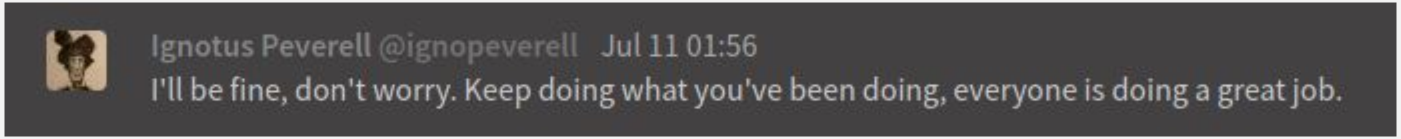
grincon1
2019.11.22 // c-base berlin

State of Grin

Thoughts on the past, present, and future

@yeastplume
@lehnberg

Ignotus Peverell


Ignotus Peverell @ignopeverell Jul 11 01:56

I'll be fine, don't worry. Keep doing what you've been doing, everyone is doing a great job.

grin tl;dr

Better money ツ

A lightweight implementation of a cryptocurrency that aims to be privacy preserving, scalable, and fair.

🐘

Addressing the elephant in the room

- Ability to see inputs/outputs and link them is a known limitation of Mimblewimble. This can be turned into a transaction graph.
- If you see unaggregated txs you can link inputs and outputs (related to a specific tx). We see very little aggregation today due to low activity.
- Even with high volume *some* nodes will still see little aggregation -> attacker can deploy more nodes to uncover more relationships.
- Open research problem, of constant high priority since before launch.

What does this mean for privacy?

- Privacy is hard, complex and involves many layers. The cryptocurrency protocol that you use is only one of these layers.
- If you transact with an attacker in Grin, you are exposed to risk.
- If you transact with an exchange that you give KYC to, you increase this risk.
- Side-channel and intersection attacks can be really powerful.
- Grin needs to communicate these limitations more clearly.

grincon0+1

Grin development since grincon0

- Launch 🚀
- 7 releases 📦: 1 x Major, 2 Minor, 4 patches
- 2 x Security Audits, 1 x Critical Vulnerability fixed 👾
- Bulletproof rewind, v2 wallet API, slate versioning, variable size kernels, windows support, tor support 🛠️
- A slightly more formalized planning process 👩‍🏫

Grin developer ecosystem since grincon0

- 12+ wallet projects, 7+ active 👛
- 3+ infrastructure services 🔗
- 30+ exchange integrations 🏦
- 8+ mining pools 🏊‍♀️
- 10+ mining software ⛏️
- 4(?) ASICs announced, 2(?) still in race 🍟
- 5+ forks 🍴

Grin governance since grincon0

- BDFL gone missing 💨
- New governance iteration proposed 👨‍💼
- RFC process introduced: 5 accepted, 4 in process, 4 in draft ✍️
- Three subteams created 👫
- Two anonymous BTC coinbase donations received 💰
- ~126 BTC in the Grin General Fund 🤑

Grin community since grincon0

- Multiple community chat groups on Gitter, Keybase, Telegram, Discord 💬
- 20+ local events held in 13 cities across the world 🌐
- One project website migrated and redesigned 🕸
- Webshops like TMGOX selling Grin swag 👕
- 2k+ readers of the weekly newsletter 📰
- Israeli football team shirt sponsorship (!?) ⚽
- LovelyGrin: our resident abstract painter 👨‍🎨

grincon1

2019.11.22 // c-base berlin

Questions I ask myself

How can onboarding be made easier?

- Contributors 👦💻
- Researchers 👦🔬
- Engineers 👦🔧
- Users 👦🌾

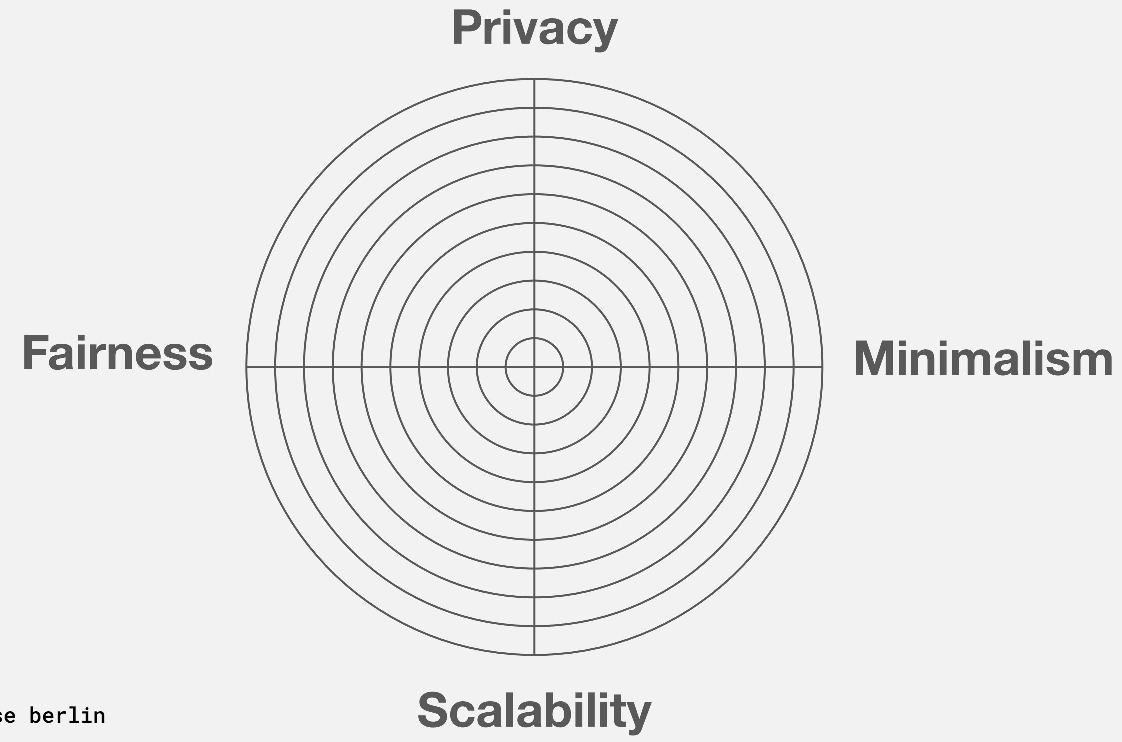
How does governance become more resilient?

- Eliminating single points of failures?
- Improve on Council / core team structures?
- Making sub-teams more autonomous?
- More open and transparent decisions and actions?

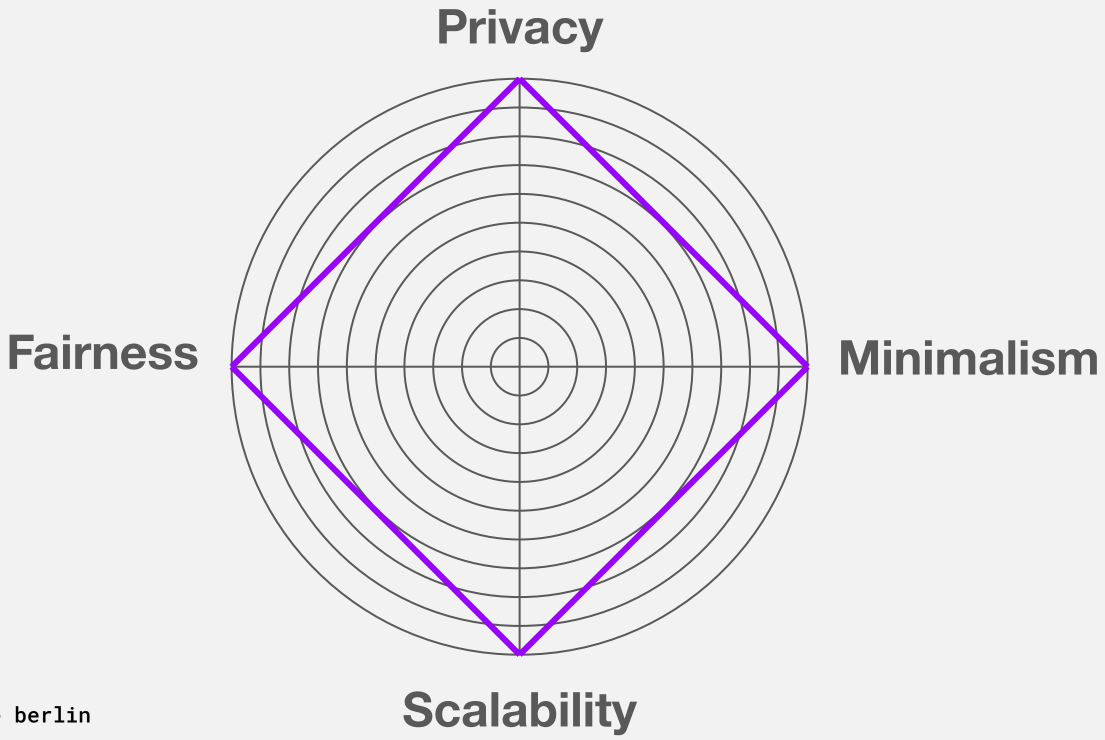
How can funds be put to good use?

- Does throwing money on a problem in this space help solve it?
- All contributors so far have started by contributing freely, is it time to rethink this approach? Are ‘guns for hire’ an option here?
- How can funding process be ensured to be fair for all who apply?
- What else could be done? Bounties, grants, fellowships? Community funding models? Sub-team budgets?
- What is the horizon, and what are the objectives?

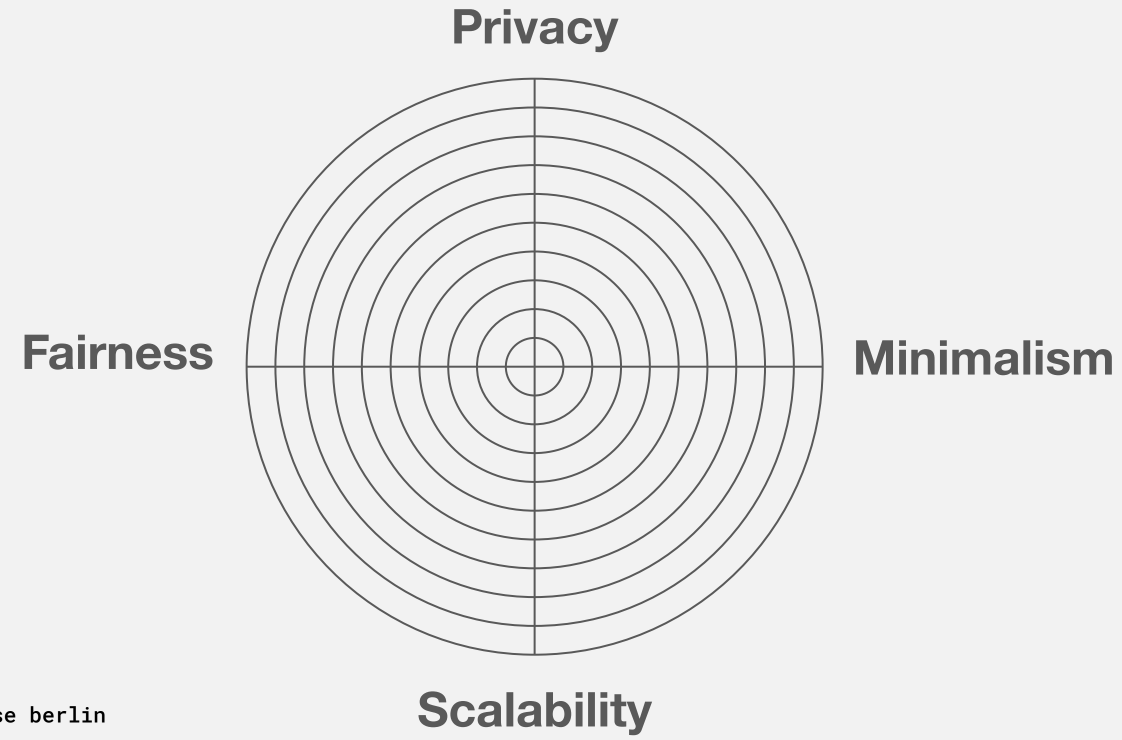
How are Grin's core values prioritized?



Is this us today?



Prioritizing helps focus and making trade-offs



Things I'd like to see for Grin in 2020

Underpinning. *The process of strengthening the foundation of an existing building or other structure.*

General

- Define more clear design priorities for Grin.
- Improve Developer Experience (DX).
- Establish more sub-teams, add more strong voices to the community.
- Formalize funding proposal process.
- Attract more attention from the research community.
- Define upgrade strategies beyond the scheduled hard forks.
- Improve overall communication and messaging.

Development

- Improve transaction building for better wallet designs.
- Prevent accidental use of unsafe practices.
- Improve the the experience for light clients / reduce reliance on hosted nodes for wallets.
- Iterate on transaction propagation.
- Iterate on p2p network.
- Reduce third party dependencies.

What is the roadmap process?

- Idea is to define prioritized areas and items, it’s still an open source project.
- Community has shared ideas and proposals on the forum.
- RFC to be drafted and published as an open document.
- Developers and community reviews, feedbacks, and merges.
- Pay attention to forum and `/grin-rfcs` repo.

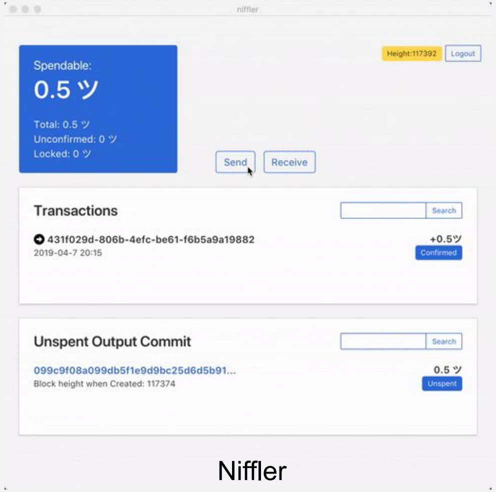
Y U No GUI?

(Wallet Development Since Launch)

Wallet Development Strategy

- Focus on the core technology
- Focus on tooling that enables the community
- Panic and worry about whether this is the right strategy

Wallet Ecosystem


niffler

Spendable:
0.5 ツ
Total: 0.5 ツ
Unconfirmed: 0 ツ
Locked: 0 ツ

Height:117392 | Logout | Send | Receive

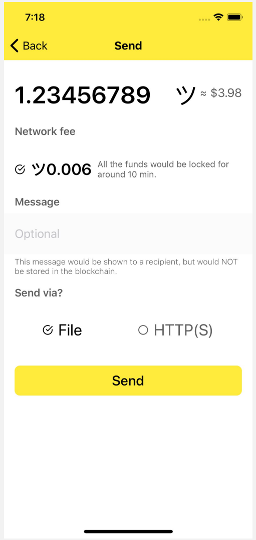
Transactions

431f029d-806b-4efc-be61-f6b5a9a19882 +0.5ツ
2019-04-7 20:15 Confirmed

Unspent Output Commit

099c9f08a099db5f1e9d9bc25d6d5b91... 0.5 ツ
Block height when Created: 117374 Unspent

Niffler


7:18

Back | Send

1.23456789 ツ ≈ $3.98

Network fee

ツ0.006 All the funds would be locked for around 10 min.

Message

Optional

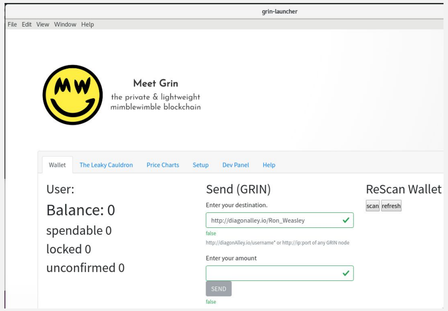
This message would be shown to a recipient, but would NOT be stored in the blockchain.

Send via?

[✓ File] ○ HTTP(S)

Send

Ironbelly


grin-launcher

File Edit View Window Help

Meet Grin
the private & lightweight
mimblewimble blockchain

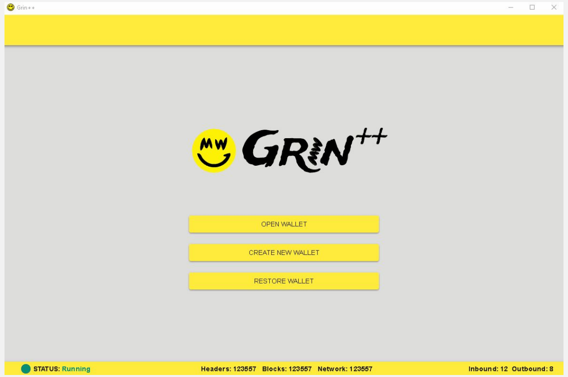
Wallet | The Leaky Cauldron | Price Charts | Setup | Dev Panel | Help

User:
Balance: 0
spendable 0
locked 0
unconfirmed 0

Send (GRIN)
Enter your destination.
http://diagonalley.io/Ron_Weasley
false
http://diagonAlley.io/username* or http://ip:port of any GRIN node
Enter your amount
SEND
false

ReScan Wallet
scan refresh

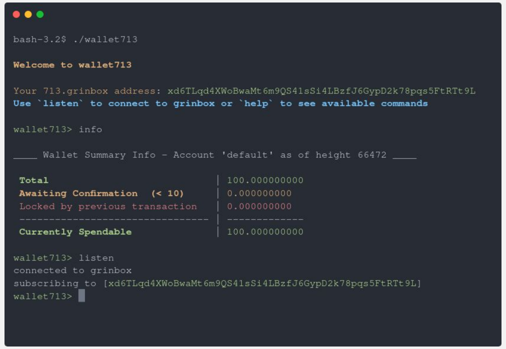
Diagon Alley


Grin++

OPEN WALLET
CREATE NEW WALLET
RESTORE WALLET

STATUS: Running | Headers: 123557 Blocks: 123557 Network: 123557 | Inbound: 12 Outbound: 8

Grin++



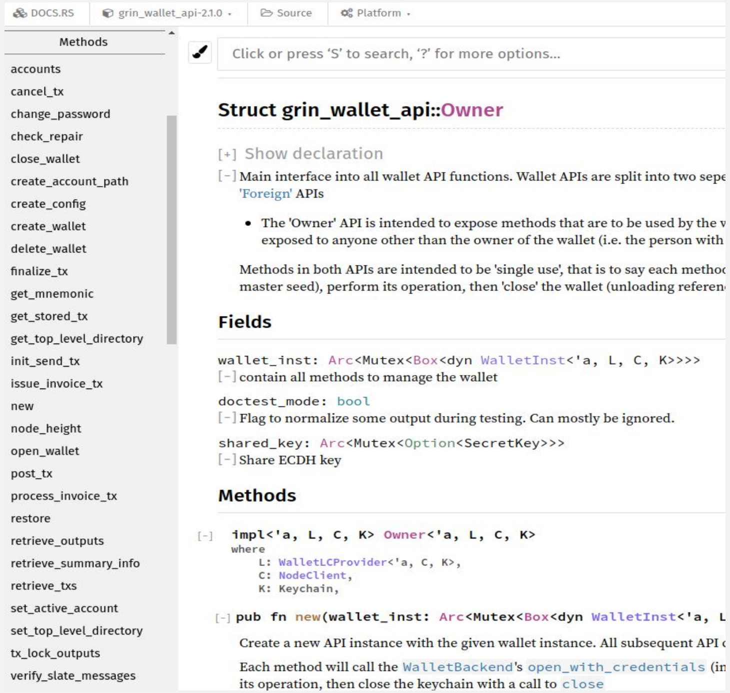
```
bash-3.2$ ./wallet713

Welcome to wallet713

Your 713.grinbox address: xd6TLqd4XWoBwaMt6m9QS41sSi4LBzfJ6GypD2k78pqs5FtRTt9L
Use `listen` to connect to grinbox or `help` to see available commands

wallet713> info

____ Wallet Summary Info - Account 'default' as of height 66472 ____

 Total                                | 100.000000000
 Awaiting Confirmation  (< 10)        | 0.000000000
 Locked by previous transaction       | 0.000000000
 ------------------------------------ | -------------
 Currently Spendable                  | 100.000000000

wallet713> listen
connected to grinbox
subscribing to [xd6TLqd4XWoBwaMt6m9QS41sSi4LBzfJ6GypD2k78pqs5FtRTt9L]
wallet713>
```

Wallet 713 (u no GUI)

Wallet Focus Since Launch

- Split wallet from Node
- Refine and solidify APIs
    - Provide Rust and secure JSON-RPC APIs
    - Refactor to provide ‘pluggable’ architecture
    - Full lifecycle management (init, recover, change password, etc)
    - Everything the command line wallet does can be done via the APIs
    - More robustness around wallet interoperability (Version check in API)

Wallet Focus Since Launch



DOCS.RS | grin_wallet_api-2.1.0 | Source | Platform

Methods

accounts
cancel_tx
change_password
check_repair
close_wallet
create_account_path
create_config
create_wallet
delete_wallet
finalize_tx
get_mnemonic
get_stored_tx
get_top_level_directory
init_send_tx
issue_invoice_tx
new
node_height
open_wallet
post_tx
process_invoice_tx
restore
retrieve_outputs
retrieve_summary_info
retrieve_txs
set_active_account
set_top_level_directory
tx_lock_outputs
verify_slate_messages

Click or press 'S' to search, '?' for more options...

Struct grin_wallet_api::Owner

[+] Show declaration

[−] Main interface into all wallet API functions. Wallet APIs are split into two sepe
'Foreign' APIs

- The 'Owner' API is intended to expose methods that are to be used by the w
exposed to anyone other than the owner of the wallet (i.e. the person with

Methods in both APIs are intended to be 'single use', that is to say each methoc
master seed), perform its operation, then 'close' the wallet (unloading referen

Fields

```
wallet_inst: Arc<Mutex<Box<dyn WalletInst<'a, L, C, K>>>>
```
[−] contain all methods to manage the wallet

```
doctest_mode: bool
```
[−] Flag to normalize some output during testing. Can mostly be ignored.

```
shared_key: Arc<Mutex<Option<SecretKey>>>
```
[−] Share ECDH key

Methods

```
[−] impl<'a, L, C, K> Owner<'a, L, C, K>
    where
        L: WalletLCProvider<'a, C, K>,
        C: NodeClient,
        K: Keychain,

[−] pub fn new(wallet_inst: Arc<Mutex<Box<dyn WalletInst<'a, L
```

Create a new API instance with the given wallet instance. All subsequent API c

Each method will call the WalletBackend's open_with_credentials (in
its operation, then close the keychain with a call to close

- Link Directly
- JSON-RPC from node.js
- Optionally include libs from Grin Node for all-in-1 solutions

Upcoming Features

- 3.0.0
  - Transaction exchange via TOR
  - Payment Proofs
  - Better state updates (UTXO scanning)
- Beyond
  - Let code base mature (to an extent)
  - Offline problem (grinbox? BBS? Something else?)
  - Better developer documentation, guides, etc
  - Better out-of-box command line experience

Things I’d like to Research

- BLS Signatures
    - As applied to transaction exchange + kernel aggregation
- Flyclient
- ‘Grinsecure’