

The State of the Lightning Network

Giacomo Zucco

The State of the Lightning Network

1. Unimportant Stuff
2. Important Stuff

1. Unimportant Stuff

1.1 Payment UX

1.2 Payment Privacy

1.3 Fund Security

1.4 Open Standards



2. Important Stuff

Non-stop Bikeshedding Over
Terminology, Acronyms, Symbols,
Icons, Fonts, Color Palettes

1. Unimportant Stuff

1.1 Payment UX

1.1 Payment UX


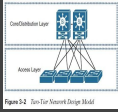
































1.1.1 L1/L2 Abstraction

1.1.2 L1/L2 Automation

1.1.3 Failure Management

1.1.4 Offline Management

1.1 Payment UX

		 <small>Figure 5-2 The Tor Network Design Model</small>		
				
				
				
				
				
				

1.1 Payment UX



1.1 Payment UX



Bitcoin Lightning Debate: LNURL vs BOLT 12

1.5K views • 11 months ago

1.1 Payment UX

ACINQ / [eclair](#) Public

[Code](#) [Issues](#) 22 [Pull requests](#) 20 [Discussions](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)

Add support for paying offers #2479

Merged

thomash-acinq merged 11 commits into `master` from `offer-pay` on Feb 16

lightningdevkit / [rust-lightning](#) Public

[Code](#) [Issues](#) 230 [Pull requests](#) 45 [Discussions](#) [Actions](#) [Projects](#) 6 [Wiki](#) [Security](#) [Insights](#)

BOLT 12 deserialization fuzzers #1977

Merged

TheBlueMatt merged 15 commits into `lightningdevkit:main` from `jkczyk:2023-01-offers-fuzz` last month

ElementsProject / [lightning](#) Public

[Code](#) [Issues](#) 433 [Pull requests](#) 44 [Actions](#) [Projects](#) 2 [Security](#) [Insights](#)

Offers compat fixes #5892

Merged

rustyrussell merged 10 commits into `ElementsProject:master` from `rustyrussell:guilt/offers-compat-fixes` on Jan 30

1.1 Payment UX



1.1 Payment UX



Unified QRs for Bitcoin

No more on-chain and lightning UI tabs. No more wallet interoperability issues. A simple, backwards-compatible way to request bitcoin for on-chain and lightning.

<https://bitcoinqr.dev/> (BIP21)

1.1 Payment UX

How to Payjoin

Scan a Unified QR Code

This is a BIP21 [unified URI](#) with a payjoin parameter. Even if a wallet does not support payjoin, it can still fall back to the address.



Raw Data

```
bitcoin:BC1QCVS6K3TPVLGN9HGTZ97LTX6GTPXVDMYYQPGMV?pj=HTTPS://BTCPAY.HRF.ORG/BTC/pj
```

<http://payjoin.org/>

1.1 Payment UX

ACINQ / eclair Public

Code Issues 22 Pull requests 20 Discussions Actions Projects Security Insights

Add initial support for async payment trampoline relay

Merged remyers merged 15 commits into `ACINQ:master` from `remyers:async-payments` on Sep 29, 2022

Conversation 67 Commits 15 Checks 1 Files changed 10

remyers commented on Sep 23, 2022

This is the first step in adding "async payments" as described in issue [#2424](#).

Add PeerReadyNotifier actor #2464

Merged t-bast merged 2 commits into `master` from `peer-ready-notifier` on Dec 21, 2022

Conversation 15 Commits 2 Checks 1 Files changed 10

t-bast commented on Oct 24, 2022

We add an actor that waits for a given peer to be connected and ready to process payments. This is useful in the context of async payments for the receiver's LSP.

@remyers you can start building the mechanism to trigger a pending async payment on top of that branch

1.1 Payment UX

[Lightning-dev] Async payments proof-of-payment: a wishlist for researchers

Anthony Towns [aj at erisian.com.au](mailto:aj@erisian.com.au)

Thu Jan 26 01:04:12 UTC 2023

- Previous message: [[Lightning-dev](#)] [Async payments proof-of-payment: a wishlist for researchers](#)
- Next message: [[Lightning-dev](#)] [Reputation Credentials renaming and iteration: the Staking_Credentials architecture](#)
- Messages sorted by: [[date](#)] [[thread](#)] [[subject](#)] [[author](#)]

On Tue, Jan 10, 2023 at 07:41:09PM +0000, vwallace via Lightning-dev wrote:

> *The open research question relates to how the sender will get an invoice from the receiver, given that they are offline at sending-time.*

Assuming the overall process is:

- * Alice sends a payment to Bob, who has provided a reusable address
AddrBob
- * Bob is offline at the time the payment is sent, but his semi-trusted
LSP Larry is online
- * Alice is willing/able to do bidirectional communication with Larry
- * The payment does not complete until Bob is online (at which point
Alice may be offline)

I think in this case you want to aim for the receipt to be a BIP340 signature of the message "Alice has paid me \$50 -- signed Bob".

Given Bob's public signature nonce, R, Alice (and Larry) can calculate $S = R + H(R,P,m)*P$ (m is the receipt message, P is Bob's public key), and then Alice can send a PTLC conditional on revealing the log of S, ie s where $s*G=S$; and at that point (s, R) is a valid signature by Bob of a message confirming payment to Bob, which then serves as the final receipt.

However for this to work, Alice needs to discover "R" while Bob is offline. I think this is only doable if Bob pre-generates a set of nonces and shares the public part with Larry, who can then share them with potential payers. I think to avoid attacks via Wagner's algorithm, you probably need to do a similar setup as musig2 does, ie share (R1,R2) pairs, and calculate $R = H(P,R1,R2,m)*R1+R2$.

1.1 Payment UX



Roy Sheinfeld

Oct 29, 2019 · 7 min read · Listen

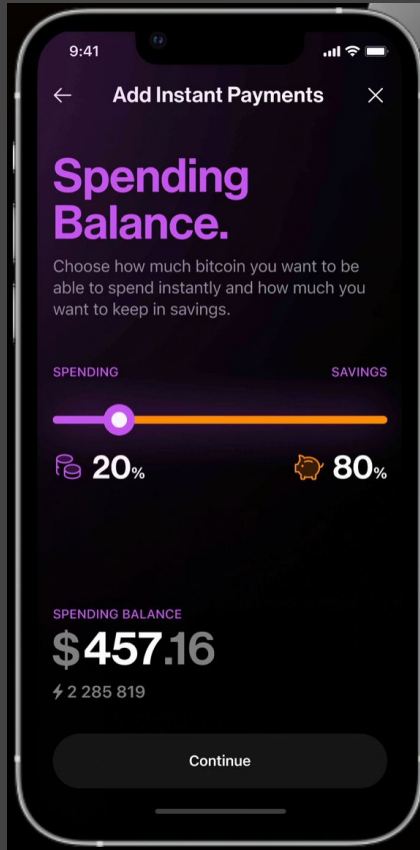


Introducing Lightning Rod

Asynchronous Lightning Payments for an On-Demand Culture



1.1 Payment UX



1. Unimportant Stuff

1.1 Payment UX

1.2 Payment Privacy

1.2 Payment Privacy

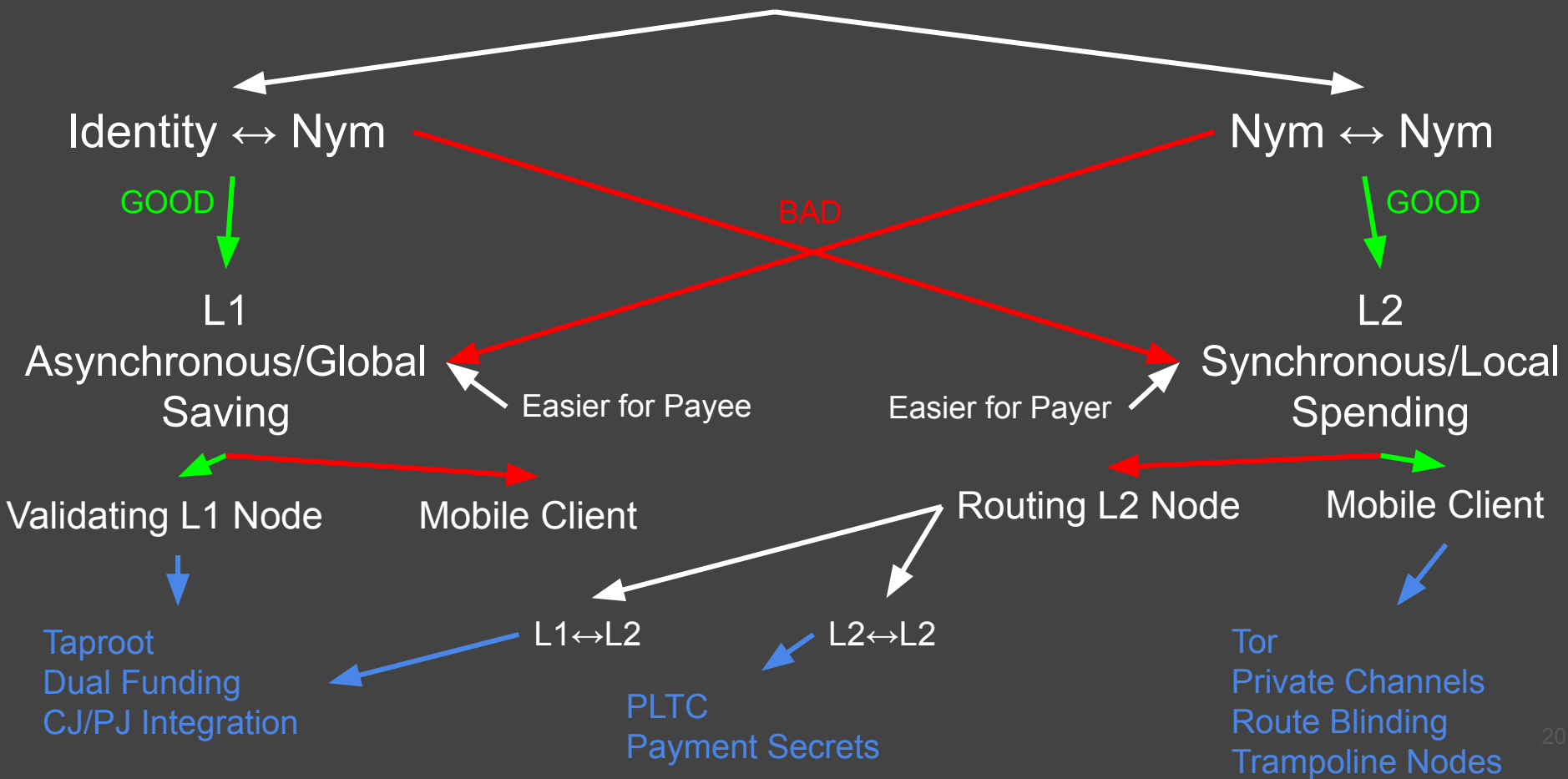
1.2.1 L2 vs L2 Privacy

1.2.2 L1 vs L2 Privacy

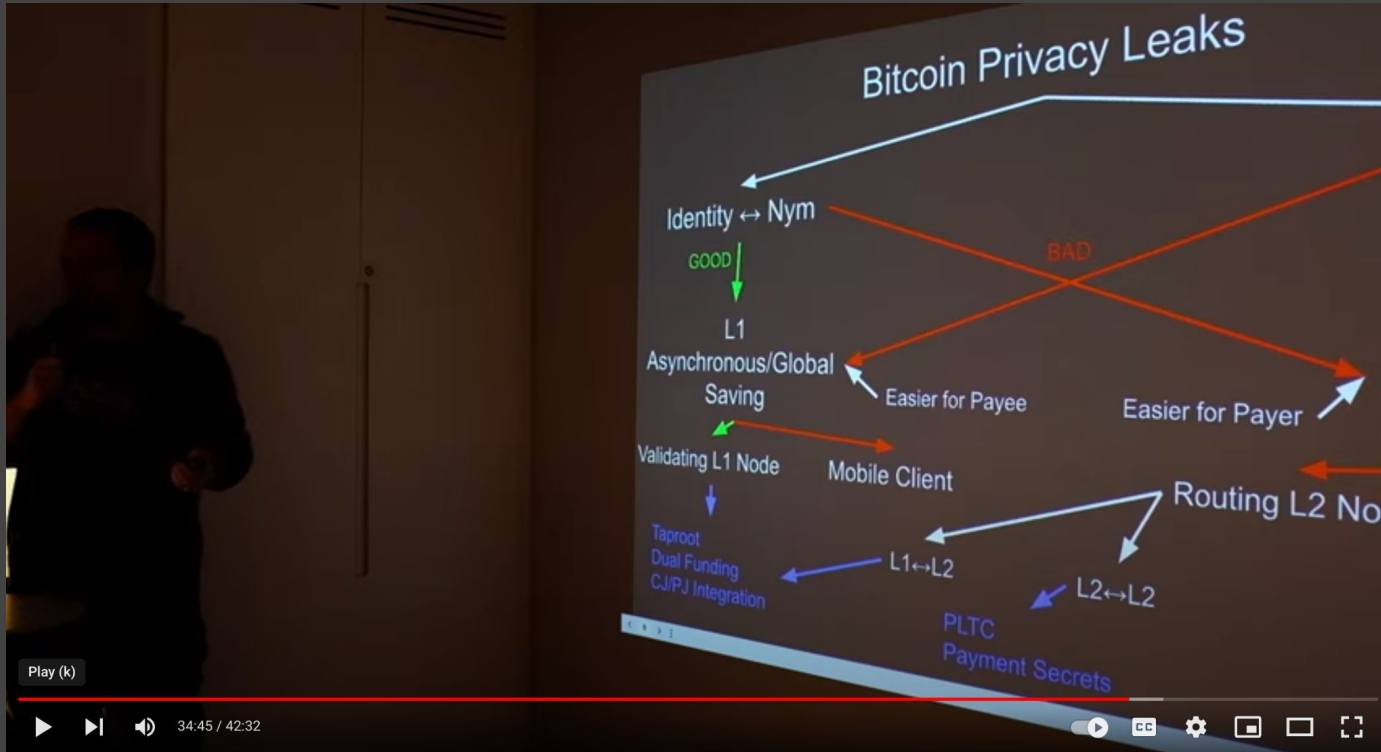
1.2.3 Receiving Privacy

1.2.4 Routing Privacy

Bitcoin Privacy Leaks



1.2 Payment Privacy



Lightning Hack Day Istanbul, February 26th 2022

Giacomo Zucco
855 subscribers

Analytics

Edit video

25

Share

Download

Clip

Save

...

1.2 Payment Privacy



RGB

- on shitcoins now
- on RGBTC someday?
- on BTC now

1.2 Payment Privacy

BIP21 Payment URIs with an optional lightning parameter

BIP-21 defines a URI scheme for creating a “payment link”. By default, it includes an on-chain address to send funds to.

BIP-21 was designed to be extensible. The spec allows for **optional parameters** in the URI. Why can't one of these parameters be used to include a BOLT 11 invoice, or even a BOLT 12 offer in the

future?

1.2 Payment Privacy

How to Payjoin

Scan a Unified QR Code

This is a BIP21 [unified URI](#) with a payjoin parameter. Even if a wallet does not support payjoin, it can still fall back to the address.



Raw Data

```
bitcoin:BC1QCVSR6K3TPVLGN9HGTZ97LTX6GTPXVDMYYQPGMV?pj=HTTPS://BTCPAY.HRF.ORG/BTC/pj
```


1.2 Payment Privacy

The image shows a vertical stack of three GitHub pull request cards. Each card displays the repository name, the pull request title, a 'Merged' badge, the merge action, and the merge date. The first card is for 'lightning / bolts' with pull request #765, 'Route Blinding (Feature 24/25)', merged 7 hours ago. The second card is for 'ACINQ / eclair' with pull request #2482, 'Send payments to blinded routes', merged on Dec 16, 2022. The third card is for 'ElementsProject / lightning' with pull request #44, 'Blinded payments spec update and infrastructure for forwarding', merged on Oct 26, 2022. The 'Pull requests' tab is highlighted in red in each repository's navigation bar.

lightning / bolts Public

Code Issues 71 Pull requests 66 Actions Projects 1 Wiki Security Insights

Route Blinding (Feature 24/25) #765

Merged t-bast merged 3 commits into master from route-blinding 7 hours ago

ACINQ / eclair Public

Code Issues 22 Pull requests 20 Discussions Actions Projects Security Insights

Send payments to blinded routes #2482

Merged t-bast merged 2 commits into master from send-to-blinded-route on Dec 16, 2022

ElementsProject / lightning Public

Code Issues 433 Pull requests 44 Actions Projects 2 Security Insights

Blinded payments spec update and infrastructure for forwarding

Merged rustyussell merged 27 commits into ElementsProject:master from rustyussell:blinded-payments-infra on Oct 26, 2022

1.2 Payment Privacy

ElementsProject / lightning Public

Code Issues 433 Pull requests 44 Actions Projects 2 Security Insights

dual funding -> API changes/breakage #5670

Merged rustyussell merged 19 commits into ElementsProject:master from niftynei:nifty/dual-fund-updates on Feb 4

ACINQ / eclair Public

Code Issues 22 Pull requests 20 Discussions Actions Projects Security Insights

Prepare InteractiveTxBuilder to support splicing #2595

Merged t-bast merged 5 commits into master from interactive-tx-splice-builder on Feb 17

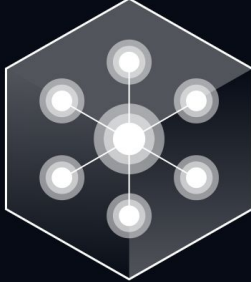
1.2 Payment Privacy

bitcoin / bitcoin Public Watch

Code Issues 370 Pull requests 309 Projects 7 Security Insights

I2P: add support for transient addresses for outbound connections #25355

Merged achow101 merged 7 commits into `bitcoin:master` from `vasild:i2p_transient_outbound_addr` on Aug 26, 2022



Delegated Staking

Delegate NYM to mix nodes that provide good quality of service, and earn a share of the rewards.

Onion messages

Onion messages are messages that can be sent across the LN network by nodes that support the protocol. Messages don't use HTLCs, minimizing the use of LN node resources.

1.2 Payment Privacy

- **LN summit 2021 notes:** Olaoluwa Osuntokun [posted](#) an extensive summary from the recent virtual and in-person LN developers meeting in Zurich. The summary includes notes about using [taproot](#) in LN, including [PTLCs](#), [MuSig2](#) for [multisignatures](#), and [eltoo](#); moving specification discussion from IRC to video chats; changes to the current BOLTs specification model; onion messages and [offers](#); stuckless payments (see [Newsletter #53](#)); [channel jamming attacks](#) and various mitigations; and [trampoline routing](#).

1. Unimportant Stuff

1.1 Payment UX

1.2 Payment Privacy

1.3 Fund Security

1.3 Fund Security

1.3.1 L1 Backup

1.3.2 L2 Backup

1.3.3 Watchtowers

1.3.4 L1/L2 Automation

1.3 Fund Security



STORM

1.3 Fund Security

- **LN summit 2021 notes:** Olaoluwa Osuntokun [posted](#) an extensive summary from the recent virtual and in-person LN developers meeting in Zurich. The summary includes notes about using [taproot](#) in LN, including [PTLCs](#), [MuSig2](#) for [multisignatures](#), and [eltoo](#); moving specification discussion from IRC to video chats; changes to the current BOLTs specification model; onion messages and [offers](#); stuckless payments (see [Newsletter #53](#)); [channel jamming attacks](#) and various mitigations; and [trampoline routing](#).

1.3 Fund Security

Bitcoin BIP-85 deterministic entropy: 10,000 seeds; one backup

Here are just a few ways to take advantage of this powerful function:

- Back up all your phone wallets with one seed
- Simplify key management for your organization
- Try out new wallets without having to create and track individual backups
- Make learner wallets for family members

1. Unimportant Stuff

1.1 Payment UX

1.2 Payment Privacy

1.3 Fund Security

1.4 Open Standards

1.4 Open Standards

1.3.1 Basic Specifications

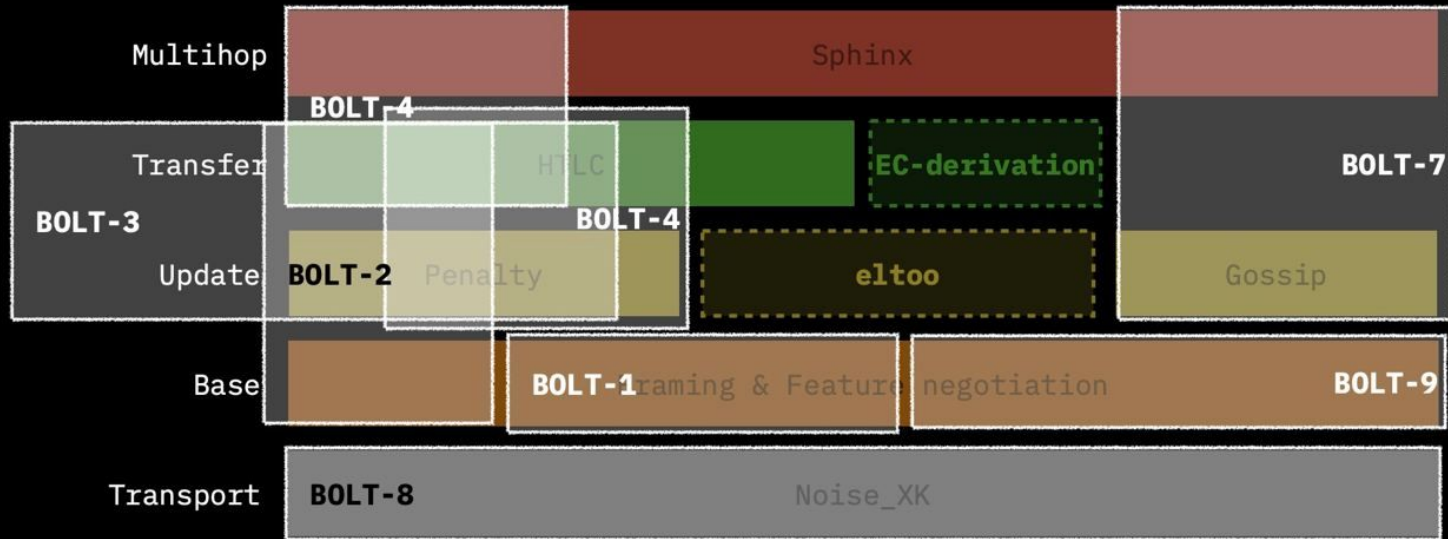
1.3.2 Advanced Specifications

1.3.3 Kits and Libraries

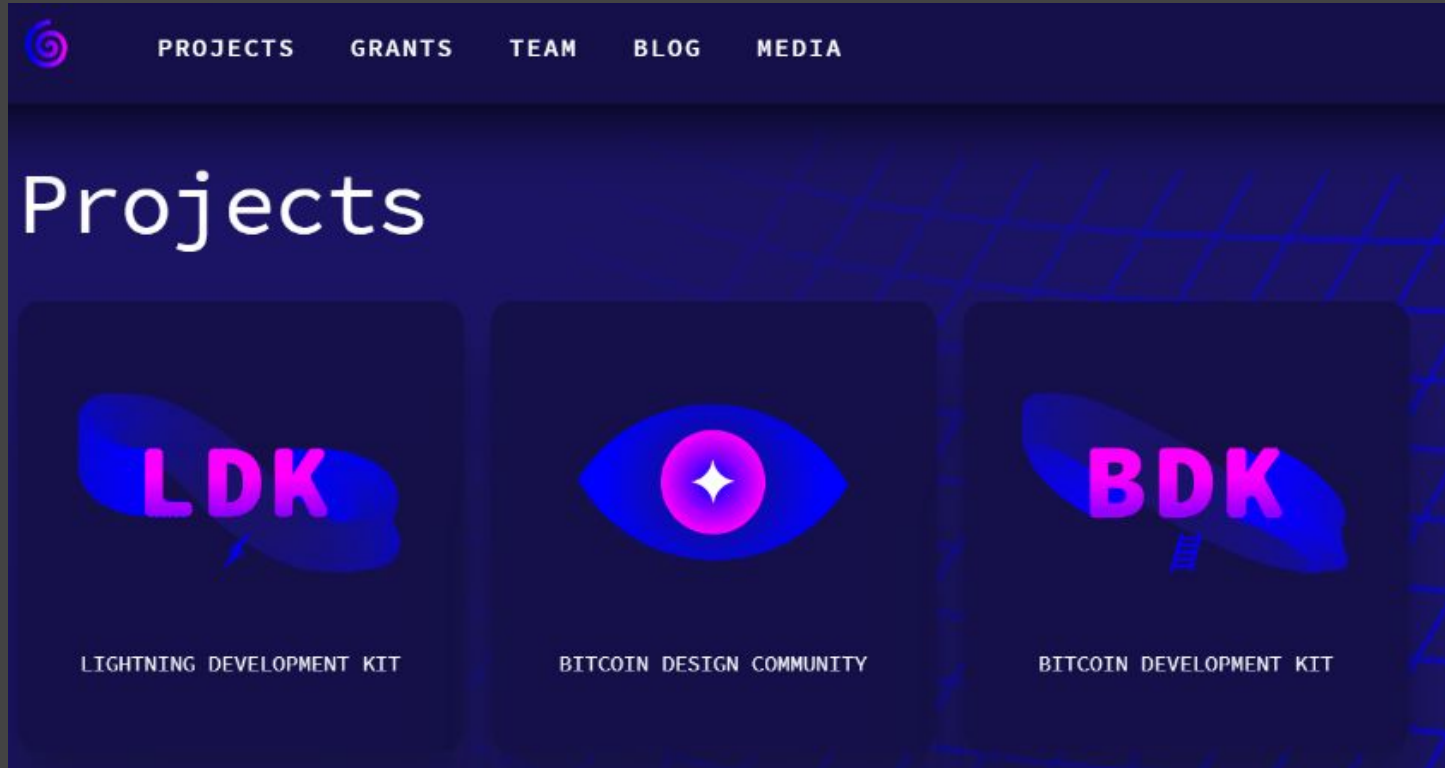
1.3.4 Turnkey Solutions

1.4 Open Standards

Real BOLT Specifications



1.4 Open Standards



1.4 Open Standards

LNP/BP Nodes



RGB Node

Smart contract
validation &
state RPC



BP Node

Bitcoin indexing
node (faster &
more efficient
than Electrum
server)



LNP Node

Lightning node
supporting RGB,
Taproot, DEX,
BiFi



Storm Node

Decentralized
storage,
messaging &
search

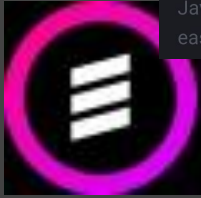
1.4 Open Standards



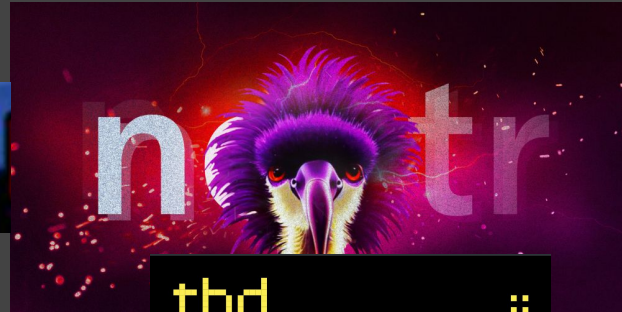
Holepunch

Introducing Hypercore

Holepunch provides a collection of *small Javascript modules* that can be combined to create an unlimited variety of P2P apps, from VPNs to communications tools like Keet. Mix and match them how you like. Since it's all just Javascript, developing a Holepunch app is as easy as building a web app.



Slashtags vs



tbd



**WEB5: AN EXTRA
DECENTRALIZED WEB
PLATFORM**

1.4 Open Standards



Roy Sheinfeld

Feb 9 · 8 min read ·  Listen



Lightning for Everyone in Any App: Lightning as a Service via the Breez SDK

Archimedes famously (is supposed to have) said: “Give me a lever long enough and a fulcrum on which to place it, and I shall move the world.”

While you’ve gotta love the bravado, the quote also reflects the simple point that the right technology at the right place and the right time *can* move the world. Steam power was nearly 2000 years old before the conditions were right for it to scale and change everything.

The State of the Lightning Network

1. Unimportant Stuff
2. Important Stuff

2. Important Stuff

Non-stop Bikeshedding Over
Terminology, Acronyms, Symbols,
Icons, Fonts, Color Palettes

2.1 Unit of Account

“0.00000001 BTC” doesn’t work

“1 SAT”

VS

“0.01 BITS”
(BIP 176)



2.2 SAT Symbol

SAT Symbol Design Options/Warz

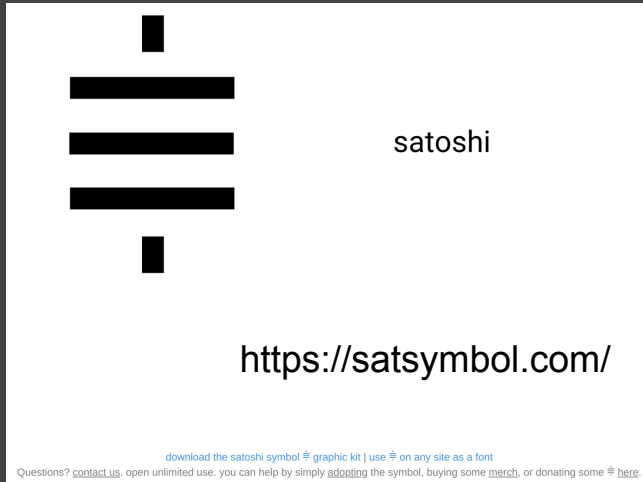
Community Suggested Options



A fraction of a bitcoin....

100M SAT's = 1 Bitcoin

2.2 SAT Symbol



VS



2.2 SAT Symbol



CULTURAL APPROPRIATION!!!!!!!







VS





2.3 LNP/BP vs "BTC/LN"



2.3 LNP/BP vs “BTC/LN”


    **Cart**

Catalogue  

LNP/BP: A gentle introduction

[Review](#) Updated • 21. july 2020

The Lightning Network is a relatively new and quickly developing system that promises cheap, fast and private payments on top of Bitcoin. It's already working in production, many people use it daily. But is that all? What if that was just the tip of some technological iceberg? What if what lies beneath the surface has a chance to significantly affect some parts of the Internet as we know it? Does that sound crazy? Maybe it is, but bear with me for a little while.



[LNP/BP \(EVERYTHING YOU WANTED TO KNOW\) – TABLE OF CONTENT](#)