

# Private Payments in Practice

Integrating Zcash from  
Scratch to Production

ETHBoulder Feb 2026

Mylo Bennett aka ReadyMouse





# Motivation: Ethics in Practice

Maybe it's crazy, but I don't want the government (or anyone) tracking my purchases



## Banana Betting

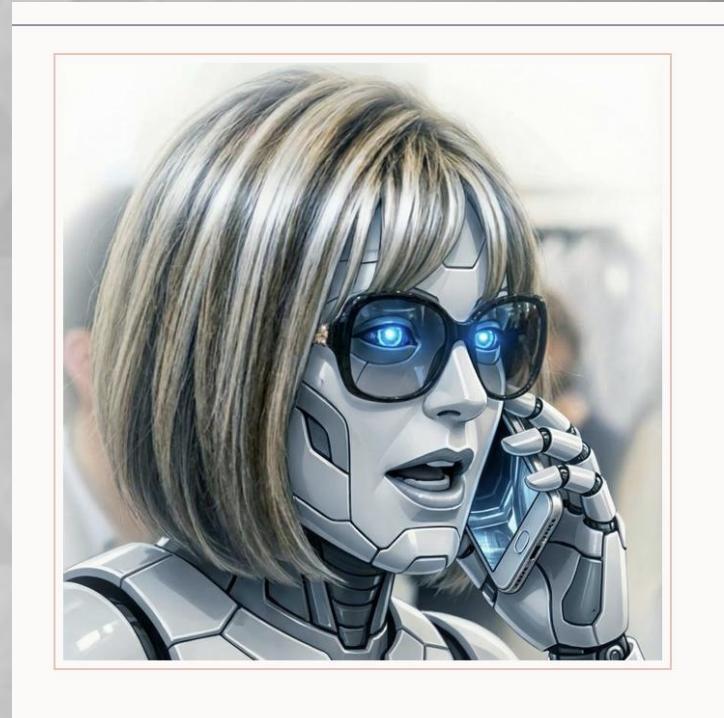
Zcash-Native Prediction Market

The screenshot shows a prediction market interface for a TikTok video. The main title is "Tiktok likes: Man-nana Dadbob Cheerleaders dancing to Shania Twain". Below it, there's a summary: "How many likes will the Tiktok video of Savannah Banana's Man-nana Dadbob Cheerleaders dancing to 'Man! I Feel Like a Woman!' by Shania Twain get in the next week? <https://www.tiktok.com/t/ZP8SKLYGw/>".  
**Betting Information:**  
Total Pool: **0.0000 ZEC**  
Bet Range: **0.000001 - 5 ZEC**  
Fees: House 5.0% + Creator 5.0% + Validators 20.0%  
**Event Stats:**  
Players: **0** Status: **Open**  
**Event Timeline:**  
Start Time (EST): **9/29/2025 1:33:00 PM**  
End Time (EST): **10/6/2025 1:33:00 PM**  
Settlement Time (EST): **10/7/2025**  
**Choose Your Outcome:**  
Option A: 300-350k Likes: **0%**  
Option B: >350k Likes: **0%**  
**Bet Amount:**  
Amount (ZEC): **0.000001 - 5 ZEC**  
Range: **0.000001 - 5 ZEC**  
**Place Bet**

Full node, from scratch with Zcash's SDK

## KarenBot.Com

Customer Service Advocate



BTCPay + ZEC Plugin



# Comment on Electric Coin Company



In January 2026, the entire Electric Coin Company development team resigned following a governance dispute with Bootstrap, the nonprofit that oversees ECC. The core of the conflict centers on differing views between ECC leadership and the Bootstrap board over how to structure funding and commercialization, particularly around the Zashi wallet, within the constraints of nonprofit law. The former ECC team has since formed a new company and plans to continue building on Zcash.

The Zcash *protocol* is open source, permissionless, and completely unaffected

Today's focus is on the technology



# Mainstream money...



		Square
Privacy	Shielded TX	No
	Native ZEC	No
Control	Custodial	Yes
	Run Own Node	No
	Fees	1%
Complexity	Set Up	Low
	Readiness	Mature
Multi-coin		BTC only



# Now accepting private money...



		Square	Zcash SDK	Zgo	BTCPay	NOWPayments
Privacy	Shielded TX	No	Yes	Yes	Yes	No
	Native ZEC	No	Yes	Yes	Yes	Yes
Control	Custodial	Yes	No	No	No	Optional
	Run Own Node	No	Yes	No	Optional	No
	Fees	1%	-	\$22/mo	\$10-30/mo	0.5%
Complexity	Set Up	Low	High	Moderate	Moderate	Low
	Readiness	Mature	Mature	Early	Mature	Mature
Multi-coin		BTC only	No	No	XMR, DASH,...	XMR, DASH,...

No clear winner. It's all about trade-offs.



		Square	Zcash SDK	Zgo	BTCPay	NOWPayments
Privacy	Shielded TX	No	Yes	Yes	Yes	No
	Native ZEC	No	Yes	Yes	Yes	Yes
Control	Custodial	Yes	No	No	No	Optional
	Run Own Node	No	Yes	No	Optional	No
	Fees	1%	-	\$22/mo	\$10-30/mo	0.5%
Complexity	Set Up	Low	High	Moderate	Moderate	Low
	Readiness	Mature	Mature	Early	Mature	Mature
Multi-coin		BTC only	No	No	XMR, DASH,...	XMR, DASH,...



# Wallet-to-Wallet TX Development Pipeline



## ETH-Based

1. Develop w. Hardhat's blockchain simulator
2. Test on fake blockchain
3. Find a faucet for testnet tokens
4. Connect to testnet node
5. Verify smart contract works on testnet
6. Publish on mainnet

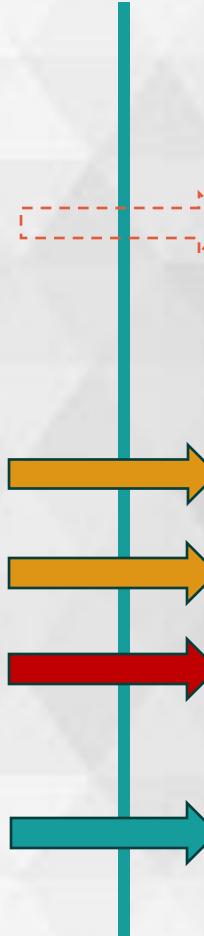


# Wallet-to-Wallet TX Development Pipeline



## ETH-Based

1. Develop w. Hardhat's blockchain simulator
2. Test on fake blockchain
3. Find a faucet for testnet tokens
4. Connect to testnet node
5. Verify smart contract works on testnet
6. Publish on mainnet



## ZEC-Based Development

- I can't find a ZEC blockchain simulator
- Testnet.zecfaucet → Wait, what wallet supports TAZ? Zingo
- Tatum.io RPC → READ ONLY, not helpful for a private blockchain
- No Smart Contracts, no programmable money
- Own Code + Own Node → Create wallets + receive addresses

Lack of smart contracts and mature development infrastructure for zcash



# Wallet-to-Wallet TX Development Pipeline



## ETH-Based

1. Develop w. Hardhat's blockchain simulator
2. Test on fake blockchain
3. Find a faucet for testnet tokens
4. Connect to testnet node
5. Verify smart contract works on testnet
6. Publish on mainnet

## ZEC-Based Development

I can't find a ZEC blockchain simulator

Testnet.zecfaucet → Wait, what wallet supports TAZ?

Tatum.io RPC → READ ONLY, not helpful for a private blockchain

No Smart Contracts, no programmable money

Own Code + Own Node → Create wallets + receive addresses

Lack of smart contracts and mature development infrastructure for zcash



# Own Code + Own Node



ZECHUB 2025 Hackathon 3rd Place Winner: Banana Betting

## Frontend

The screenshot shows a user interface for managing a Zcash wallet. It includes a yellow header with the title "Wallet Management". Below it, a large orange button displays the "Current Balance" as "0.0001 ZEC" with a refresh icon. Underneath, there are two input fields: one for a "Shielded Address (Private)" and another for a "Transparent Address (Public)", each with a copy icon. At the bottom, a green bar indicates "Wallet Connected" with a green dot and a "Refresh" button.

Not covered in this talk

## Backend

```
def z_getaddressforaccount(account: int, receiver_type=None, diversifier_index: int=None):
    params = [account]
    if receiver_type is not None:
        params.append(receiver_type)
    if diversifier_index is not None:
        params.append(diversifier_index)

    try:
        # RPC request payload
        payload = {
            "jsonrpc": "1.0",
            "id": "z_getaddressforaccount",
            "method": "z_getaddressforaccount",
            "params": params
        }

        # Make the request to the Zcash node
        response = requests.post(ZCASH_RPC_URL, json=payload, auth=(ZCASH_RPC_USER, ZCASH_RPC_PASSWORD))

        # Handle Zcash node response
        if response.status_code != 200:
            print(response.json())
            raise HTTPException(status_code=500, detail="Failed to connect to Zcash node")

        # Parse response
        validation_data = response.json()
        return validation_data['result']['address']

    except Exception as e:
        raise HTTPException(status_code=500, detail=str(e))
```

- Generate addresses
- Create transactions
- Get balances

## Full Node

```
1  # Simple `zcashd` Docker Compose configuration.
2  #
3  # THIS DAEMON IS BEING DEPRECATED AND SHOULD NOT BE USED FOR NEW
4  # SERVERS. ONLY USE THIS IF YOU KNOW YOU HAVE A SPECIFIC NEED FOR
5  # THE OLDER IMPLEMENTATION OF A ZCASH FULL NODE.
6  #
7  # Instead of this configuration, you should use `compose.yaml`, the
8  # new default with the `zebrad` service configuration.
9  ---
10 services:
11     zcashd:
12         image: electriccoinco/zcashd:v6.3.0
13         restart: unless-stopped
14         configs:
15             - source: zcashd_conf
16               target: /etc/zcashd/zcashd.conf
17         volumes:
18             - ./data/zcashd-cache:/srv/zcashd/.zcash
19             - ./data/zcashd-params:/srv/zcashd/.zcash-params
20         ports:
21             - 0.0.0:8232:8232
22             - 0.0.0:8233:8233
23         command: >
24             -conf=/etc/zcashd/zcashd.conf
25
26     configs:
27         zcashd_conf:
28             file: ./configs/zcashd.conf
```

- Docker + server
- Interacts with blockchain
- Read/write capabilities

Backend: "Smart contracts" for your own wallet, but it's python



# Own Code + Own Node



ZECHUB 2025 Hackathon 3rd Place Winner: Banana Betting

## Frontend

Wallet Management

Current Balance  
0.0001 ZEC

Shielded Address (Private)  
.....

Transparent Address (Public)  
.....

● Wallet Connected      Refresh

Not covered in this talk

## Backend

```
def z_getaddressforaccount(account: int, receiver_type=None, diversifier_index: int=None):
    params = [account]
    if receiver_type is not None:
        params.append(receiver_type)
    if diversifier_index is not None:
        params.append(diversifier_index)

    try:
        # RPC request payload
        payload = {
            "jsonrpc": "1.0",
            "id": "z_getaddressforaccount",
            "method": "z_getaddressforaccount",
            "params": params
        }

        # Make the request to the Zcash node
        response = requests.post(ZCASH_RPC_URL, json=payload, auth=(ZCASH_RPC_USER, ZCASH_RPC_PASSWORD))

        # Handle Zcash node response
        if response.status_code != 200:
            print(response.json())
            raise HTTPException(status_code=500, detail="Failed to connect to Zcash node")

        # Parse response
        validation_data = response.json()
        return validation_data['result']['address']

    except Exception as e:
        raise HTTPException(status_code=500, detail=str(e))
```

- Generate addresses
- Create transactions
- Get balances

## Full Node

```
1  # Simple `zcashd` Docker Compose configuration.
2  #
3  # THIS DAEMON IS BEING DEPRECATED AND SHOULD NOT BE USED FOR NEW
4  # SERVERS. ONLY USE THIS IF YOU KNOW YOU HAVE A SPECIFIC NEED FOR
5  # THE OLDER IMPLEMENTATION OF A ZCASH FULL NODE.
6  #
7  # Instead of this configuration, you should use `compose.yaml`, the
8  # new default with the `zebrad` service configuration.
9  --
10 services:
11     zcashd:
12         image: electriccoinco/zcashd:v6.3.0
13         restart: unless-stopped
14         configs:
15             - source: zcashd_conf
16               target: /etc/zcashd/zcashd.conf
17         volumes:
18             - ./data/zcashd-cache:/srv/zcashd/.zcash
19             - ./data/zcashd-params:/srv/zcashd/.zcash-params
20         ports:
21             - 0.0.0:8232:8232
22             - 0.0.0:8233:8233
23         command: >
24             -conf=/etc/zcashd/zcashd.conf
25
26     configs:
27         zcashd_conf:
28             file: ./configs/zcashd.conf
```

- Docker + server
- Interacts with blockchain
- Read/write capabilities

Backend: "Smart contracts" for your own wallet, but it's python



# Zcash Backend – To the Docs!



The documentation for the RPC call

## **z\_getbalanceforaccount - Zcash 6.2.0 RPC**

`z_getbalanceforaccount account ( minconf asOfHeight )`

Returns the account's spendable balance for each value pool ("transparent", "sapling", and "orchard").

Arguments:

1. account (numeric) The account number.
2. minconf (numeric, optional, default=1) Only include transactions confirmed at least this many times.
3. asOfHeight (numeric, optional, default=-1) Execute the query as if it were run when the blockchain was at the height specified by



The implementation for the RPC call

```
payload = {  
    "jsonrpc": "1.0",  
    "id": "z_getbalanceforaccount",  
    "method": "z_getbalanceforaccount",  
    "params": [account or 0, minconf]  
}
```

*# Make the request to the Zcash node*

```
response = requests.post(  
    ZCASH_RPC_URL,  
    json=payload,  
    auth=(ZCASH_RPC_USER, ZCASH_RPC_PASSWORD)  
)
```

Major functions needed:

- `Z_get_new_account`: make an account (wallet)
- `Z_getAddressForAccount`: make receive address(es)
- `Z_send_many`: send transaction(s)
- `Z_getBalance`
- `Z_listReceivedByAddress`: get all transactions for an address



# Zcash Backend – Examples



Woah! That's so cool...I want to write the backend of my production system using Python someone wrote for a hackathon!!

```
zcash_sdk > zcash_utils.py
 1 import requests
 2 from fastapi import Depends, FastAPI, HTTPException, status, Query
 3 from zcash_sdk > main_transactions.py
 4
 5     1 from fastapi import Depends, FastAPI, HTTPException, status
 6     2 from fastapi.middleware.cors import CORSMiddleware
 7     3 from sqlalchemy.orm import Session
 8     4 from zcash_sdk > zcash_wallet.py
 9     5
10    fr 6     import requests
11    fr 7     from fastapi import Depends, FastAPI, HTTPException, status, Query
12    fr 8     from ..zcash_mod import ZCASH_RPC_URL, ZCASH_RPC_USER, ZCASH_RPC_PASSWORD, DISABLE_ZCASH_NODE
13    fr 9     from decimal import Decimal
14    fr 10    import simplejson
15    fr 11
16    fr 12    # Mock balances for development (user_id -> balance)
17    fr 13    _mock_user_balances = {}
18    mo 14    _mock_pool_balance = 1000.0
19
20    ap 15
21    ap 16    def backupwallet(destination: str):
22    ap 17        try:
23    ap 18            # RPC request payload
24    ap 19            payload = {
25    ap 20                "jsonrpc": "1.0",
26    ap 21                "id": "backupwallet",
27    ap 22                "method": "backupwallet",
28    ap 23                "params": [destination]
29    ap 24        }
30
31    ap 25
32    ap 26        # Make the request to the Zcash node
33    ap 27        response = requests.post(ZCASH_RPC_URL, json=payload, auth=(ZCASH_RPC_USER, ZCASH_RPC_PASSWORD))
34
35    ap 28
36    ap 29        # Handle Zcash node response
37    ap 30        if response.status_code != 200:
38    ap 31            print(response.json())
39    ap 32            raise HTTPException(status_code=500, detail="Failed to connect to Zcash node")
40
41    de 33
42    de 34        # Parse response
43    de 35        validation_data = response.json()
44    de 36        print(validation_data)
45    de 37        return validation_data['result']
46
47    except Exception as e:
48        raise HTTPException(status_code=500, detail=str(e))
49
50
51    def z_get_new_account():
52        try:
```

<https://github.com/ReadyMouse/private-crypto-processors>

RPC Function Call definitions:

- Zcash\_wallet.py
- Zcash\_utils.py

Example: main\_transactions.py

Now we have code to...

- Make unique receive addresses for customers
- Check addresses balances for correct amount deposited
- Withdraw ZEC to outside wallet(s) using z\_sendmany

-> We need the node to test



# Own Code + Own Node



## Frontend

Wallet Management

Current Balance  
**0.0001 ZEC**

Shielded Address (Private)  
.....

Transparent Address (Public)  
.....

● Wallet Connected      Refresh

Not covered in this talk

## Backend

```
def z_getaddressforaccount(account: int, receiver_type=None, diversifier_index: int=None):
    params = [account]
    if receiver_type is not None:
        params.append(receiver_type)
    if diversifier_index is not None:
        params.append(diversifier_index)

    try:
        # RPC request payload
        payload = {
            "jsonrpc": "1.0",
            "id": "z_getaddressforaccount",
            "method": "z_getaddressforaccount",
            "params": params
        }

        # Make the request to the Zcash node
        response = requests.post(ZCASH_RPC_URL, json=payload, auth=(ZCASH_RPC_USER, ZCASH_RPC_PASSWORD))

        # Handle Zcash node response
        if response.status_code != 200:
            print(response.json())
            raise HTTPException(status_code=500, detail="Failed to connect to Zcash node")

        # Parse response
        validation_data = response.json()
        return validation_data['result']['address']

    except Exception as e:
        raise HTTPException(status_code=500, detail=str(e))
```

- Generate addresses
- Create transactions
- Get balances

## Full Node

```
1  # Simple `zcashd` Docker Compose configuration.
2  #
3  # THIS DAEMON IS BEING DEPRECATED AND SHOULD NOT BE USED FOR NEW
4  # SERVERS. ONLY USE THIS IF YOU KNOW YOU HAVE A SPECIFIC NEED FOR
5  # THE OLDER IMPLEMENTATION OF A ZCASH FULL NODE.
6  #
7  # Instead of this configuration, you should use `compose.yaml`, the
8  # new default with the `zebrad` service configuration.
9  ---
10 services:
11     zcashd:
12         image: electriccoinco/zcashd:v6.3.0
13         restart: unless-stopped
14         configs:
15             - source: zcashd_conf
16               target: /etc/zcashd/zcashd.conf
17         volumes:
18             - ./data/zcashd-cache:/srv/zcashd/.zcash
19             - ./data/zcashd-params:/srv/zcashd/.zcash-params
20         ports:
21             - 0.0.0:8232:8232
22             - 0.0.0:8233:8233
23         command: >
24             -conf=/etc/zcashd/zcashd.conf
25
26     configs:
27         zcashd_conf:
28             file: ./configs/zcashd.conf
```

- Docker + server
- Interacts with blockchain
- Read/write capabilities

Full Node: Decentralized database of all user transactions (transparent + shielded)

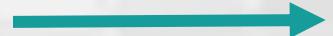


# Zcash Node – To the Docs!



## Node Requirements:

- ZEC-accepting server provider
- 300-400GB Blockchain storage
- 4-6GB RAM
- ZECsy, DevOps girlfriend who can run your infrastructure





# Zcash Node – To the Docs!



## Node Requirements:

- ZEC-accepting server provider
- 300-400GB Blockchain storage
- 4-6GB RAM
- ZECsy, DevOps girlfriend who can run your infrastructure



## Documentation:

- <https://github.com/zecrocks/zcash-stack/>
- Workshop on hosting read-only, lightwallet nodes

## Process:

- Create an account (wallet)
- Save your seed phrase/keys
- Start making addresses to accept TXs



```
# Zcashd configuration file.  
i-am-aware-zcashd-will-be-replaced-  
by-zebrad-and-zallet-in-2025=1  
txindex=1  
experimentalfeatures=1  
lightwalletd=1  
disablewallet=0  
rpcbind=0.0.0.0  
rpcallowip=0.0.0.0/0  
  
rpcuser=rpcuser  
rpcpassword=PASSWORD
```

## Example Docker File for Node Instance

### services:

#### zcashd:

```
image: electriccoinco/zcashd:v6.3.0  
restart: unless-stopped
```

#### configs:

```
- source: zcashd_conf  
  target: /etc/zcashd/zcashd.conf
```

#### volumes:

```
- ./data/zcashd-cache:/srv/zcashd/.zcash  
- ./data/zcashd-params:/srv/zcashd/.zcash-params
```

#### ports:

```
- 0.0.0.0:8232:8232  
- 0.0.0.0:8233:8233
```

#### command:

```
>  
-conf=/etc/zcashd/zcashd.conf
```

### configs:

#### zcashd\_conf:

```
file: ./configs/zcashd.conf
```



# End to End Testing



Yay! Let's check the blockchain.

Zcash Block Explorer

transaction / block / address

Mempool Blocks Tools

Details for the Zcash Transaction ID 0b1cf9e8d4536c68c52246e334504c296c5658bb07a45674cb9eb292d45dc909

Confirmations	Time (UTC)	Tx Type
1	2025-09-29 16:37:55 (17 seconds ago)	Shielded
Block Id	Public Inputs / Outputs	Shielded Inputs / Outputs
3082024	0 / 0	0 / 0
JoinSplits ?	Transferred from shielded pool	Overwintered?
No	0.0001 ZEC	true
Size (bytes)	Version	Locktime
9165	5	0
Transaction fee	Raw TX	Orchard Action transfers
0.0001 ZEC	JSON	2
PublicTransfers	Inputs (0)	Outputs (0)
	→	Shielded

Zashi Deposit

Wallet Management

Current Balance  
0.0001 ZEC

Shielded Address  
.....

Transparent Address  
.....

Wallet Co

Receiving  
0.000582ZEC

Transaction Details

Transaction ID  
b5157...cd4fd

Completed  
Sep 29, 2025 16:14

Confirmation it occurred, no details.

ZEC-native, private, non-custodial payment processing but with high technical overhead



		Square	Zcash SDK	Zgo	BTCPay	NOWPayments
Privacy	Shielded TX	No	Yes	Yes	Yes	No
	Native ZEC	No	Yes	Yes	Yes	Yes
Control	Custodial	Yes	No	No	No	Optional
	Run Own Node	No	Yes	No	Optional	No
	Fees	1%	-	\$22/mo	\$10-30/mo	0.5%
Complexity	Set Up	Low	High	Moderate	Moderate	Low
	Readiness	Mature	Mature	Early	Mature	Mature
Multi-coin		BTC only	No	No	XMR, DASH,...	XMR, DASH,...

ZEC-native, private, non-custodial payment processing but with high technical overhead



		Square	Zcash SDK	Zgo	BTCPay	NOWPayments
Privacy	Shielded TX	No	Yes	Yes	Yes	No
	Native ZEC	No	Yes	Yes	Yes	Yes
Control	Custodial	Yes	No	No	No	Optional
	Run Own Node	No	Yes	No	Optional	No
	Fees	1%	-	\$22/mo	\$10-30/mo	0.5%
Complexity	Set Up	Low	High	Moderate	Moderate	Low
	Readiness	Mature	Mature	Early	Mature	Mature
Multi-coin		BTC only	No	No	XMR, DASH,...	XMR, DASH,...



# Zgo



Starting off, let's check out their website...

Screenshot of Zgo home page

The screenshot shows the Zgo homepage. At the top is an orange header bar with the Zgo logo and navigation links: Home, News, Features, Guides, FAQ, Info, Contact, and Archive. Below the header is a white main content area. The headline reads "Electronic payment software for the 21st century." followed by the sub-headline "Helping you receive payments in a safe, private, and independent way." Under "Latest News", there are two items: "Unified Addresses are here!" (October 30, 2023) and "ZGo's Security Hardening" (June 27, 2023). To the right of the news is a red "Launch App" button. Below the news is a large graphic featuring two hands. One hand holds a blue shopping bag, and the other hand holds a yellow Zcash coin (z). Below this graphic is the text "Like cash, but digital" and a subtitle: "An electronic payment that goes directly from your customer to you, no third-parties involved."



# Zgo



Starting off, let's check out their website...

Screenshot of Zgo home page

The screenshot shows the Zgo homepage with the following content:

- Header:** ZGo The Zcash Register
- Navigation:** Home, News, Features, Guides, FAQ, Info, Contact, Archive
- Section:** Electronic payment software for the 21st century.  
Helping you receive payments in a safe, private, and independent way.
- Latest News:**
  - Unified Addresses are here! - October 30, 2023
  - ZGo's Security Hardening - June 27, 2023
- Launch App** button
- Image:** Two hands holding a yellow Zcash coin above a blue shopping bag.
- Text:** Like cash, but digital  
An electronic payment that goes directly from your customer to you, no third-parties involved.
- Pricing:** \$1 / day, \$6 / week, \$22 / month

...integration options. No APIs.

### Xero Accounting

- Designed for full accounting
- Manually create invoices
- View keys optional

### WooCommerce

- Requires Wordpress
- Requires view keys

### Zgo App

- Not a Point-of-Sale system
- Set-up and tracking



# Zgo

Starting off, let's check out their website...

Screenshot of Zgo home page

Electronic payment software for the 21st century.

Helping you receive payments in a safe, private, and independent way.

Latest News

- Unified Addresses are here! - October 30, 2023
- ZGo's Security Hardening - June 27, 2023

Launch App

Like cash, but digital

An electronic payment that goes directly from your customer to you, no third-parties involved.

\$1 / day  
\$6 / week  
\$22 / month

...integration options. No APIs.

### Xero Accounting

- Designed for full accounting
- Manually create invoices
- View keys optional

### WooCommerce

- Requires WordPress
- Requires view keys

### Zgo App

- Not a Point-of-Sale system
- Set-up and tracking

Example from Zechub.store

ZGo

ZecHub Inc

Invoice

Order ID: 23a14ba3-1c8e-4b84-8239-d5b9ad958381  
Date: Feb 6, 2026

Zcash price: \$246.63  
Total: 0.11957182

Item	Qty.	Price(USD)
ZecHub Inc order 3495	1	\$29.49
<b>Invoice Total:</b>		<b>\$29.49</b>

Payment Pending!!

Scan the QR code with your wallet to make payment  
Can't scan?  
Use this [wallet link](#), or

Copy Address      Copy Amount  
Copy Memo

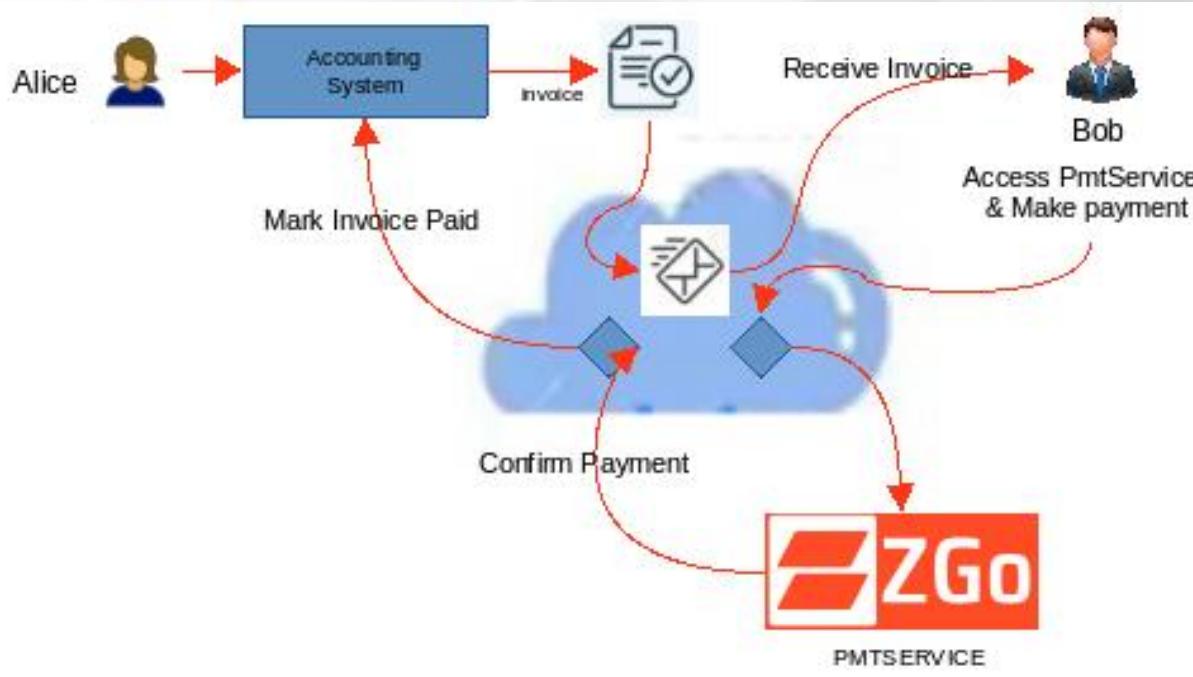
No API-token based access, requires multiple party integrations



# Zgo + Xero Accounting + Ywallet



Screenshot from Zgo.cash



## Xero Accounting

- Full accounting software
- Manually create invoices
- View keys optional
- Need a Ywallet for the business



## View Keys Optional

- Zgo would need READ ONLY access to your shielded transactions to confirm receipt and mark invoice as paid
- Zashi doesn't support view keys currently
- Need the older Ywallet to access view keys
- Alice could manually confirm invoice received and mark as paid, but then why bother with zgo?

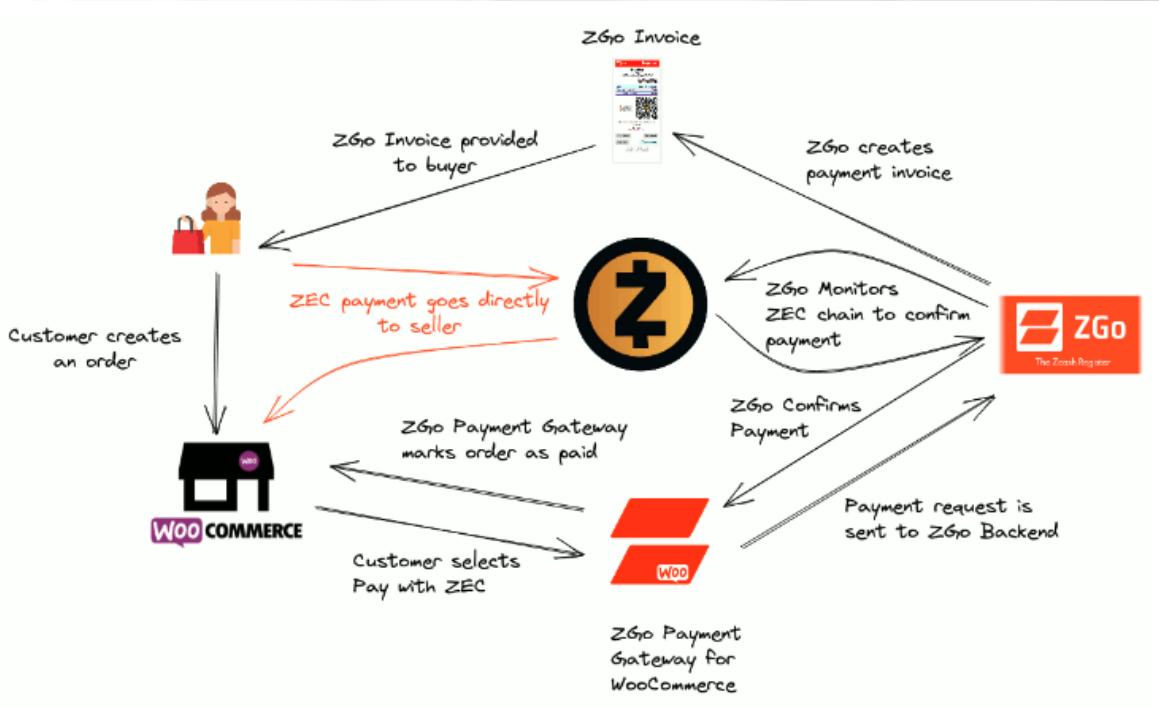
Accounting software designed for manual invoices but automatic confirmation



# Zgo + WooCommerce + WordPress + Ywallet



Screenshot from Zgo.cash



## Woo Commerce

- Requires WordPress
- Automatically create invoices
- Need a Ywallet for the business (desktop app available)

## Non-Rotating Receive Addresses

- On store set-up with Zgo, Alice sends a test message to Zgo with a code in the memo (no signing transactions like ETH)
- This sets ONE receive address
- View keys for this receive address
- Relies on the memo field to match invoice + payment
- !! Bookkeeping nightmare if user erases memo with unique invoice note and sends mismatched amount**

Simple option for WordPress-based sites looking for basic services



		Square	Zcash SDK	Zgo	BTCPay	NOWPayments
Privacy	Shielded TX	No	Yes	Yes	Yes	No
	Native ZEC	No	Yes	Yes	Yes	Yes
Control	Custodial	Yes	No	No	No	Optional
	Run Own Node	No	Yes	No	Optional	No
	Fees	1%	-	\$22/mo	\$10-30/mo	0.5%
Complexity	Set Up	Low	High	Moderate	Moderate	Low
	Readiness	Mature	Mature	Early	Mature	Mature
Multi-coin		BTC only	No	No	XMR, DASH,...	XMR, DASH,...

Simple storefront for Wordpress sites or low volume businesses using Xero accounting



		Square	Zcash SDK	Zgo	BTCPay	NOWPayments
Privacy	Shielded TX	No	Yes	Yes	Yes	No
	Native ZEC	No	Yes	Yes	Yes	Yes
Control	Custodial	Yes	No	No	No	Optional
	Run Own Node	No	Yes	No	Yes	No
	Fees	1%	-	\$22/mo	\$10-30/mo	0.5%
Complexity	Set Up	Low	High	Moderate	Moderate	Low
	Readiness	Mature	Mature	Early	Mature	Mature
		BTC only	No	No	XMR, DASH,...	XMR, DASH,...



# Major System Components



## BTCPay Server

The screenshot shows the BTCPay Server dashboard. At the top, there's a green banner with the text "Store successfully created". Below it, the main area is titled "BTCPayServerDemo" with the sub-instruction "To start accepting payments, set up a wallet or a Lightning node". There are three main buttons: "Create your store" (highlighted with a green checkmark), "Set up a wallet", and "Set up a Lightning node". On the left sidebar, there are sections for Dashboard, Settings, Wallets (Bitcoin and Lightning), Payments (Invoices, Reporting, Requests, Pull Payments, Payouts), and Plugins (Shopify, Point of Sale, Pay Button, CrowdFund, Manage Plugins). A small "BTCPAY 87" badge is in the top right corner.

## Zcash Plugin

The screenshot shows a GitHub repository named "btcpayserver-zcash-plugin" which is public. It has 4 branches and 1 tag. The repository contains several files and folders: ".gitignore", ".gitmodules", "LICENSE.md", "README.md", "Plugins/ZCash", "docs", ".dockerrcignore", "btcpayserver @ 6b727dd", and "btcpay-zcash-plugin.sln". The commit history includes a pull request from "macintoshhelper" to fix a block explorer url.

+

=>

## Operational Front

The screenshot shows the operational front end of BTCPay Server. It displays a QR code for payment, with a Bitcoin icon overlaid on it. The UI includes fields for "Pay with" (Bitcoin (BTC)), "BTCPay Server" (0.0000010 BTC, 1 BTC = \$(USD)), "Scan" and "Copy" buttons, and a green "Open in wallet" button at the bottom. A note at the bottom right says "Recommended fee: sat/byte".



# Major System Components



## BTCPay Server

BTCPay Server Demo

To start accepting payments, set up a wallet or a Light

- Create your store
- Set up a wallet
- Set up a Lightning node

WALLETS

- Bitcoin
- Lightning

PAYMENTS

- Invoices
- Reporting
- Requests
- Pull Payments
- Payouts

PLUGINS

- Shopify
- Point of Sale
- Pay Button
- Crowdfund
- Manage Plugins



## Zcash Plugin

btcpayserver-zcash-plugin Public

master 4 Branches 1 Tag

macintoshhelper fix block explorer url

Plugins/ZCash fix block

btcpayserver @ 6b727dd add btc/c

docs fix custo

.dockerignore Init comm

.gitignore Init comm

.gitmodules Init comm

LICENSE.md Init

README.md add full r

btcpay-zcash-plugin.sln Init

## Operational Elements

### Hosted BTCPay Server

- 1-Click VPS/Cloud options
- Self-Hosted, Self-Custody
- Enables multi-coin support

### Full ZCash Node

- Enables own custodial zcash wallet
- Self-Hosted
- + Zingo/Ywallet view keys



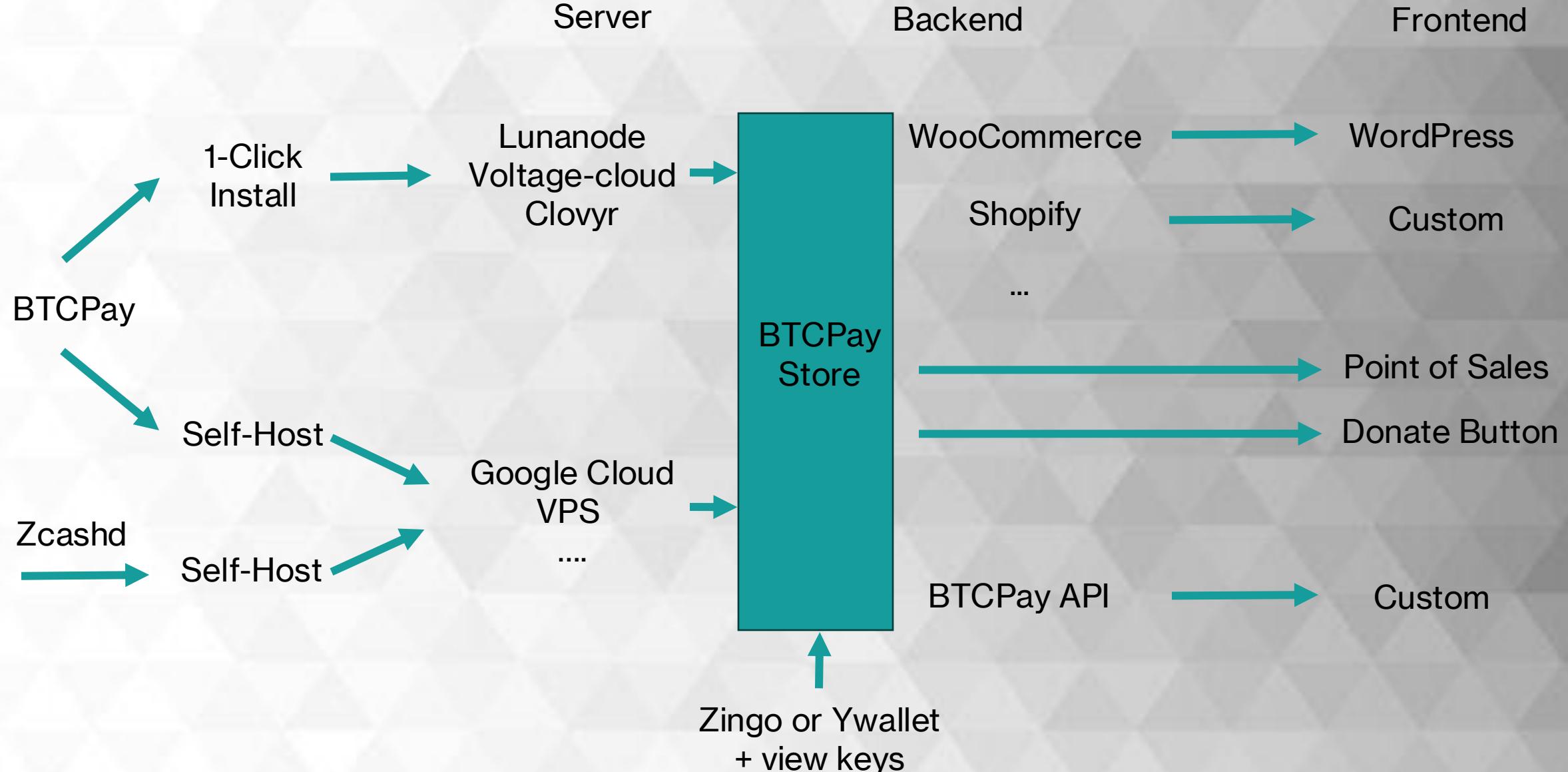
### Frontend:

- Custom Site w. API
- WooCommerce, Zapier, Shopify, etc
- Plus 24+ other integration partners
- Point of Sales App
- Embeddable HTML pay button

Zcash Community Grant funded Zcash Plugin, enables ZEC payments via BTCPay

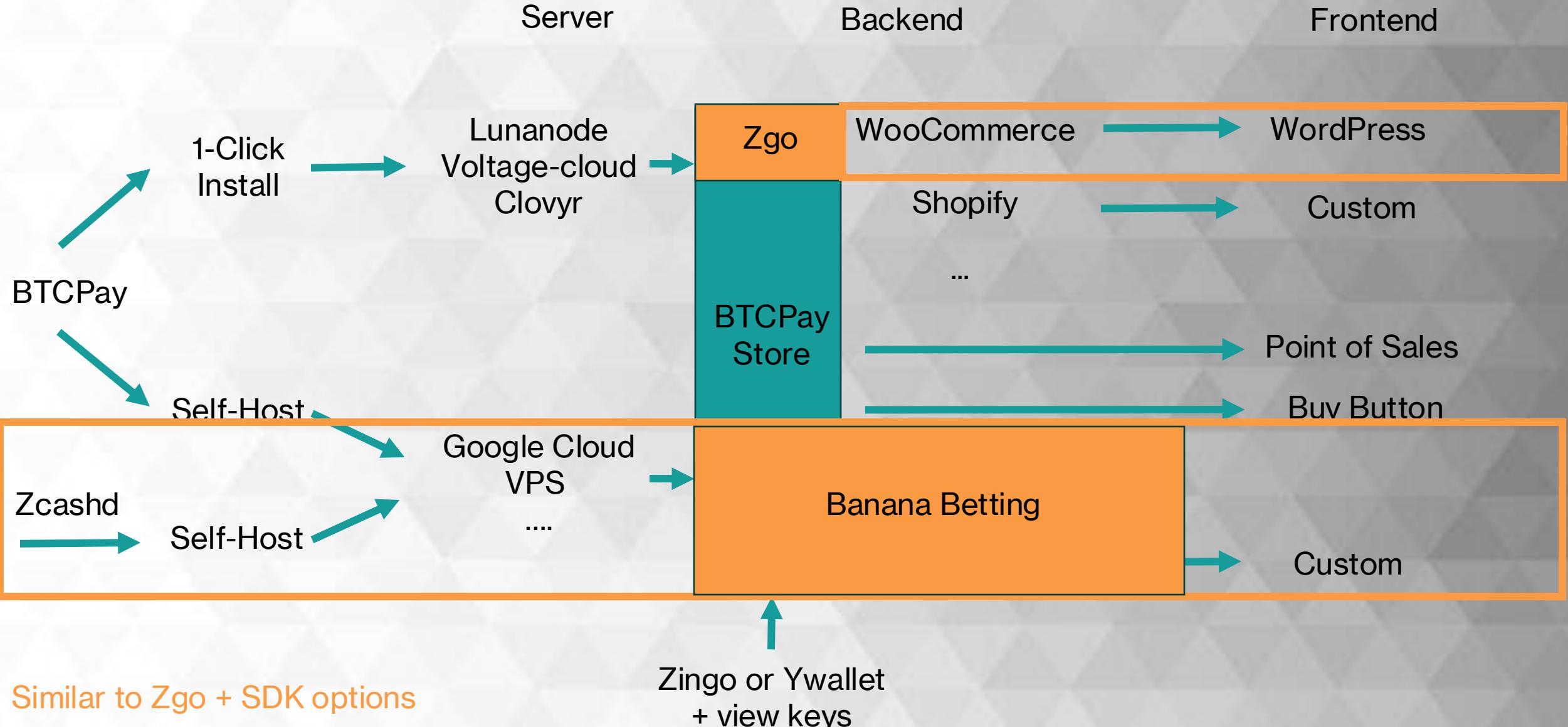


# Deployment Options



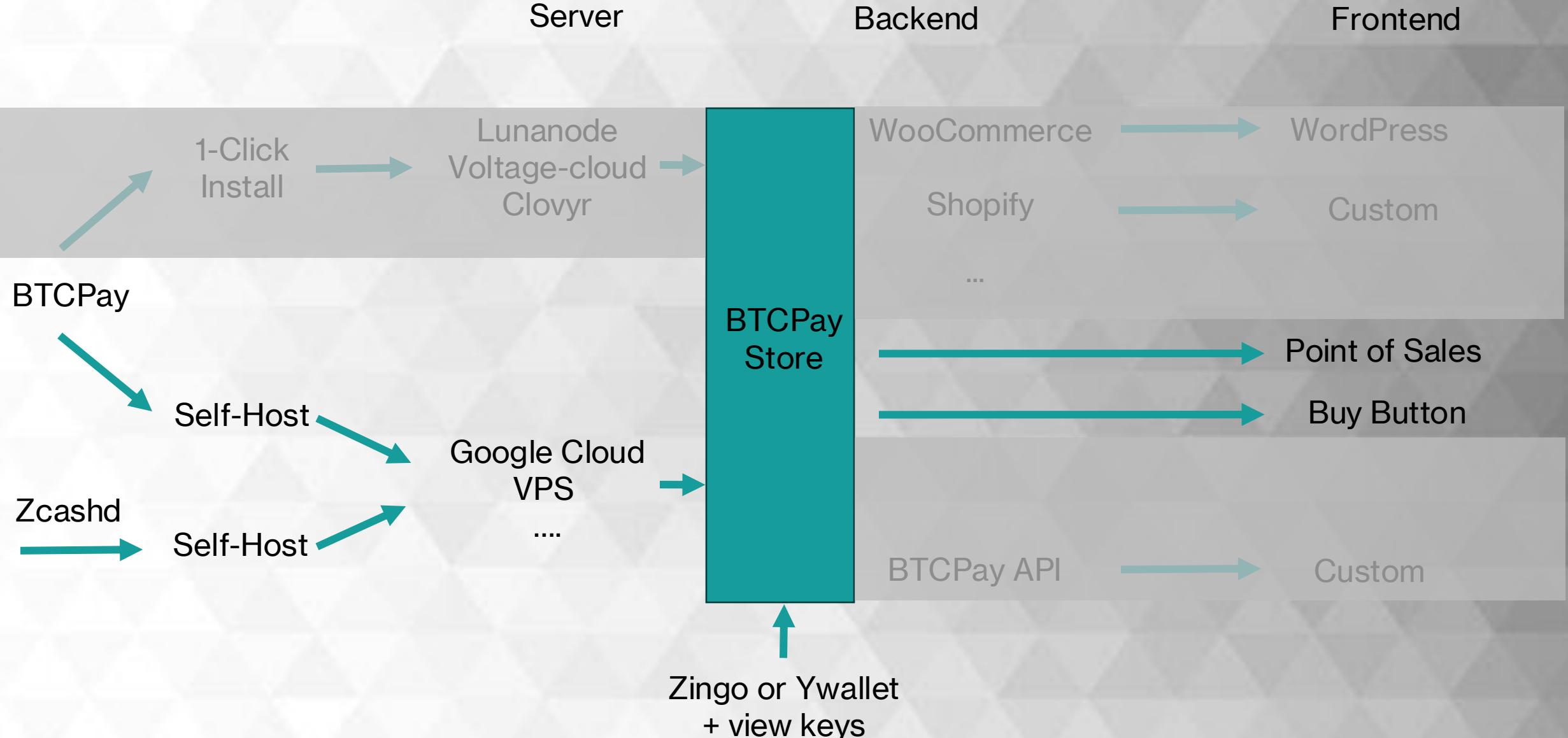


# Deployment Options





# Deployment Options

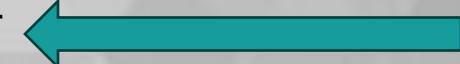




# From-Scratch Backend Implementation



```
# Clone the docker fragment repository  
git clone https://github.com/btcpay-zcash/btcpayserver-docker
```



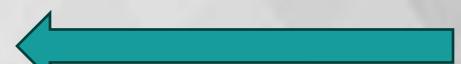
**Copy the code!**

```
# Add ZEC plugin  
git checkout feat/zec
```



**Add ZEC plugin**

```
# Run btcpay-setup.sh with the right parameters  
export BTCPAY_HOST="btcpay.example.com"  
export NBITCOIN_NETWORK="mainnet"  
export BTCPAYGEN_CRYPTO1="zec"  
export BTCPAYGEN_CRYPTO2="btc"  
export BTCPAYGEN_CRYPTO2="..."  
export BTCPAYGEN_ADDITIONAL_FRAGMENTS=""  
export BTCPAYGEN_REVERSEPROXY="nginx" ./btcpay-setup.sh -l
```



**Multi-coin Support**

```
# Full Node  
-> **The first part of the talk on running Zcash nodes**
```

Requires CLI familiarity and networking skills



# BTCPay GUI Setup 1 of 2



The screenshot shows the BTCPay Server demo interface. On the left, there's a sidebar with various menu items like Dashboard, Settings, Wallets (Bitcoin, Lightning), Payments (Invoices, Reporting, Requests, Pull Payments, Payouts), and Plugins (Shopify, Point of Sale, Pay Button, Crowdfund). The main area has a green header bar that says "Store successfully created". Below it, the title "BTCPayServerDemo" is displayed, followed by the instruction "To start accepting payments, set up a wallet or a Lightning node." A teal arrow points from the text "Create your store!" to the "Create your store" button, which is highlighted with a green checkmark icon.

**Create your store!**

This screenshot shows the "Available Plugins" section of the BTCPay Server interface. It lists the "BTCPay Server: ZCash support plugin" by btcpay-zcash, version 1.0.2.0. The plugin details include its purpose ("This plugin extends BTCPay Server to enable users to receive payments via ZCash."), dependencies ("BTCPayServer: >=2.1.0"), and resources ("Sources", "Documentation"). A red "Uninstall" button is visible. Below this, a search bar contains the text "Zcash" and a "Search" button. A yellow arrow points from the text "Add ZEC plugin" to the "Available Plugins" section.

**Add ZEC plugin**

Graphic interface for store creation and ZEC plugin



# BTCPay GUI Setup 2 of 2



The screenshot shows the BTCPay Server demo interface. At the top, there's a green banner with the text "Store successfully created". Below it, the title "BTCPayServerDemo" is displayed, followed by the instruction "To start accepting payments, set up a wallet or a Lightning node." Three options are listed: "Create your store" (marked with a checkmark), "Set up a wallet", and "Set up a Lightning node". A large teal arrow points from the text "Set up your wallets" on the right towards the "Set up a wallet" button. On the left sidebar, there are sections for "Dashboard", "Settings", "WALLETS" (with "Bitcoin" and "Lightning" selected), "PAYMENTS" (with "Invoices", "Reporting", "Requests", "Pull Payments", and "Payouts"), and "PLUGINS" (with "Shopify", "Point of Sale", "Pay Button", "Crowdfund", and "Manage Plugins").

**Set up your wallets**

**Point of Sales  
Pay Button integrations**

**BTCPay includes plugins for Point of Sales App and HTML site integrations**



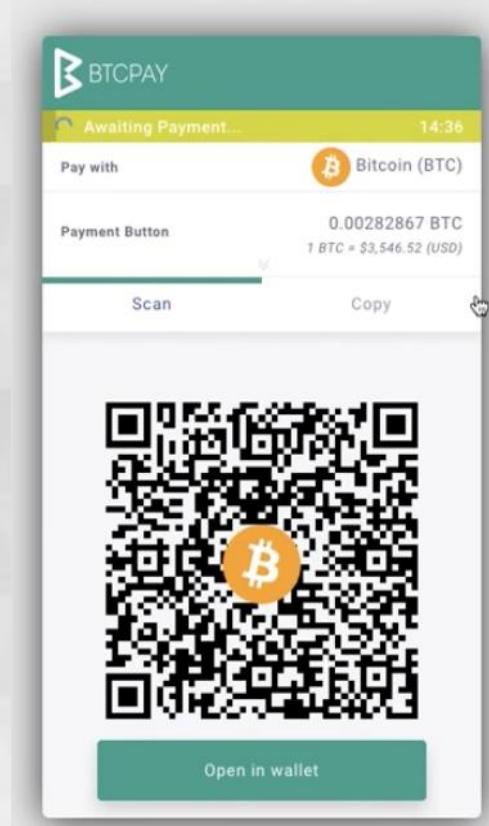
# Frontend Implementation



## BTCPay Point of Sales



## Embeddable HTML



Now accepting shielded ZEC



# Frontend Implementation



BTCPay Point of Sales



Embeddable HTML



Ready for Business



Now accepting shielded ZEC



		Square	Zcash SDK	Zgo	BTCPay	NOWPayments
Privacy	Shielded TX	No	Yes	Yes	Yes	No
	Native ZEC	No	Yes	Yes	Yes	Yes
Control	Custodial	Yes	No	No	No	Optional
	Run Own Node	No	Yes	No	Yes	No
	Fees	1%	-	\$22/mo	\$10-30/mo	0.5%
Complexity	Set Up	Low	High	Moderate	Moderate	Low
	Readiness	Mature	Mature	Early	Mature	Mature
		BTC only	No	No	XMR, DASH,...	XMR, DASH,...



# NOWpayments



Forbes ADVISOR BEST OF 2025

SIGMA AWARDS EUROMED

**Dashboard**

Balance: 24.687 BTC | Payouts: 56 | Turnover: 184.58 BTC 🔥

System status: ok | Currencies availability: 99%

Check the whole list on the [coin status page](#)

**Statistics**

Turnover: 400k\$

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Now Dec

**Useful resources**

**Transaction history**

Payment ID	Order ID	Original price	Amount sent / received	Status	Created / Last update
...	...	...	...	Finished	...
...	...	...	...	Finished	...

- + Fiat off-ramping
- + Currency Exchange

## Accept Zcash Payments

NOWPayments is a crypto payment gateway that lets you accept ZCash payments and donations in an easy and convenient way. Try this ZCash payment gateway NOW!

- ✓ Accept anything: 300+ cryptocurrencies supported
- ✓ Let your customers pay with anything: Auto coin conversion
- ✓ Explore many tools: API, plugins, invoices, donation tools and customized solutions
- ✓ Non-custodial: you are in control
- ✓ Save on fees: cheapest on the crypto market

- Only transparent ZEC

Fully featured PoS payment processor, including fiat off-ramping, but only transparent ZEC



# Summary



		Square	Zcash SDK	Zgo	BTCPay	NOWPayments
Privacy	Shielded TX	No	Yes	Yes	Yes	No
	Native ZEC	No	Yes	Yes	Yes	Yes
Control	Custodial	Yes	No	No	No	Optional
	Run Own Node	No	Yes	No	Optional	No
	Fees	1%	-	\$22/mo	\$10-30/mo	0.5%
Complexity	Set Up	Low	High	Moderate	Moderate	Low
	Readiness	Mature	Mature	Early	Mature	Mature
Multi-coin		BTC only	No	No	XMR, DASH,...	XMR, DASH,...

No clear winner. It's all about trade-offs.



# The Future



What's Missing...

- Square + NOWPayments: Convince them to support shielded ZEC
- SDK:
  - Terraform modules for zcash full nodes (mainnet + testnet)
  - Maintained Python packages, JavaScript library, etc for major function calls
    - Some partial projects exist
- Zgo:
  - Upgrade the UX
  - Add other integration partners
- BTCPay:
  - Rotating shielded address for Zcash plugin
  - Terraform module for BTCPay + Zcash node combo
- Development:
  - Hardhat-like blockchain simulator for local development



Wanna build one of these? Apply for a grant: <https://zcashcommunitygrants.org>



# Resources



Code Examples + This Slide Deck	<a href="https://github.com/ReadyMouse/BananaBetting">https://github.com/ReadyMouse/BananaBetting</a>
ZEC RPC Nodes	<a href="https://docs.tatum.io/reference/rpc-zcash">https://docs.tatum.io/reference/rpc-zcash</a>
Pay Processor for Non-profits	<a href="https://www.every.org/crypto">https://www.every.org/crypto</a>
Zcashd Node Docker Files	<a href="https://github.com/zecrocks/zcash-stack/blob/main/docker/compose.zcashd.yaml">https://github.com/zecrocks/zcash-stack/blob/main/docker/compose.zcashd.yaml</a>
Zcash Community Grants	<a href="https://zcashcommunitygrants.org/">https://zcashcommunitygrants.org/</a>
BTCPay Zcash Plugin	<a href="https://github.com/btcpay-zcash/btcpayserver-zcash-plugin/tree/master">https://github.com/btcpay-zcash/btcpayserver-zcash-plugin/tree/master</a>