorld.sol");
3
4 module.exports = function(deployer) {
5   deployer.deploy(Migrations);
6   deployer.deploy(HelloWorld);
7 };
8
```

# Compiler



## - *truffle compile*

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

```
PS C:\temp\demo\1> truffle init
Downloading...
Unpacking...
Setting up...
Unbox successful. Sweet!

Commands:

  Compile:      truffle compile
  Migrate:      truffle migrate
  Test contracts: truffle test
PS C:\temp\demo\1> truffle compile
Compiling .\contracts\HelloWorld.sol...
Compiling .\contracts\Migrations.sol...
Writing artifacts to .\build\contracts
```

# Run Migrations

- *truffle console*
- *migrate*

```
truffle(develop)> migrate --reset
Using network 'develop'.

Running migration: 1_initial_migration.js
  Deploying Migrations...
    ... 0xa800e9180f72b2d60bc874efad341ae1e8eb2e6dfc20e020cf118f09c4a5921b
  Migrations: 0x8cdaf0cd259887258bc13a92c0a6da92698644c0
  Deploying HelloWorld...
    ... 0xdb07975e86c975c77839ea930d809a6f947d0bf31e600c1702ac022f178163ae
  HelloWorld: 0xf12b5dd4ead5f743c6baa640b0216200e89b60da
Saving successful migration to network...
  ... 0x57b1b513cedfb4bd94fedf949f8c494435abb4e0b43a5b9fadeae23f5bc83256
Saving artifacts...
truffle(develop)> █
```



# Llamar el Contrato

```
var helloWorldContract  
HelloWorld.deployed().then(function(contract){ helloWorldContract = contract; })  
helloWorldContract.getGreeting.call().then(console.log)
```

```
truffle(develop)> var helloWorldContract;  
undefined  
truffle(develop)> HelloWorld.deployed().then(function(contract){ helloWorldContract = contract; });  
undefined  
truffle(develop)> helloWorldContract.getGreeting.call().then(console.log)  
Hello smart world!  
undefined  
truffle(develop)> █
```

[illegible]

# Referencias

[https://www.youtube.com/playlist?list=PLTS-iVn0k2\\_8CYAmEGfZQ2vyrwzjDiBvi](https://www.youtube.com/playlist?list=PLTS-iVn0k2_8CYAmEGfZQ2vyrwzjDiBvi)

<https://www.youtube.com/watch?v=1jdYVw8zLUg>

<https://github.com/rsksmart/rskj/wiki>





# Conectándonos

@RSKSMART  
#SMARTERBITCOIN

Smart contracts powered by the Bitcoin Network.



# Como entrar en este mundo

- 1 CAPACITATE: [educate.rsk.co](https://educate.rsk.co)
- 2 COMUNIDAD DE DESARROLLADORES: [ambassadors.rsk.co](https://ambassadors.rsk.co)
- 3 DESARROLLÁ TU PROPIA DAPP: [rsk.co/#getting-started](https://rsk.co/#getting-started)
- 4 GITHUB: [github.com/rksmart](https://github.com/rksmart)



# Canales de comunicación

- Telegram: @rsksmartcontracts
- Twitter: @rsksmart
- Slack: <http://bit.ly/RSKCommunitySlack>
- Blog: media.rsk.co



## Veamos casos de uso



@RSKSMART  
#SMARTERBITCOIN



La tecnología está, lo que faltan son los  
desarrolladores

# Sistema financiero 2.0



GLOBAL  
POPULATION

**7,6 B**



UNBANKED &  
UNDERBANKED

**2,2 B**



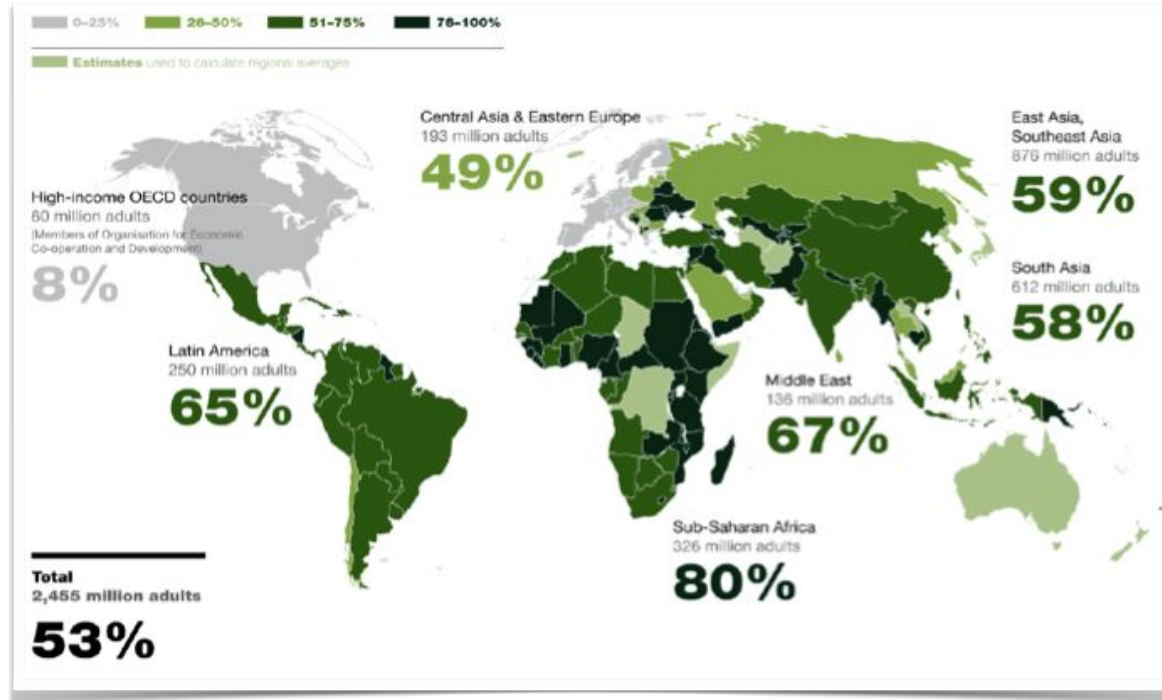
MOBILE USERS BY  
2025

**5,1 B**

@RSKSMART  
#SMARTERBITCOIN

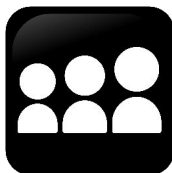
Smart contracts powered by the Bitcoin Network.

# Unbanked Population Around the Globe





¡Sólo está ocurriendo una vez más!



YAHOO!



Google

Music

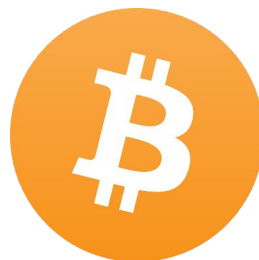
Video

Photo sharing

Mobility

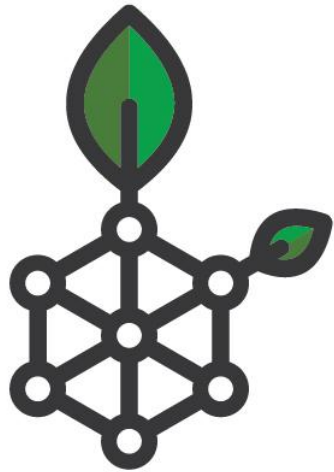
Lodging

Web services



Servicios financieros  
& Contratos  
inteligentes

@RSKSMART  
#SMARTERBITCOIN



**RSK**  
SMARTER BITCOIN

# Gracias!

Sebastián Pérez  
RSK Community Tech Evangelist  
sebastian@rsk.co - @sebaleoperez

@RSKSMART  
#SMARTERBITCOIN