Simple Multisig with Hardware Wallets

👤 Riccardo Casatta

👤 Valerio Vaccaro

Plan ₿ Forum

📆 2025-10-25

👤 Riccardo Casatta 👀

- 💻 Bitcoin Developer and Software Engineer
- 🦀 Rust enthusiast and maintainer of several Bitcoin projects
- ₿ Bitcoin and Liquid Engineer at Blockstream

👤 https://x.com/RCasatta

🐙 https://github.com/RCasatta

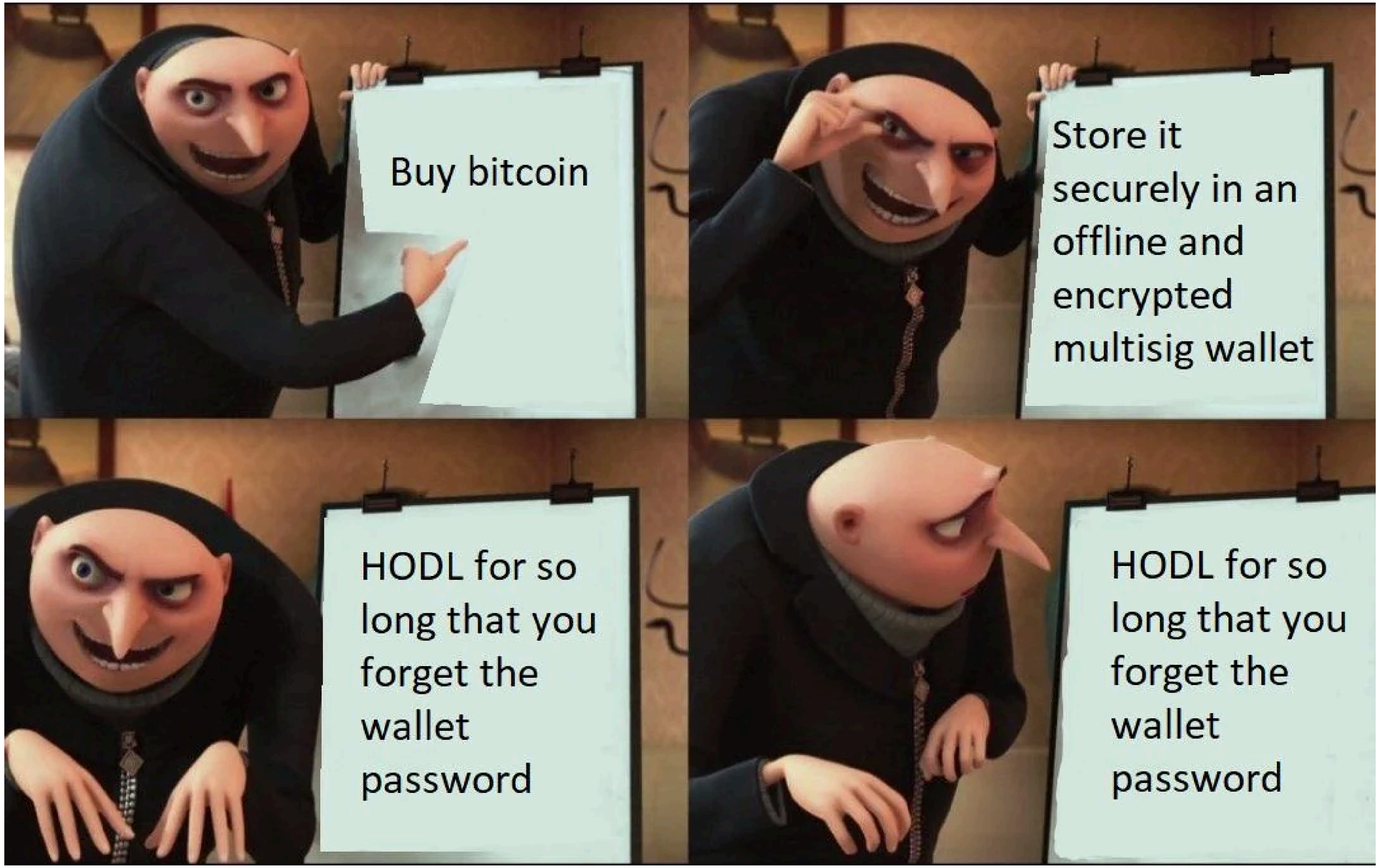
👤 Valerio Vaccaro 👀

- 💻 Bitcoin Developer and Hardware Expert
- 🔥 Contributor to Open Source Bitcoin Projects
- ⚠️ Passionate about DIY Hardware
- ₿ Bitcoin and Liquid Engineer at Blockstream

👤 https://www.linkedin.com/in/valeriovaccaro/

🐙 https://github.com/valerio-vaccaro/

Meme


Buy bitcoin
Store it securely in an offline and encrypted multisig wallet
HODL for so long that you forget the wallet password
HODL for so long that you forget the wallet password

License

May this presentation inspire you to become more self-sovereign!

Summary

🗂️ Agenda

- 🔑 What is a Multisig?
- 🔒 What is a Hardware Wallet?
- 🛠️ Preparing Our Hardware Wallets
- 🖥️ Creating a Multisig on Sparrow
- 💸 Receiving and Spending Funds
- 👀 Q&A

🔑 What is a Multisig?

- **Multisig** stands for "multi-signature"
- A Bitcoin address that requires **two or more private keys** to approve and spend funds
- 💡 Example: "2-of-3 multisig" means any 2 out of 3 keys must sign a transaction
- **Used for:**
  - Improved security (even if one key is lost or stolen, funds are safe)
  - Shared custody (businesses, families, organizations)
  - Reducing single points of failure
- Multisig setups are flexible and can be tailored to your security needs!

🔒 What is a Hardware Wallet?

- **A dedicated device designed to securely store your Bitcoin private keys**
- **Signs transactions safely on-device:** your private keys never leave the hardware, and you can always review what you are signing
- Allows you to generate **public keys** and Bitcoin **addresses**
- Supports creating recovery phrases (mnemonics) with the option of extra security using external entropy
- Can be connected to a computer or smartphone, but the secrets are never exposed
- Adds an extra layer of security and control to your funds
- Protects against malware, remote attacks, and phishing attempts
- 🔥 Makes self-custody of Bitcoin both **practical and secure**

🔒 What is NOT a Hardware Wallet?

- **Not a backup solution** for mnemonics; you must handle backups yourself
- **Not a transaction creator**; you use a software wallet for that
- **Not a portfolio management tool**; it does NOT calculate balances or track transaction history—this is done by your wallet software
- ⚠️ You can generate mnemonics on-device but ...

🔒 Examples

We will use three different hardware wallets, each from a different manufacturer, with:

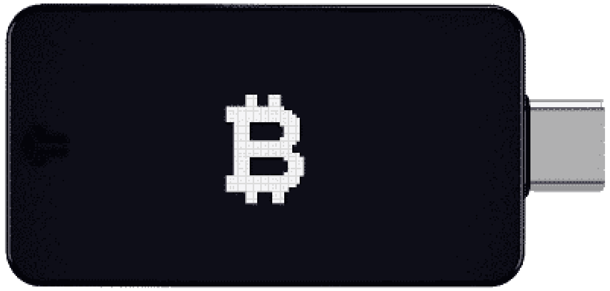
- Three separate vendors
- Three distinct approaches to mnemonic backup and storage
- Three hardware implementations (different architectures and manufacturers)

Examples: Jade

- Open-source hardware wallet developed by Blockstream
- Supports Bitcoin and the Liquid Network
- USB-C and Bluetooth compatible
- Large color screen, QR code support
- Designed for privacy and air-gapped operation
- Extensively documented DIY build process

🔒 Examples: BitBox02

- Open-source hardware wallet by Shift Crypto
- Focused on Bitcoin and security best practices
- Touch sliders for PIN and navigation
- MCU and secure chip architecture (with interface and code fully open)
- MicroSD backup



🔒 Examples: SeedSigner

- DIY, fully open-source Bitcoin hardware wallet
- Uses standard off-the-shelf parts (Raspberry Pi Zero, camera, screen)
- No specialized secure chip; stateless design —no secrets stored on device
- Camera-based QR code signing
- Targets maximum transparency and low-cost, accessible hardware
- Perfect for air-gapped cold storage and multisig setups

🛠️ Preparing Our Hardware Wallets

The first step is to update the firmware, which can usually be done using the companion app or the manufacturer's website.

Next, generate or restore a mnemonic directly on the device and complete the basic configuration.

Below is a quick summary of the initialization process for all three hardware wallets.

🛠️ Preparing Our Hardware Wallets - Jade

To initialize your Blockstream Jade hardware wallet:

1. **Connect the Jade** to your computer using USB-C or turn it on wirelessly.
2. **Update the firmware** using the official Blockstream Green app or the Blockstream Jade web setup page.
3. **Create a new wallet or restore from backup**: Choose "Create wallet" for a new mnemonic, or "Restore wallet" if you already have a seed phrase.
4. **Follow the on-screen instructions** and carefully write down or verify your 12 or 24-word recovery phrase (mnemonic).
5. **Set a device PIN** for mnemonic encryption; connecting to a compatible software wallet like Blockstream Green may be required.
6. **Back up your wallet**

🛠️ Preparing Our Hardware Wallets - BitBox

To initialize your BitBox02 hardware wallet:

1. **Connect the BitBox02** to your computer and download the official BitBoxApp.
2. **Install and launch BitBoxApp**. The app will automatically detect your BitBox and check for firmware updates.
3. **Create a new wallet or restore from a backup**: choose "Create wallet" for a new setup, or "Restore from backup" using your microSD card backup.
4. **Follow the on-screen instructions** to generate and confirm your recovery words (mnemonic).
5. **Set up a device password** for extra security.
6. **Back up your wallet**: The BitBox02 will prompt you to insert a microSD card to automatically save an encrypted backup.

🛠️ Preparing Our Hardware Wallets - SeedSigner

To initialize your SeedSigner device:

1. **Assemble and power up your SeedSigner.**
2. **Flash the SeedSigner OS** by flashing SeedSigner releases on SD card.
3. **Set to Testnet (for this example)**: from the main menu, go to "Settings" → "Select Network" → choose "Testnet" or "Signet" for safer experimentation.
4. **Generate or Import a Seed on SeedSigner**:
    - Select “Seed Tools” then “Create Seed” to make a new seed phrase (mnemonic). Write down and verify all 12 or 24 words carefully.
    - Alternatively, choose “Scan Seed QR” if restoring from a QR code backup you created earlier.

⚠️ You will need to re-enter (scan or type) your seed each time you sign a transaction.

💻 Creating a Multisig on Sparrow

Sparrow Wallet is a powerful Bitcoin wallet designed for desktop use. It is ideal for Bitcoiners who value privacy, security, and versatility:

- **Open Source & Focused on Self-sovereignty**
- **Supports Airgapped Hardware Wallets**: including DIY devices like Jade, Specter, and Passport
- **Advanced Features**: multisig wallets, coin control, custom scripts, PSBT (Partially Signed Bitcoin Transaction) workflow
- **Works on Testnet, Signet, and Mainnet**
- **Great Interface**: intuitive UI for managing addresses, UTXOs, and coin selection

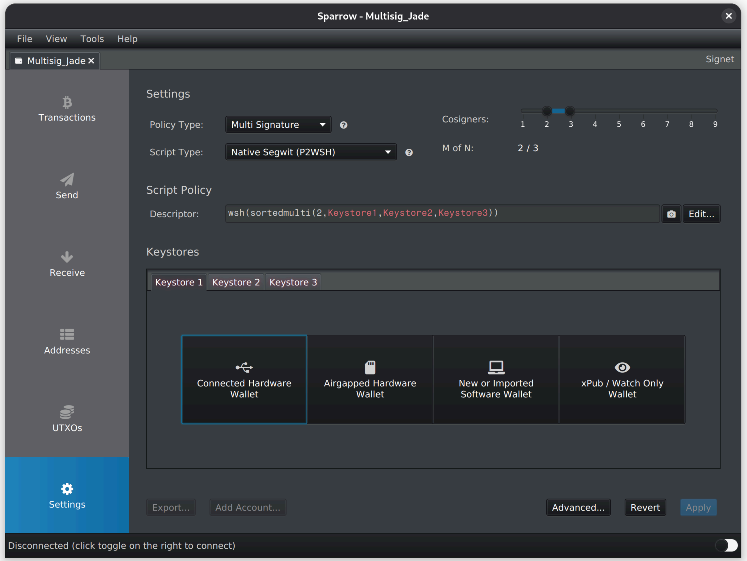
💻 Creating a Multisig on Sparrow

Signet is a public Bitcoin test network, designed for safe experimentation and development, without risking real bitcoin:

- **"Fake bitcoin"** is used on signet—no real value, free to obtain
- **Safer for Testing**: unlike testnet, blocks on signet are signed and reliable, reducing spam and instability
- **Similar Features to Mainnet**: allows you to experiment with real Bitcoin software and devices, simulating mainnet scenarios
- **Perfect for wallet development, testing firmware, or playing with new tools**

💻 Creating a Multisig on Sparrow -

Create a multisig 2of3 native segwit wallet



Sparrow - Multisig_Jade

Settings

Policy Type: Multi Signature

Script Type: Native Segwit (P2WSH)

Cosigners: 1 2 3 4 5 6 7 8 9

M of N: 2 / 3

Script Policy

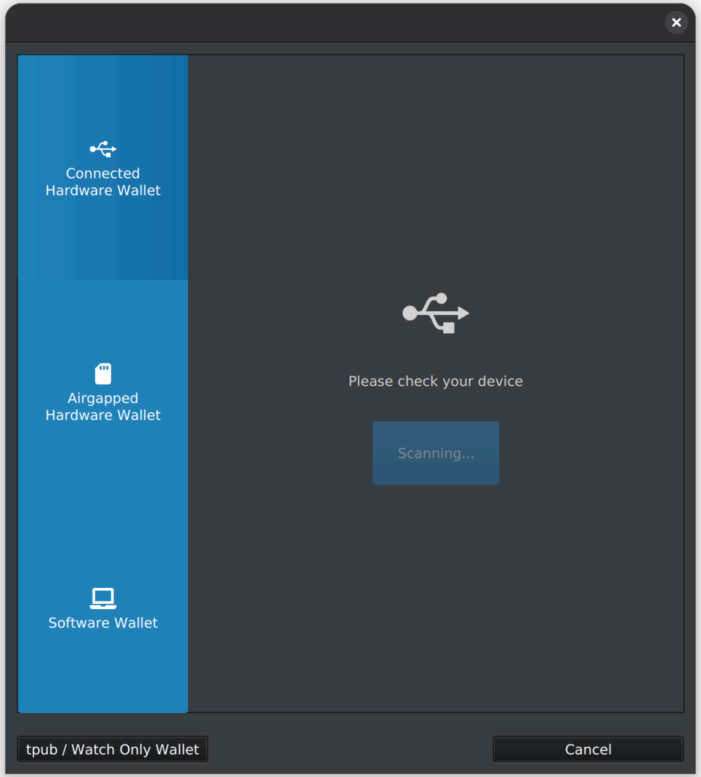
Descriptor: `wsh(sortedmulti(2,Keystore1,Keystore2,Keystore3))`

Keystores

Disconnected (click toggle on the right to connect)

Creating a Multisig on Sparrow -

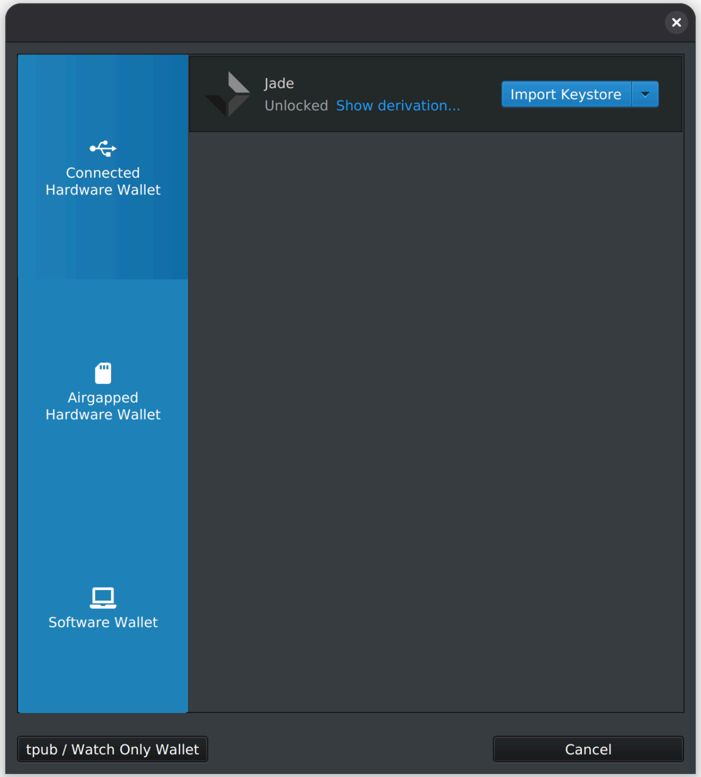
Load Jade as keystore 1.



Please check your device

Creating a Multisig on Sparrow -

Jade found! We can import the key.


Jade
Unlocked Show derivation...
Import Keystore
Connected Hardware Wallet
Airgapped Hardware Wallet
Software Wallet
tpub / Watch Only Wallet
Cancel

💻 Creating a Multisig on Sparrow - Jade

For the other two insert these data:

BitBox

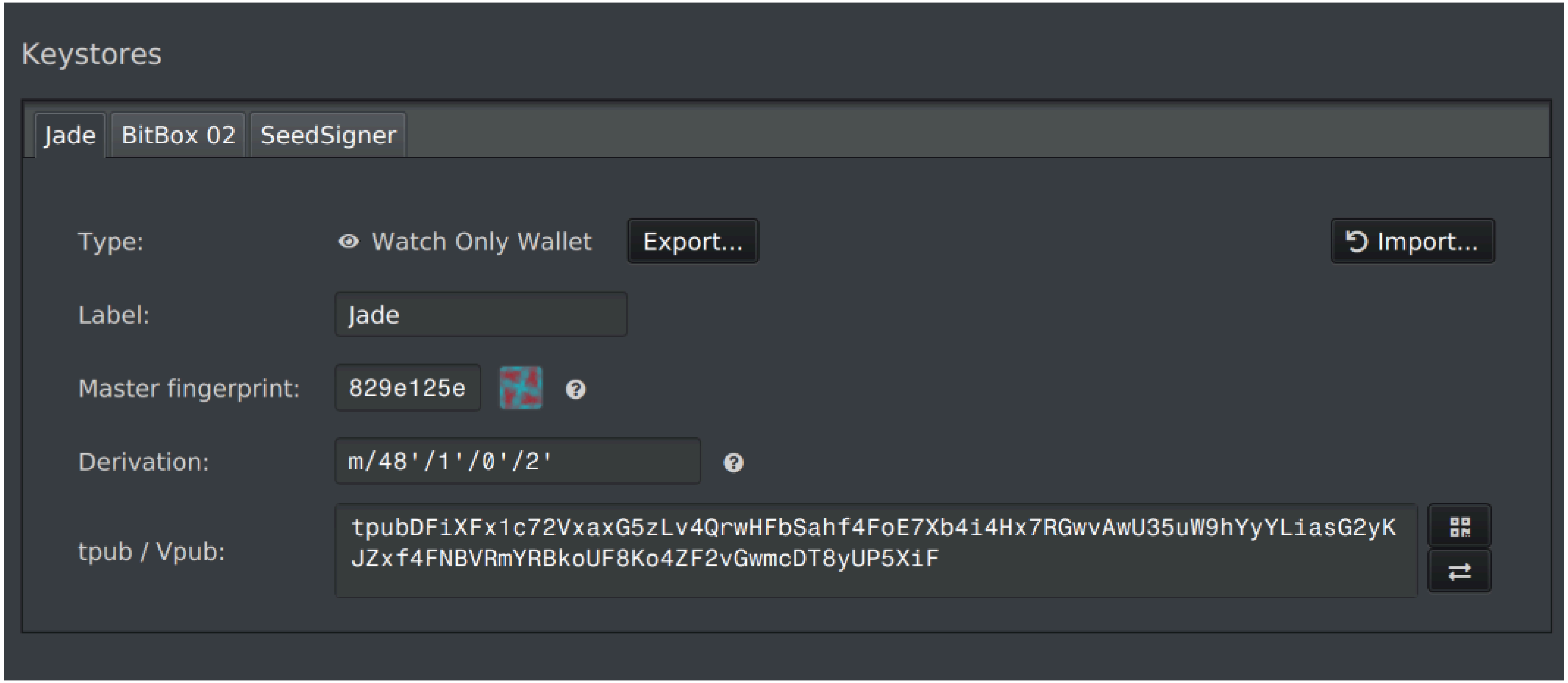
[7edda1f6/48'/1'/0'/2']tpubDEEGMxEq1otUaBFWtwLSnn3k7nZyWbfThJWqs877VQdc8dTYwTo8JmUPpWfSUShfeAsJZBXHmvzJVdNqxTAbQnFwq54AeVNnDy2YkuLuGFK

SeedSigner

[d7efaa7e/48'/1'/0'/2']tpubDEyr8wUpFxYjuDUpKvBT75cut4ZNp1ixS4RkMBxX77dJK9XrKphoeX29aX5C1tPMcWESup7DSq1JwAaaySuQCTNud8mk8HWifqoobWtUtdU

Creating a Multisig on Sparrow - Jade

You will receive these data from other partecipants after they load the hardware wallet in sparrow.



Keystores

Type: Watch Only Wallet

Label: Jade

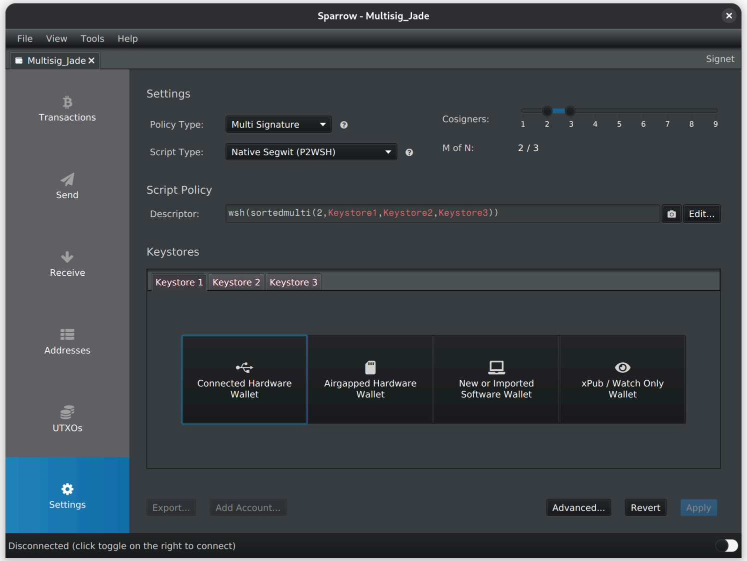
Master fingerprint: 829e125e

Derivation: m/48'/1'/0'/2'

tpub / Vpub: tpubDFiXFx1c72VxaxG5zLv4QrwHFbSahf4FoE7Xb4i4Hx7RGwvAwU35uW9hYyYLiasG2yKJZxf4FNBVRmYRBkoUF8Ko4ZF2vGwmcDT8yUP5XiF

Creating a Multisig on Sparrow -

Create a multisig 2of3 native segwit wallet



Settings

Policy Type: Multi Signature

Script Type: Native Segwit (P2WSH)

M of N: 2 / 3

Script Policy

Descriptor:

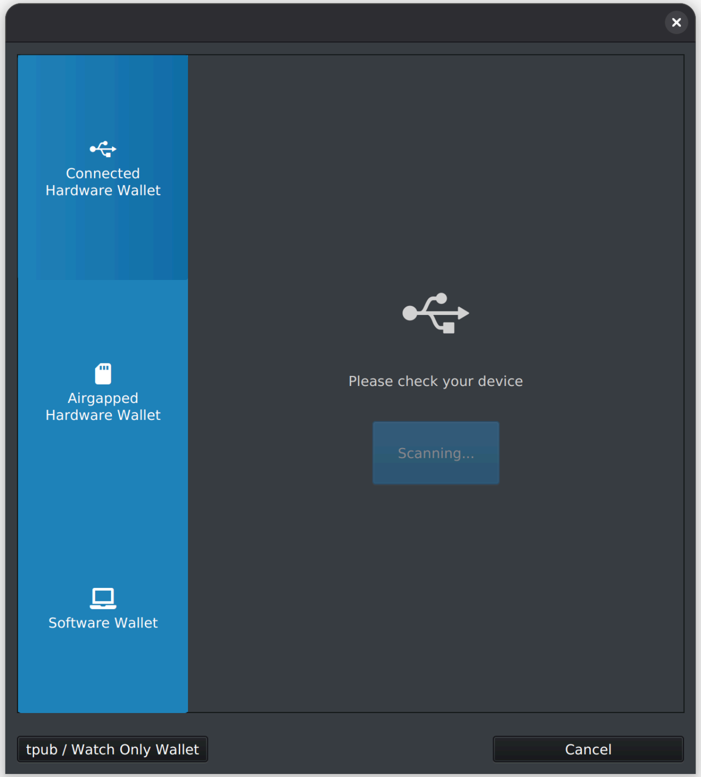
```
wsh(sortedmulti(2,Keystore1,Keystore2,Keystore3))
```

Keystores

Disconnected (click toggle on the right to connect)

💻 Creating a Multisig on Sparrow -

Load BitBox as keystore 2.



Connected Hardware Wallet

Airgapped Hardware Wallet

Software Wallet

Please check your device

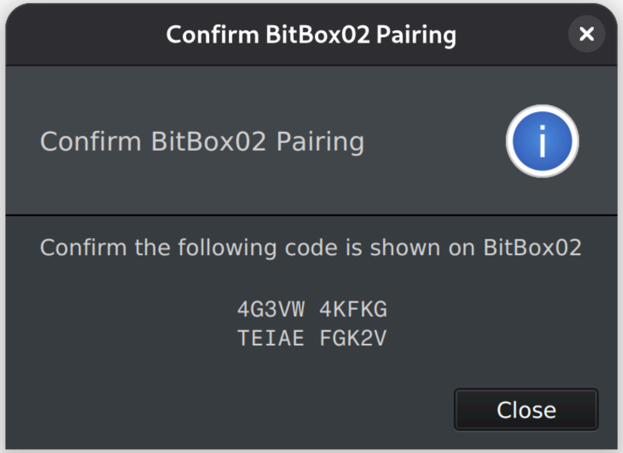
Scanning...

tpub / Watch Only Wallet

Cancel

💻 Creating a Multisig on Sparrow -

If needed confirm the pairing.



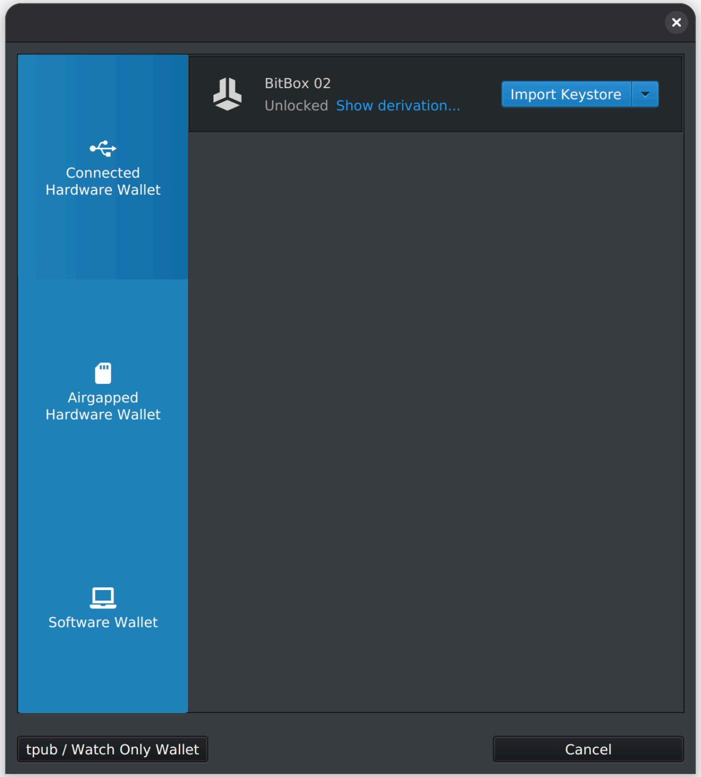
Confirm BitBox02 Pairing

Confirm the following code is shown on BitBox02

```
4G3VW 4KFKG
TEIAE FGK2V
```

Creating a Multisig on Sparrow -

BitBox found! We can import the key.



Creating a Multisig on Sparrow - BitBox

For the other two insert these data.

Jade

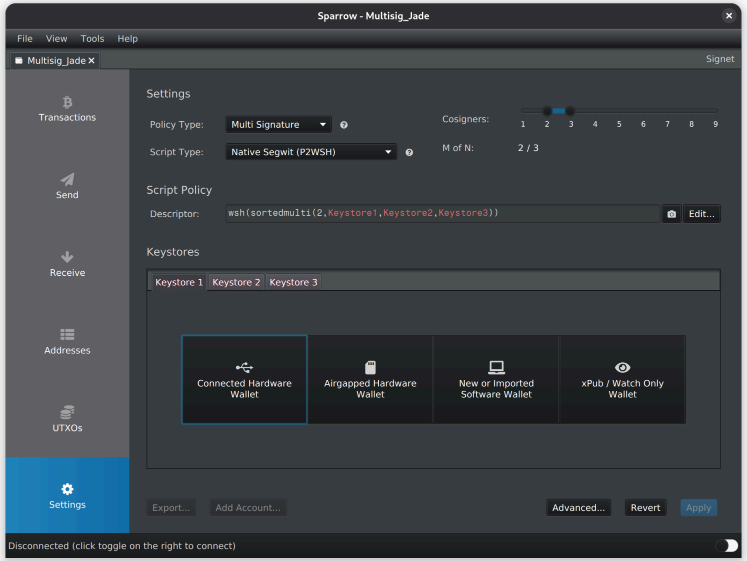
[829e125e/48'/1'/0'/2']tpubDFiXFx1c72VxaxG5zLv4QrwHFbSahf4FoE7Xb4i4Hx7RGwvAwU35uW9hYyYLiasG2yKJZxf4FNBVRmYRBkoUF8Ko4ZF2vGwmcDT8yUP5XiF

SeedSigner

[d7efaa7e/48'/1'/0'/2']tpubDEyr8wUpFxYjuDUpKvBT75cut4ZNp1ixS4RkMBxX77dJK9XrKphoeX29aX5C1tPMcWESup7DSq1JwAaaySuQCTNud8mk8HWifqoobWtUtdU

💻 Creating a Multisig on Sparrow -

Create a multisig 2of3 native segwit wallet



Sparrow - Multisig_Jade

Settings

Policy Type: Multi Signature

Script Type: Native Segwit (P2WSH)

Cosigners: 1 2 3 4 5 6 7 8 9

M of N: 2 / 3

Script Policy

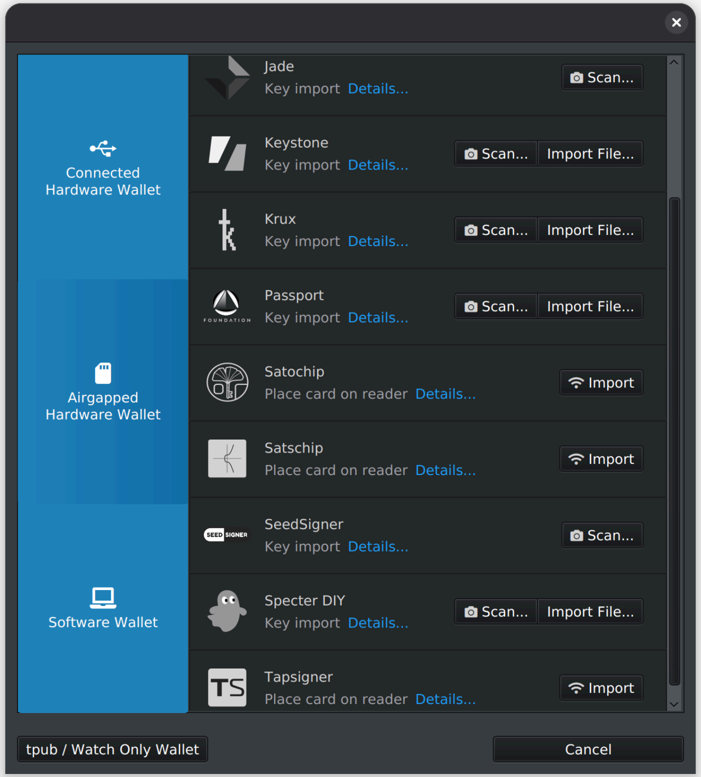
Descriptor:
```
wsh(sortedmulti(2,Keystore1,Keystore2,Keystore3))
```

Keystores

Disconnected (click toggle on the right to connect)

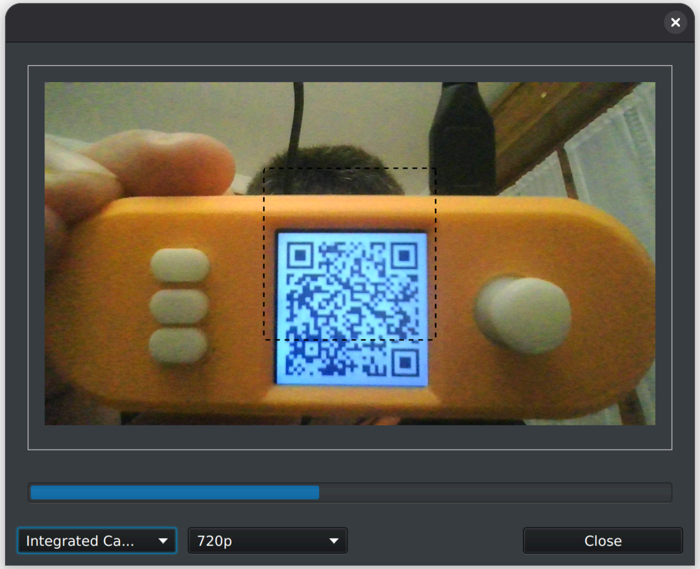
Creating a Multisig on Sparrow -

Load Seed Signer as keystore 3.



Creating a Multisig on Sparrow -

Load mnemonic and export XPub (you will need to select "multisig" and after "sparrow" as multisig type)



💻 Creating a Multisig on Sparrow - Seed Signer

For the other two insert these data.

Jade

[829e125e/48'/1'/0'/2']tpubDFiXFx1c72VxaxG5zLv4QrwHFbSahf4FoE7Xb4i4Hx7RGwvAwU35uW9hYyYLiasG2yKJZxf4FNBVRmYRBkoUF8Ko4ZF2vGwmcDT8yUP5XiF

BitBox

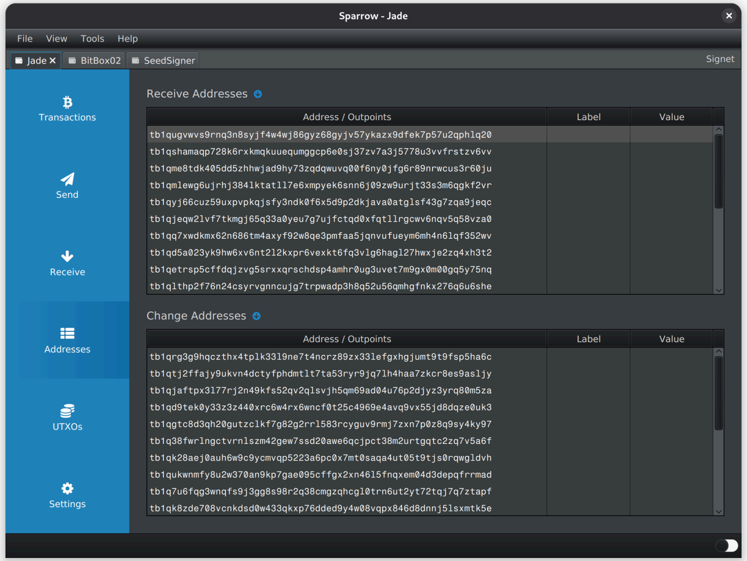
[7edda1f6/48'/1'/0'/2']tpubDEEGMxEq1otUaBFWtwLSnn3k7nZyWbfThJWqs877VQdc8dTYwTo8JmUPpWfSUShfeAsJZBXHmvzJVdNqxTAbQnFwq54AeVNnDy2YkuLuGFK

💻 Creating a Multisig on Sparrow

The created wallet can be saved as a descriptor, in my case this is the descriptor

```
wsh(
  sortedmulti(2,
    [d7efaa7e/48h/1h/0h2h]
         tpubDEyr8wUpFxYjuDUpKvBT75cut4ZNp1ixS4RkMBxX77dJK9XrKphoeX29aX5C1tPMcWESup7DSq1JwAaaySuQCTNud8mk8HWifqoobWtUtdU/<0;1>/*,
    [7edda1f6/48h/1h/0h/2h]
        tpubDEEGMxEq1otUaBFWtwLSnn3k7nZyWbfThJWqs877VQdc8dTYwTo8JmUPpWfSUShfeAsJZBXHmvzJVdNqxTAbQnFwq54AeVNnDy2YkuLuGFK/<0;1>/*,
    [829e125e/48h/1h/0h/2h]
        tpubDFiXFx1c72VxaxG5zLv4QrwHFbSahf4FoE7Xb4i4Hx7RGwvAwU35uW9hYyYLiasG2yKJZxf4FNBVRmYRBkoUF8Ko4ZF2vGwmcDT8yUP5XiF/<0;1>/*
  )
)#xvr0j9ca
```

⚠️ Remember the importance of backing up the descriptor in a multisig scenario, without it, you can't sign even if you have 2 out of 3 private keys!



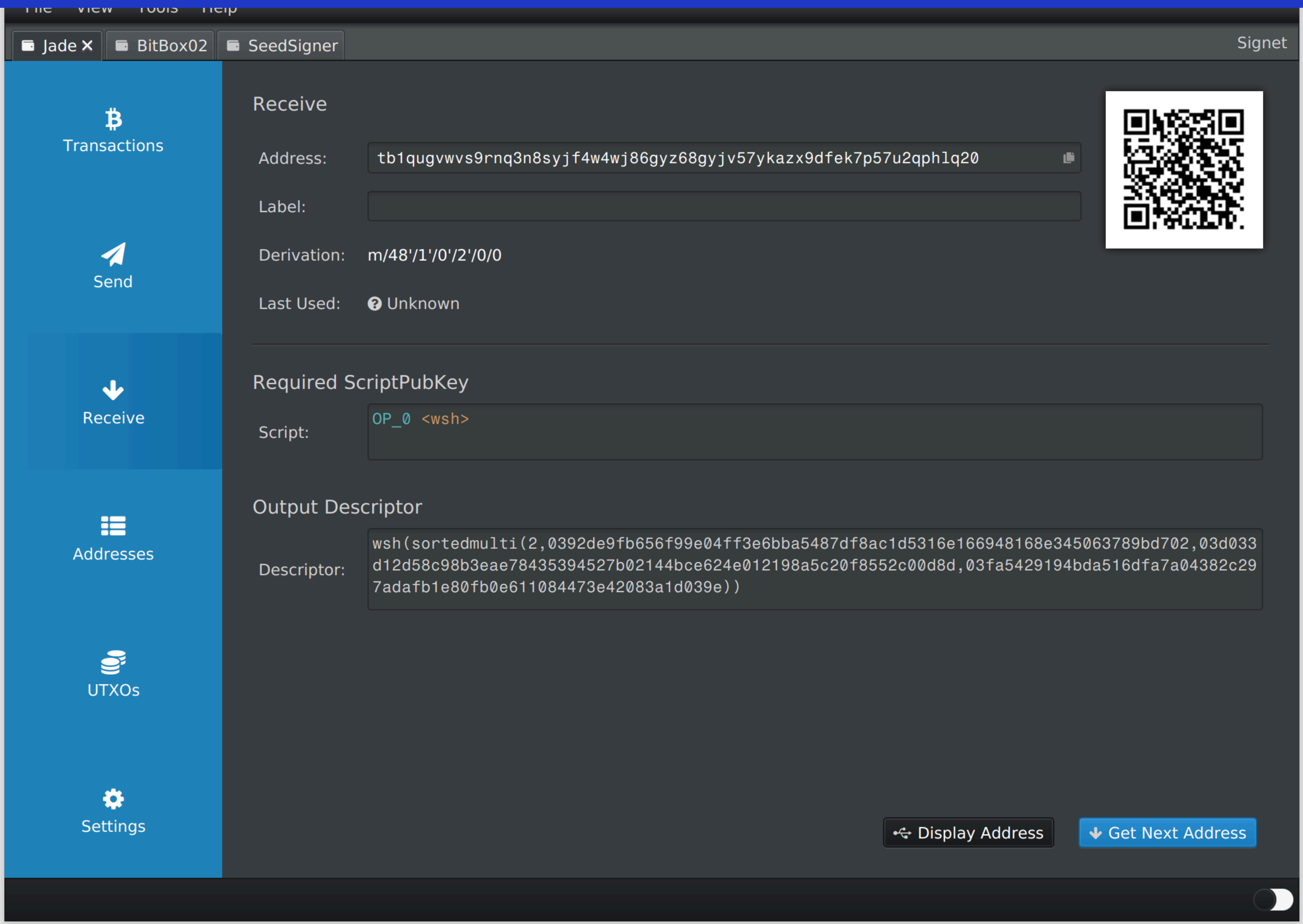
Receive Addresses

Address / Outpoints	Label	Value
tb1qugvwvs9rnq3n8syjf4w4wj86gyz68gyjv57ykazx9dfek7p57u2qphlq20		
tb1qshamaqp728k6rxkmqkuuequmggcp6e0sj37zv7a3j5778u3vvfrstzv6vv		
tb1qme8tdk405dd5zhhwjad9hy73zqdqwuvq00f6ny0jfg6r89nrwcus3r60ju		
tb1qmlewg6ujrhj384lktatll7e6xmpyek6snn6j09zw9urjt33s3m6qgkf2vr		
tb1qyj66cuz59uxpvpkqjsfy3ndk0f6x5d9p2dkjava0atglsf43g7zqa9jeqc		
tb1qjeqw2lvf7tkmgj65q33a0yeu7g7ujfctqd0xfqtllrgcwv6nqv5q58vza0		
tb1qq7xwdkmx62n686tm4axyf92w8qe3pmfaa5jqnvufueym6mh4n6lqf352wv		
tb1qd5a023yk9hw6xv6nt2l2kxpr6vexkt6fq3vlg6hagl27hwxje2zq4xh3t2		
tb1qetrsp5cffdqjzvg5srxxqrschdsp4amhr0ug3uvet7m9gx0m00gq5y75nq		
tb1qlthp2f76n24csyrvgnncujg7trpwadp3h8q52u56qmhgfnkx276q6u6she		

Change Addresses

Address / Outpoints	Label	Value
tb1qrg3g9hqczthx4tplk33l9ne7t4ncrz89zx33lefgxhgjumt9t9fsp5ha6c		
tb1qtj2ffajy9ukvn4dctyfphdmtlt7ta53ryr9jq7lh4haa7zkcr8es9asljy		
tb1qjaftpx3l77rj2n49kfs52qv2qlsvjh5qm69ad04u76p2djyz3yrq80m5za		
tb1qd9tek0y33z3z440xrc6w4rx6wncf0t25c4969e4avq9vx55jd8dqze0uk3		
tb1qgtc8d3qh20gutzclkf7g82g2rrl583rcyguv9rmj7zxn7p0z8q9sy4ky97		
tb1q38fwrlngctvrnlszm42gew7ssd20awe6qcjpct38m2urtgqtc2zq7v5a6f		
tb1qk28aej0auh6w9c9ycmvqp5223a6pc0x7mt0saqa4ut05t9tjs0rqwgldvh		
tb1qukwnmfy8u2w370an9kp7gae095cffgx2xn46l5fnqxem04d3depqfrrmad		
tb1q7u6fqg3wnqfs9j3gg8s98r2q38cmgzqhcgl0trn6ut2yt72tqj7q7ztapf		
tb1qk8zde708vcnkdsd0w433qkxp76dded9y4w08vqpx846d8dnnj5lsxmtk5e		

Receiving and Spending Funds



Receive

Address: tb1qugvwvs9rnq3n8syjf4w4wj86gyz68gyjv57ykazx9dfek7p57u2qphlq20

Label:

Derivation: m/48'/1'/0'/2'/0/0

Last Used: Unknown

Required ScriptPubKey

Script:

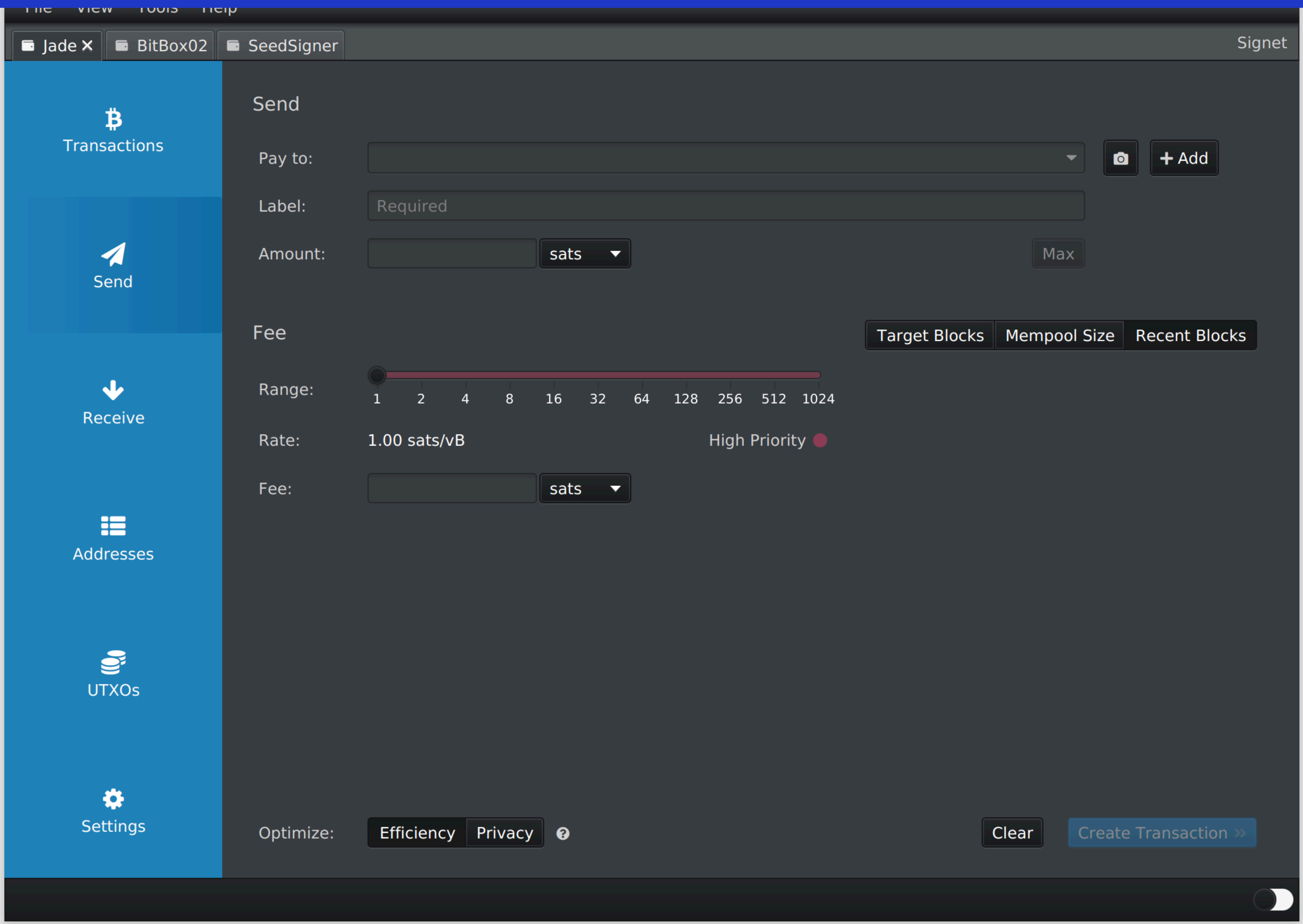
```
OP_0 <wsh>
```

Output Descriptor

Descriptor:

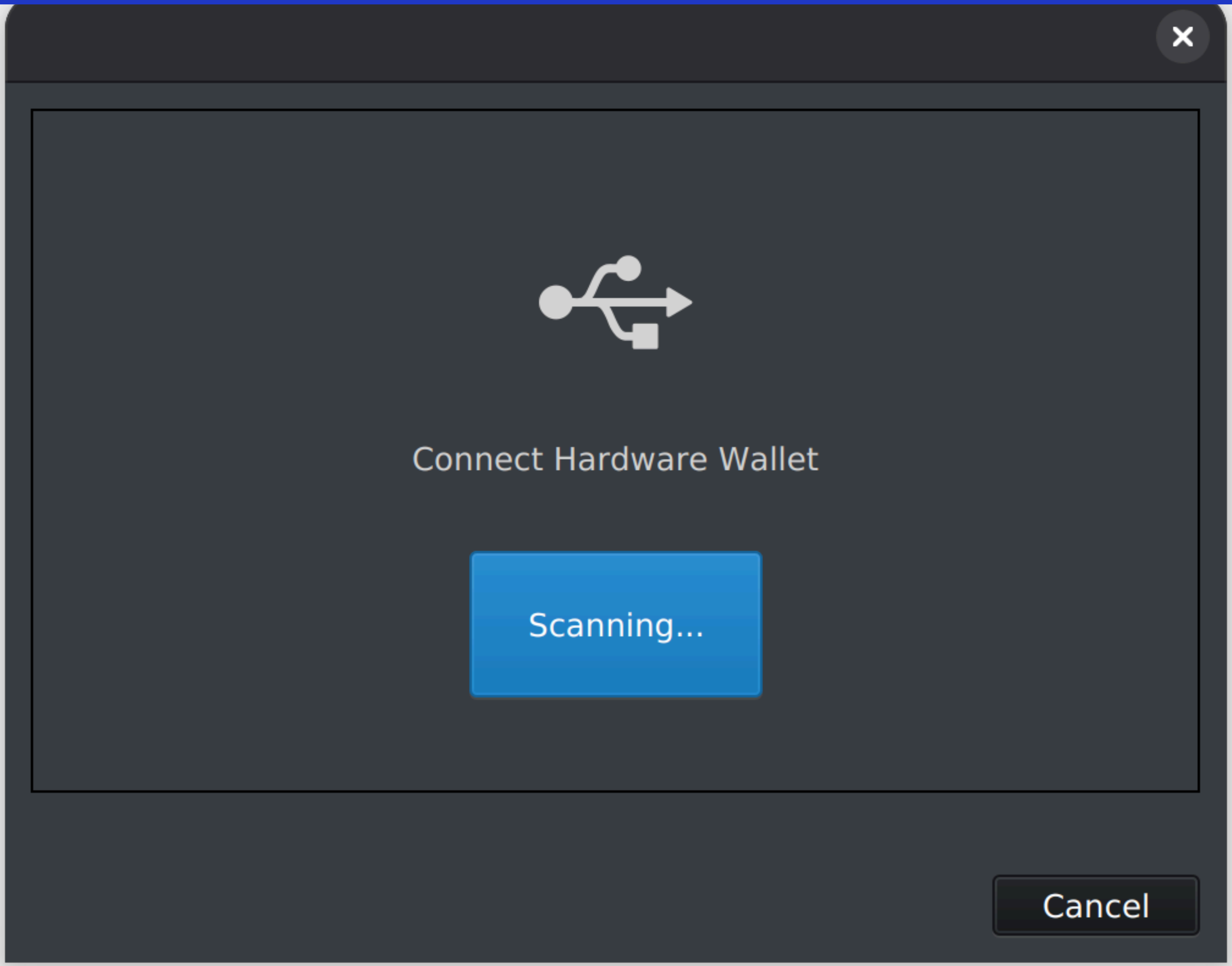
```
wsh(sortedmulti(2,0392de9fb656f99e04ff3e6bba5487df8ac1d5316e166948168e345063789bd702,03d033
d12d58c98b3eae78435394527b02144bce624e012198a5c20f8552c00d8d,03fa5429194bda516dfa7a04382c29
7adafb1e80fb0e611084473e42083a1d039e))
```

Receiving and Spending Funds



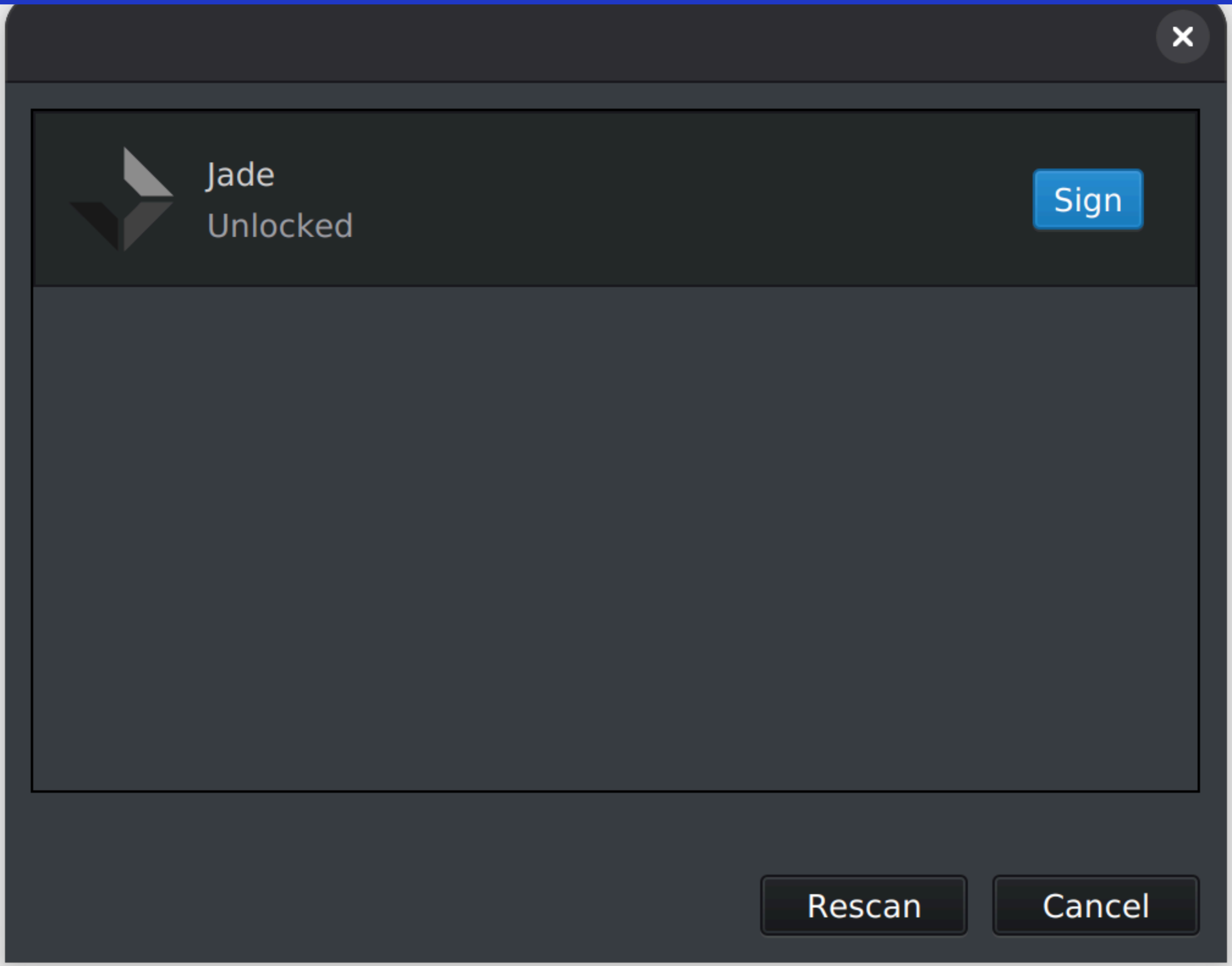
Rate: 1.00 sats/vB

💸 Receiving and Spending Funds - Jade

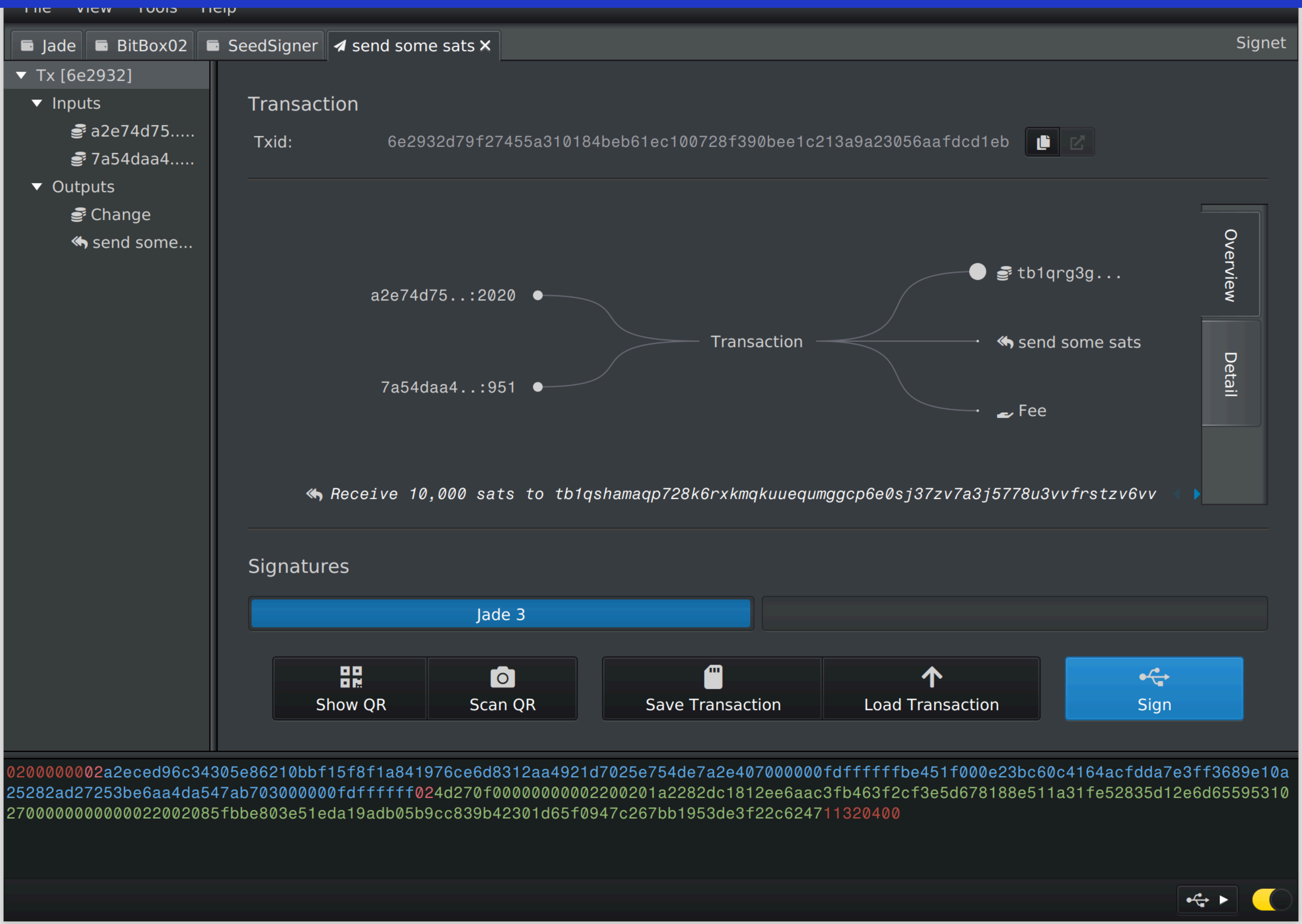


Connect Hardware Wallet

💸 Receiving and Spending Funds - Jade



Receiving and Spending Funds - Jade



Transaction

Txid: 6e2932d79f27455a310184beb61ec100728f390bee1c213a9a23056aafdcd1eb

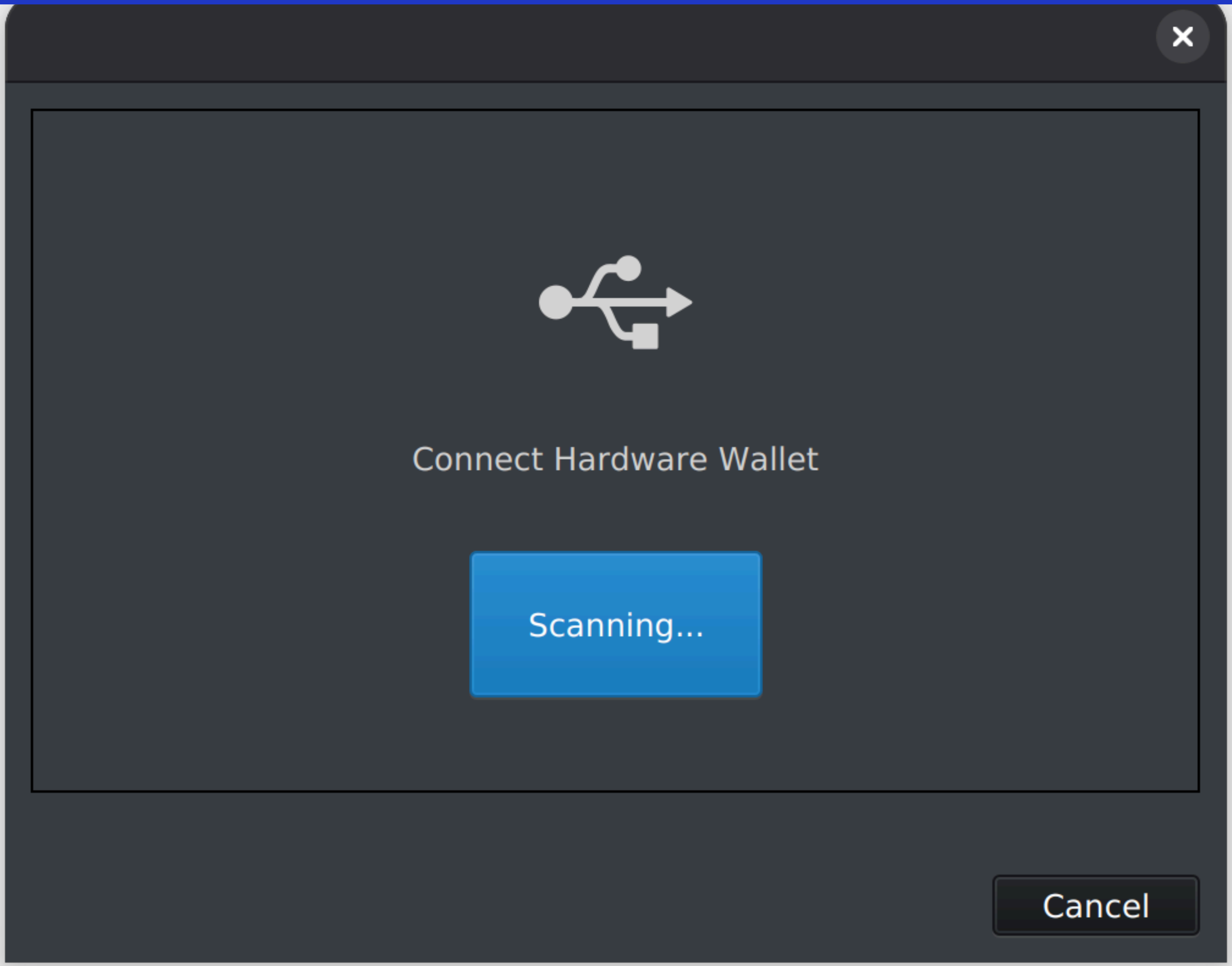
Receive 10,000 sats to tb1qshamaqp728k6rxkmqkuuequmggcp6e0sj37zv7a3j5778u3vvfrstzv6vv

Signatures

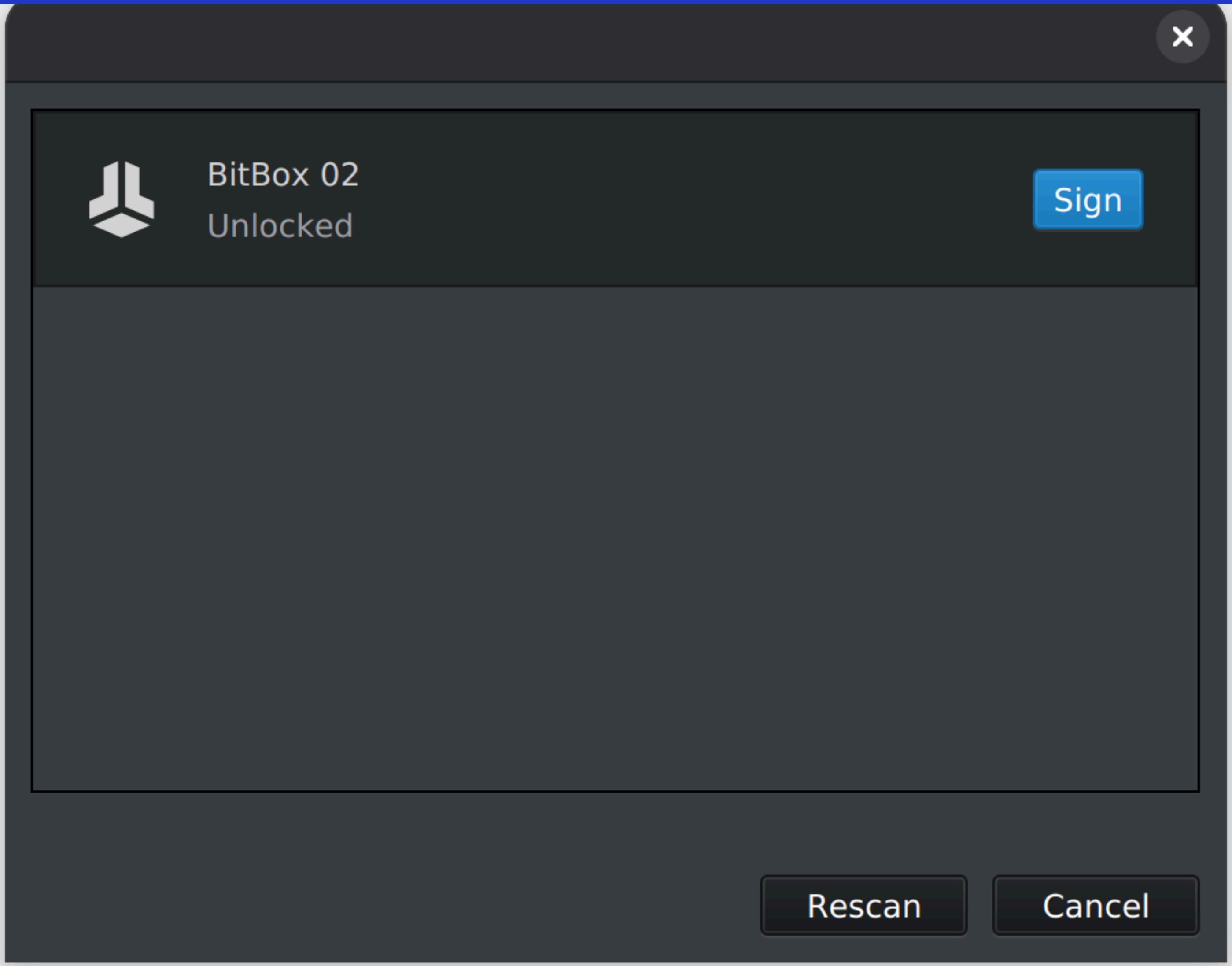
Jade 3

```
0200000002a2eced96c34305e86210bbf15f8f1a841976ce6d8312aa4921d7025e754de7a2e407000000fdffffffbe451f000e23bc60c4164acfdda7e3ff3689e10a
25282ad27253be6aa4da547ab703000000fdffffff024d270f00000000002200201a2282dc1812ee6aac3fb463f2cf3e5d678188e511a31fe52835d12e6d65595310
2700000000000022002085fbbe803e51eda19adb05b9cc839b42301d65f0947c267bb1953de3f22c624711320400
```

💸 Receiving and Spending Funds - BitBox02


Connect Hardware Wallet

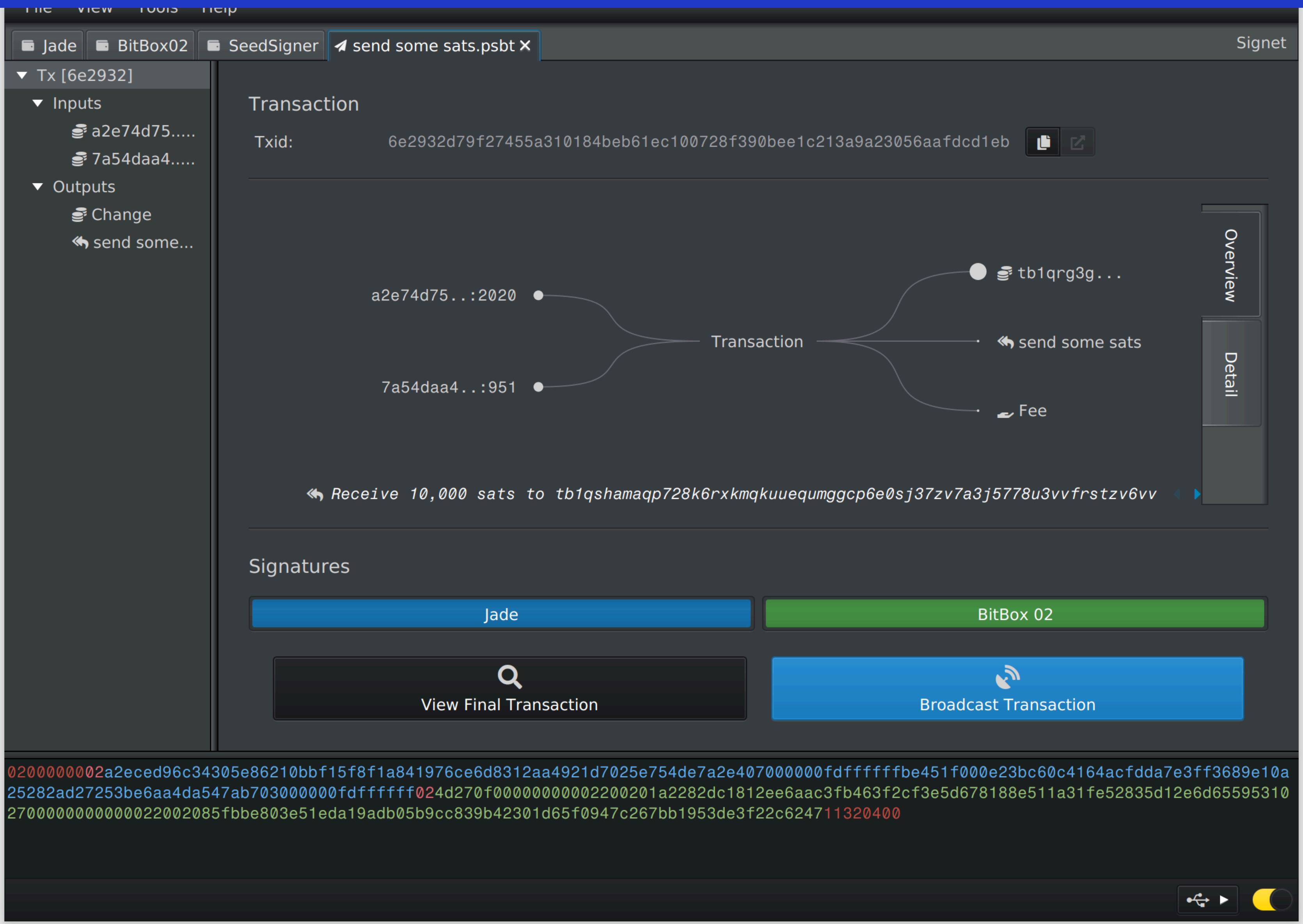
💸 Receiving and Spending Funds - BitBox02



BitBox 02

Unlocked

Receiving and Spending Funds - BitBox02



Transaction

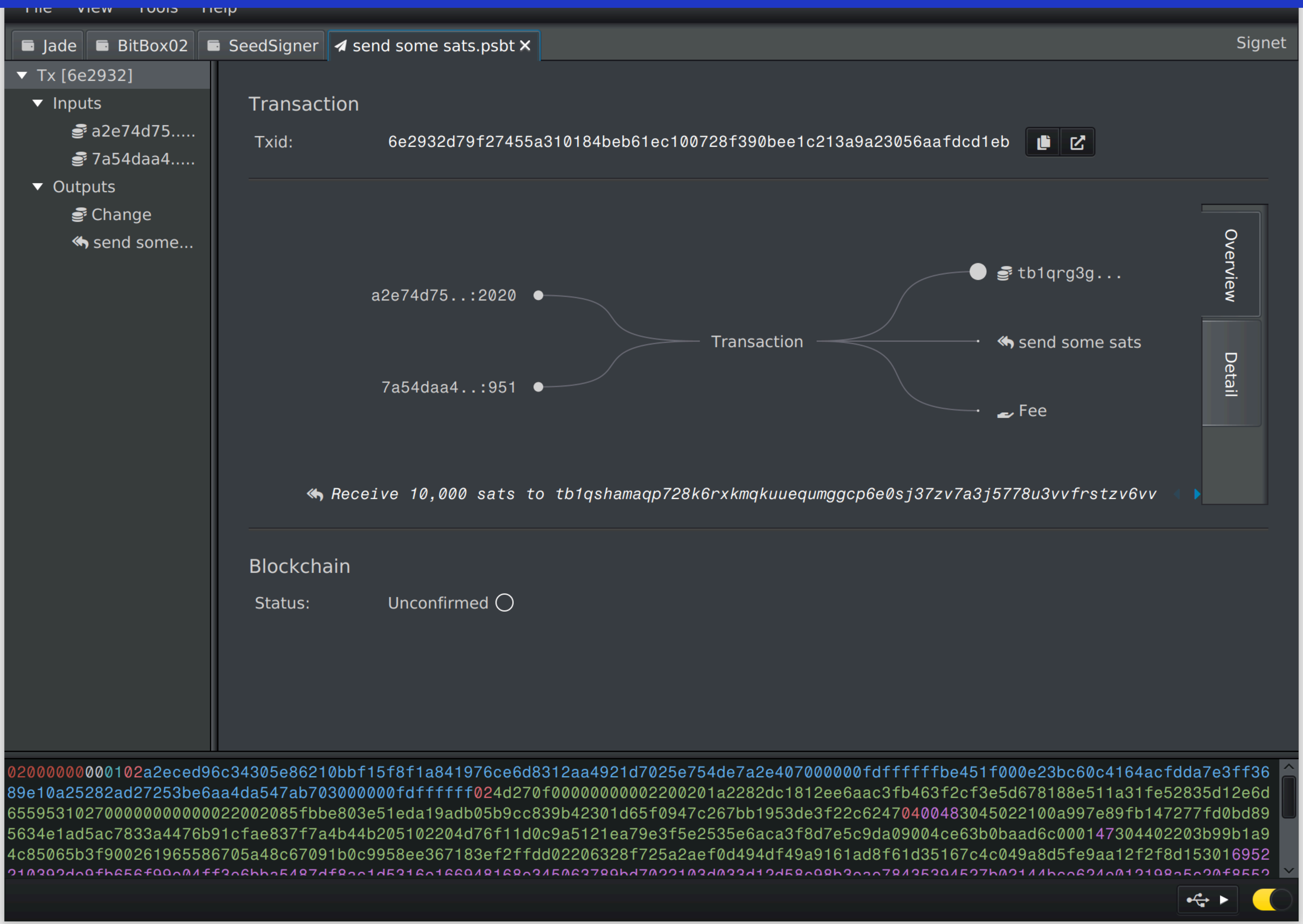
Txid: 6e2932d79f27455a310184beb61ec100728f390bee1c213a9a23056aafdcd1eb

Receive 10,000 sats to tb1qshamaqp728k6rxkmqkuuequmggcp6e0sj37zv7a3j5778u3vvfrstzv6vv

Signatures

```
0200000002a2eced96c34305e86210bbf15f8f1a841976ce6d8312aa4921d7025e754de7a2e407000000fdffffffbe451f000e23bc60c4164acfdda7e3ff3689e10a
25282ad27253be6aa4da547ab703000000fdffffff024d270f00000000002200201a2282dc1812ee6aac3fb463f2cf3e5d678188e511a31fe52835d12e6d65595310
270000000000000022002085fbbe803e51eda19adb05b9cc839b42301d65f0947c267bb1953de3f22c624711320400
```

Receiving and Spending Funds - Result



Transaction

Txid: 6e2932d79f27455a310184beb61ec100728f390bee1c213a9a23056aafdcd1eb

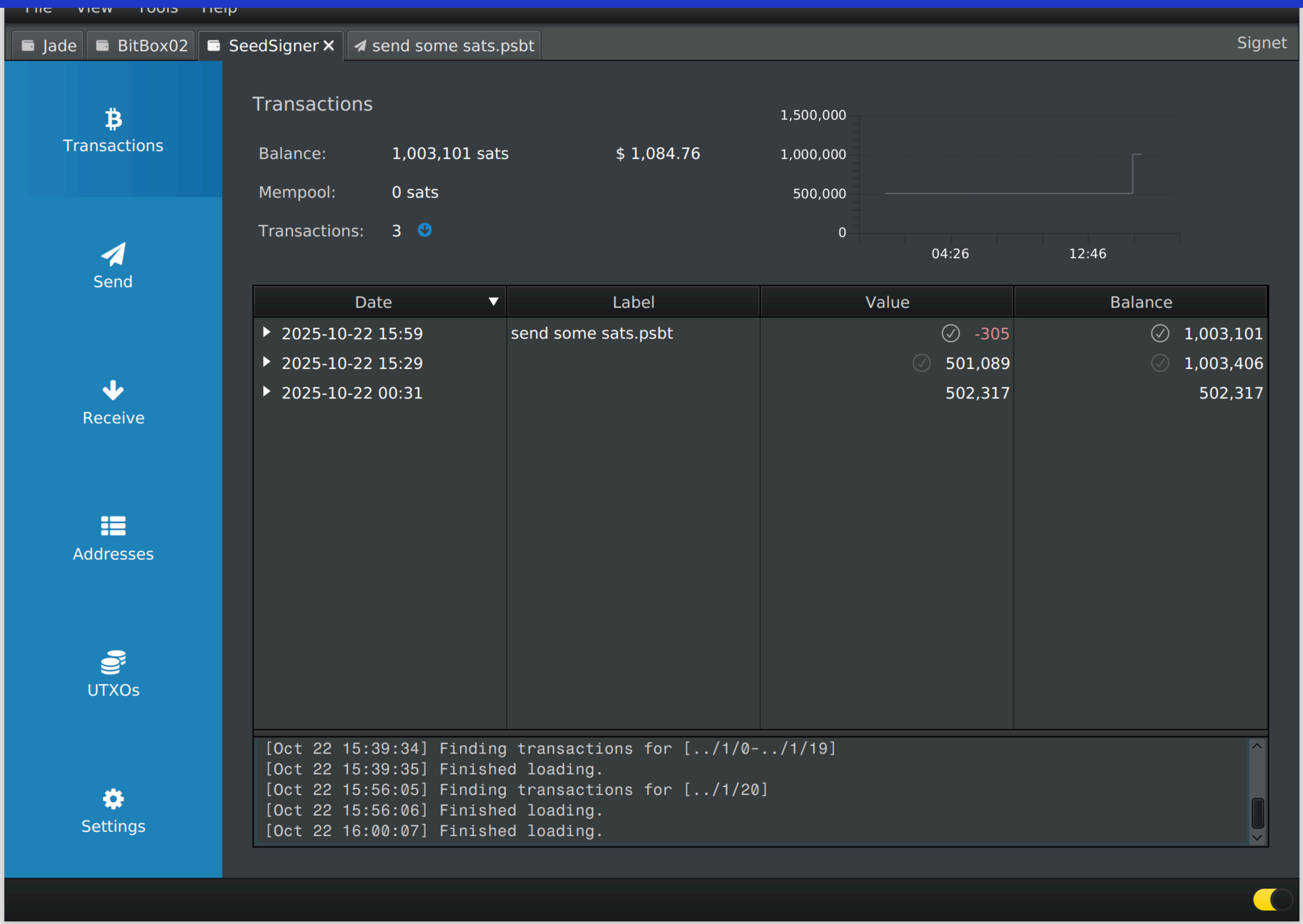
Receive 10,000 sats to tb1qshamaqp728k6rxkmqkuuequmggcp6e0sj37zv7a3j5778u3vvfrstzv6vv

Blockchain

Status: Unconfirmed ○

```
02000000000102a2eced96c34305e86210bbf15f8f1a841976ce6d8312aa4921d7025e754de7a2e407000000fdffffffbe451f000e23bc60c4164acfdda7e3ff36
89e10a25282ad27253be6aa4da547ab703000000fdffffff024d270f00000000002200201a2282dc1812ee6aac3fb463f2cf3e5d678188e511a31fe52835d12e6d
655953102700000000000022002085fbbe803e51eda19adb05b9cc839b42301d65f0947c267bb1953de3f22c62470400483045022100a997e89fb147277fd0bd89
5634e1ad5ac7833a4476b91cfae837f7a4b44b205102204d76f11d0c9a5121ea79e3f5e2535e6aca3f8d7e5c9da09004ce63b0baad6c000147304402203b99b1a9
4c85065b3f900261965586705a48c67091b0c9958ee367183ef2ffdd02206328f725a2aef0d494df49a9161ad8f61d35167c4c049a8d5fe9aa12f2f8d153016952
210392de9fb656f99e04ff3e6bba5487df8ac1d5316e166948168e345063789bd7022103d033d12d58c98b3eae78435394527b02144bce624e012198a5c20f8552
```

Receiving and Spending Funds - Result



Transactions

Balance: 1,003,101 sats $ 1,084.76

Mempool: 0 sats

Transactions: 3

Date	Label	Value	Balance
2025-10-22 15:59	send some sats.psbt	✓ -305	✓ 1,003,101
2025-10-22 15:29		✓ 501,089	✓ 1,003,406
2025-10-22 00:31		502,317	502,317

```
[Oct 22 15:39:34] Finding transactions for [../1/0-../1/19]
[Oct 22 15:39:35] Finished loading.
[Oct 22 15:56:05] Finding transactions for [../1/20]
[Oct 22 15:56:06] Finished loading.
[Oct 22 16:00:07] Finished loading.
```

👀 Q&A

On Telegram: @valeriovaccaro

Bibliography

- Blockstream Jade Documentation
- BitBox02 Documentation
- SeedSigner Documentation
- Sparrow Wallet Documentation

🖥️ Satoshi Spritz Project

- 💻 Federation of local Bitcoiner groups
- 🎓 Free and privacy-oriented events
- 🤖 BITCOIN ONLY
- 📖 Focused on learning self-sovereignty
- 🍕 Satoshi Spritz Connect every week online

https://satoshispritz.it

https://t.me/SatoshiSpritzConnect

₿ Officine Bitcoin

- 🤝 Italian Bitcoiners community, totally free
- 🤖 BITCOIN ONLY
- 🎓 Focus on education and project development
- 📋 Projects:
  - 💼 Bitcoin node development
  - 👨‍🏫 Using Hardware Wallets
  - 💻 Open source philosophy
  - 🤝 Debian installation
  - ... and much more

https://officinebitcoin.it