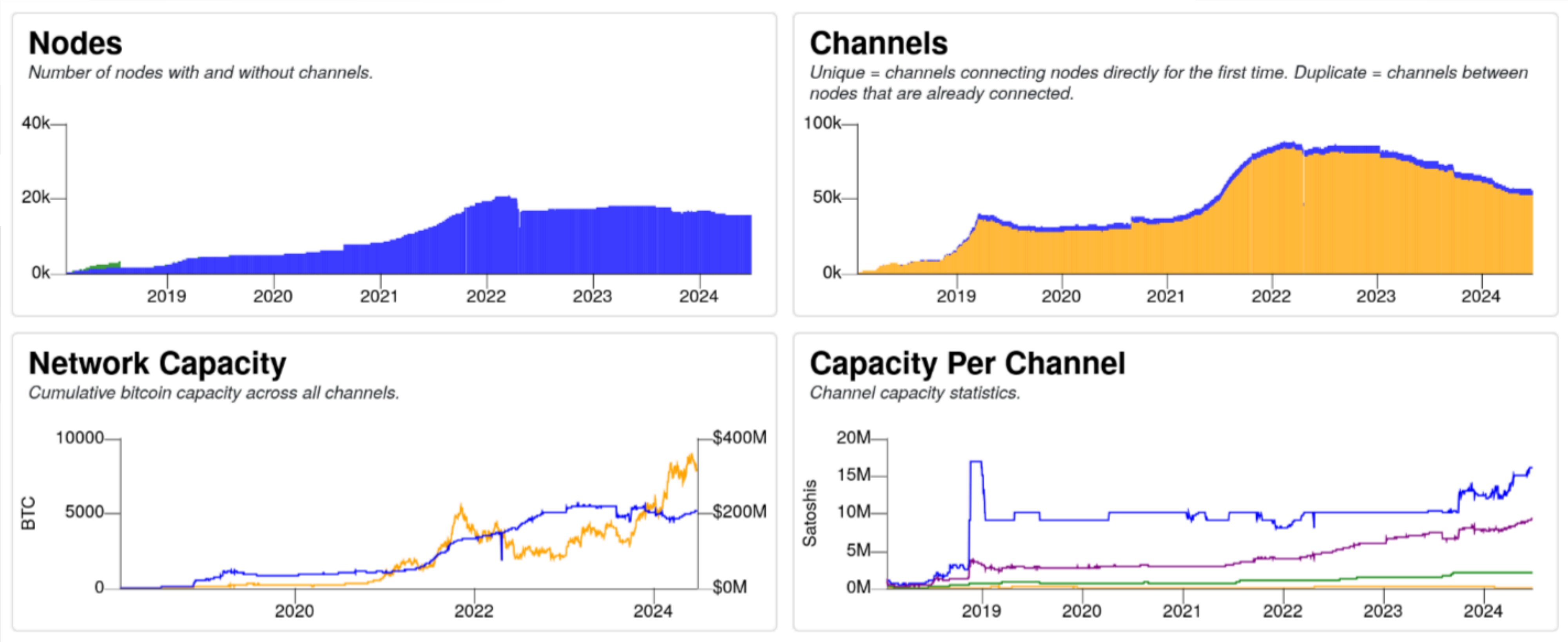
CKB Fiber Network

the best thing you may have never heard of about bitcoin lightning network

Mass adoption

Everyone is anticipating it, but can we live long enough to witness it?

2024, the year of BTC L2? All quiet on the LN front



Nodes

Number of nodes with and without channels.

Channels

Unique = channels connecting nodes directly for the first time. Duplicate = channels between nodes that are already connected.

Network Capacity

Cumulative bitcoin capacity across all channels.

Capacity Per Channel

Channel capacity statistics.

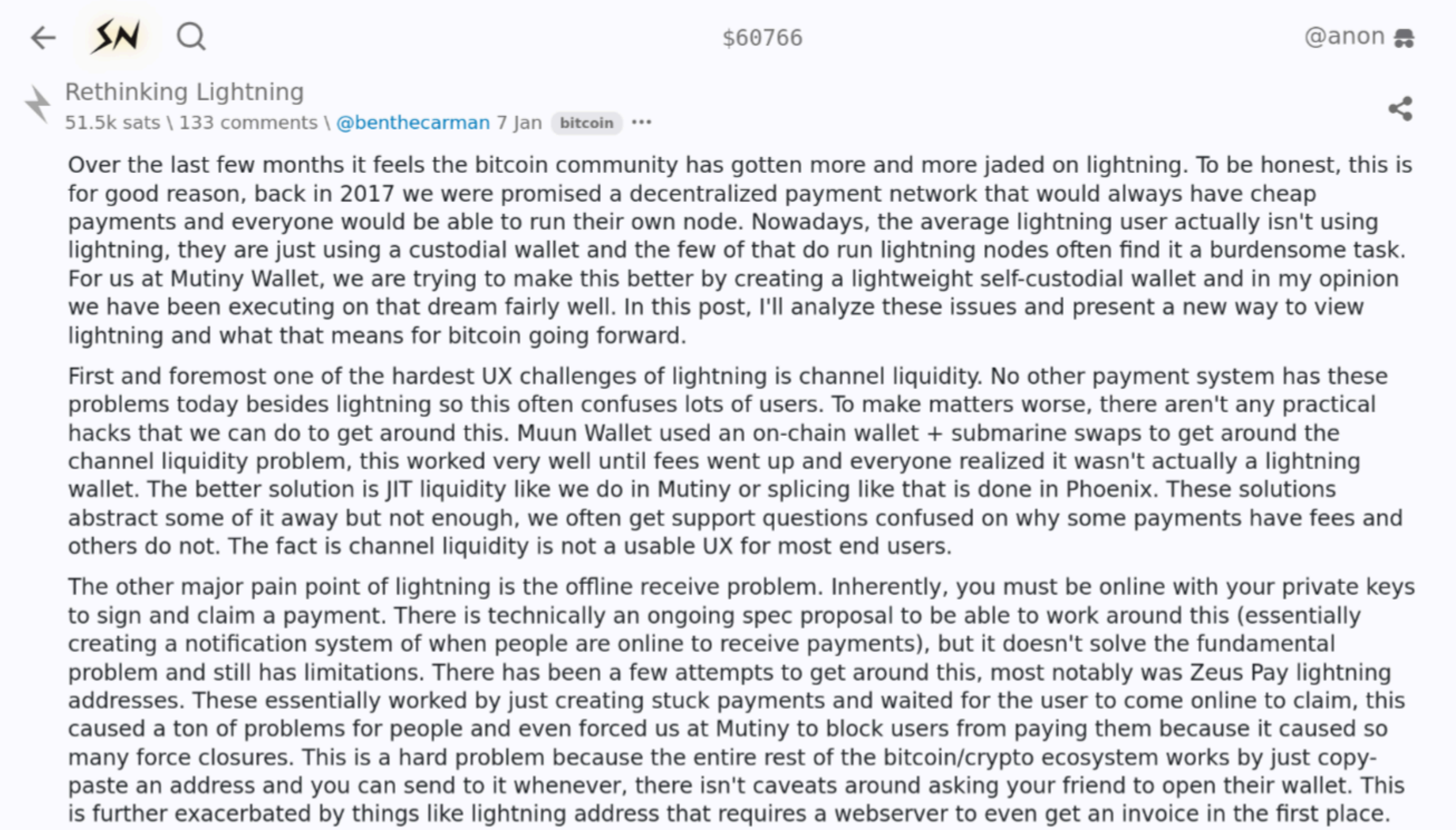
Source: https://bitcoinvisuals.com/lightning

PEOPLE ARE STILL RELUCTANT TO USE CRYPTO CURRENCY
UX OF CRYPTO IS TERRIBLE
EXACTLY WHO MADE THE UX SO BAD?

User experience of lightning network (or crypto currency in general)

Rethinking lightning

Source: https://stacker.news/items/379225



Rethinking Lightning

51.5k sats \ 133 comments \ @benthecarman 7 Jan bitcoin ···

Over the last few months it feels the bitcoin community has gotten more and more jaded on lightning. To be honest, this is for good reason, back in 2017 we were promised a decentralized payment network that would always have cheap payments and everyone would be able to run their own node. Nowadays, the average lightning user actually isn't using lightning, they are just using a custodial wallet and the few of that do run lightning nodes often find it a burdensome task. For us at Mutiny Wallet, we are trying to make this better by creating a lightweight self-custodial wallet and in my opinion we have been executing on that dream fairly well. In this post, I'll analyze these issues and present a new way to view lightning and what that means for bitcoin going forward.

First and foremost one of the hardest UX challenges of lightning is channel liquidity. No other payment system has these problems today besides lightning so this often confuses lots of users. To make matters worse, there aren't any practical hacks that we can do to get around this. Muun Wallet used an on-chain wallet + submarine swaps to get around the channel liquidity problem, this worked very well until fees went up and everyone realized it wasn't actually a lightning wallet. The better solution is JIT liquidity like we do in Mutiny or splicing like that is done in Phoenix. These solutions abstract some of it away but not enough, we often get support questions confused on why some payments have fees and others do not. The fact is channel liquidity is not a usable UX for most end users.

The other major pain point of lightning is the offline receive problem. Inherently, you must be online with your private keys to sign and claim a payment. There is technically an ongoing spec proposal to be able to work around this (essentially creating a notification system of when people are online to receive payments), but it doesn't solve the fundamental problem and still has limitations. There has been a few attempts to get around this, most notably was Zeus Pay lightning addresses. These essentially worked by just creating stuck payments and waited for the user to come online to claim, this caused a ton of problems for people and even forced us at Mutiny to block users from paying them because it caused so many force closures. This is a hard problem because the entire rest of the bitcoin/crypto ecosystem works by just copy-paste an address and you can send to it whenever, there isn't caveats around asking your friend to open their wallet. This is further exacerbated by things like lightning address that requires a webserver to even get an invoice in the first place.

> “ Nowadays, the average lightning user actually isn't using lightning. ”

> “ First and foremost one of the hardest UX challenges of lightning is channel liquidity. ”

> “ The other major pain point of lightning is the offline receive problem. ”

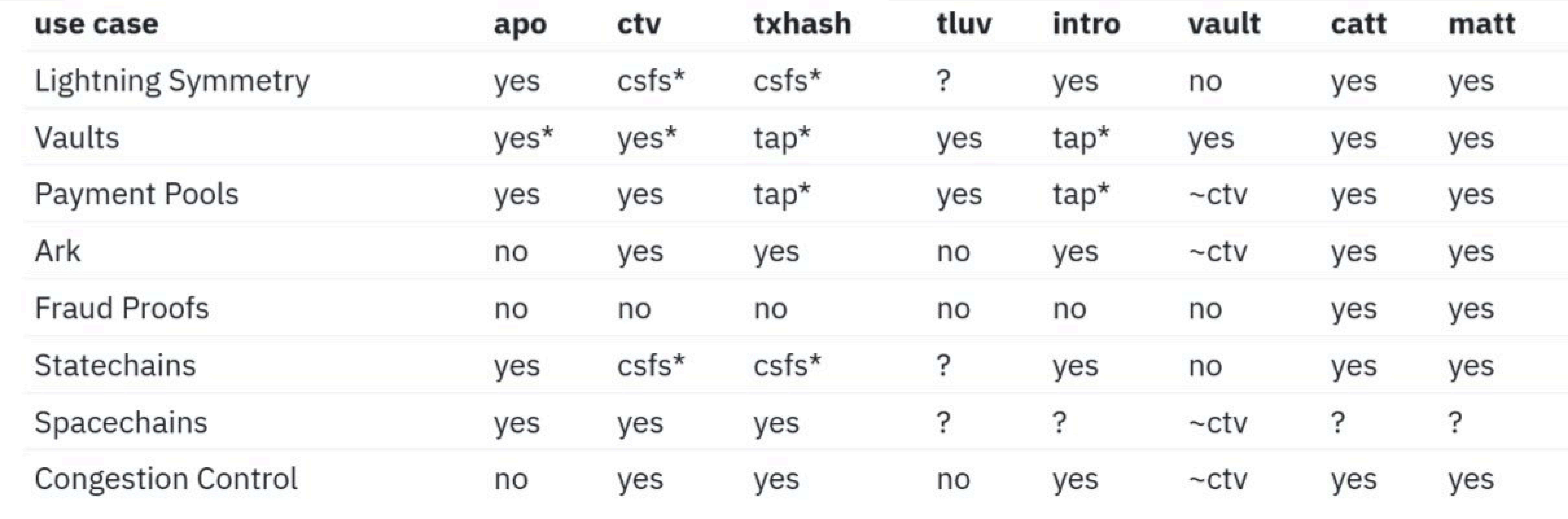
> “ Combining existing large scale lightning infrastructure with self-custodial solutions sadly, isn't totally possible. ”

> “ So how do we scale ownership? Simply put, the answer today is custody. ”

> “ Are we doomed then? Is there no way to scale bitcoin in a self-sovereign way? Luckily, the answer is no, but we need some soft-forks. Covenants are the way to scale bitcoin ownership. ”

What covenants can do?

Source: https://covenants.info/overview/summary/



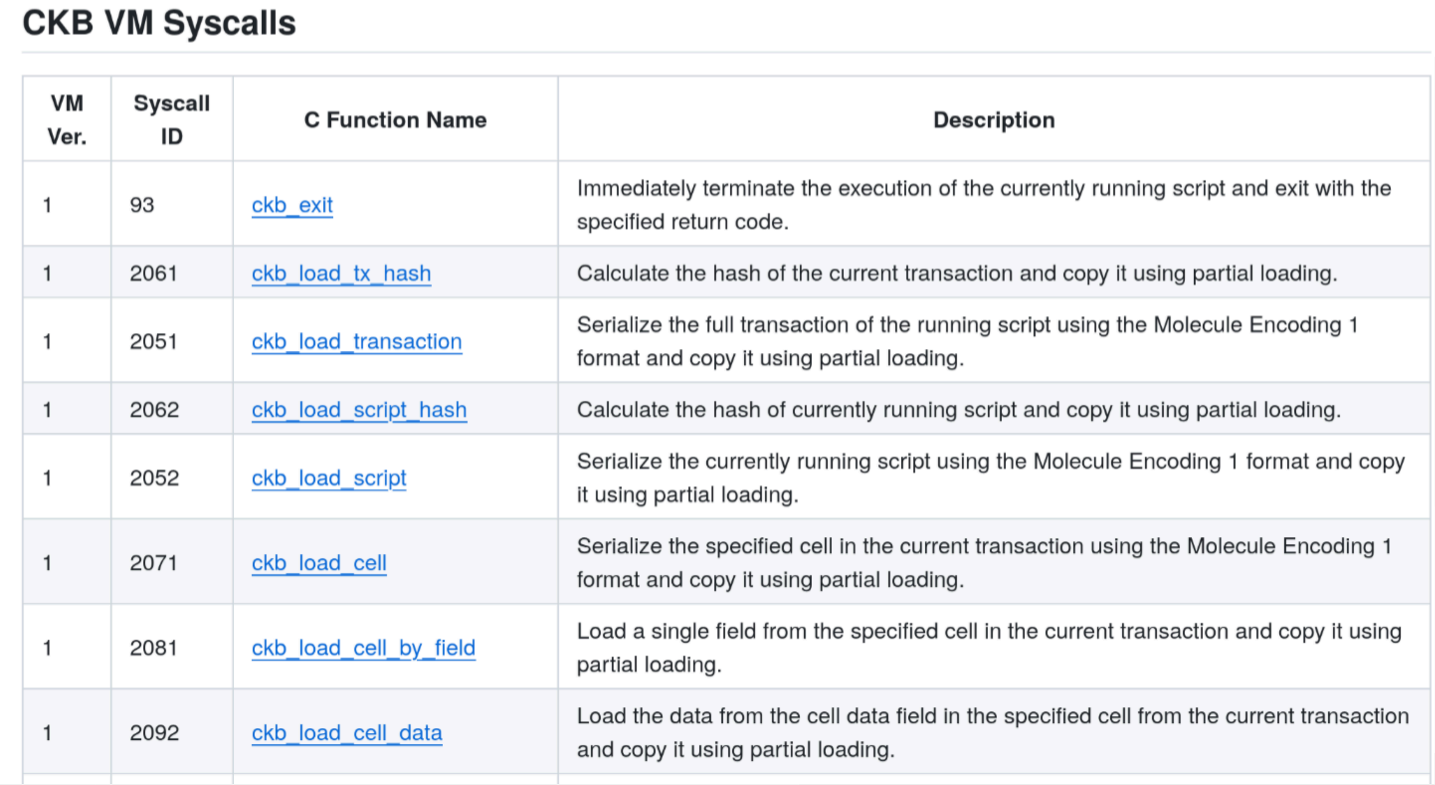
use case	apo	ctv	txhash	tluv	intro	vault	catt	matt
Lightning Symmetry	yes	csfs*	csfs*	?	yes	no	yes	yes
Vaults	yes*	yes*	tap*	yes	tap*	yes	yes	yes
Payment Pools	yes	yes	tap*	yes	tap*	~ctv	yes	yes
Ark	no	yes	yes	no	yes	~ctv	yes	yes
Fraud Proofs	no	no	no	no	no	no	yes	yes
Statechains	yes	csfs*	csfs*	?	yes	no	yes	yes
Spacechains	yes	yes	yes	?	?	~ctv	?	?
Congestion Control	no	yes	yes	no	yes	~ctv	yes	yes

MOST OF LN'S USABILITY PROBLEMS CAN BE SOLVED WITH COVENANTS
GREAT, WHEN CAN WE USE IT?
EXACTLY WHEN CAN WE USE IT?

When can we use covenants on BTC?

Join BTC by CKB

Wait, does CKB have covenants already?

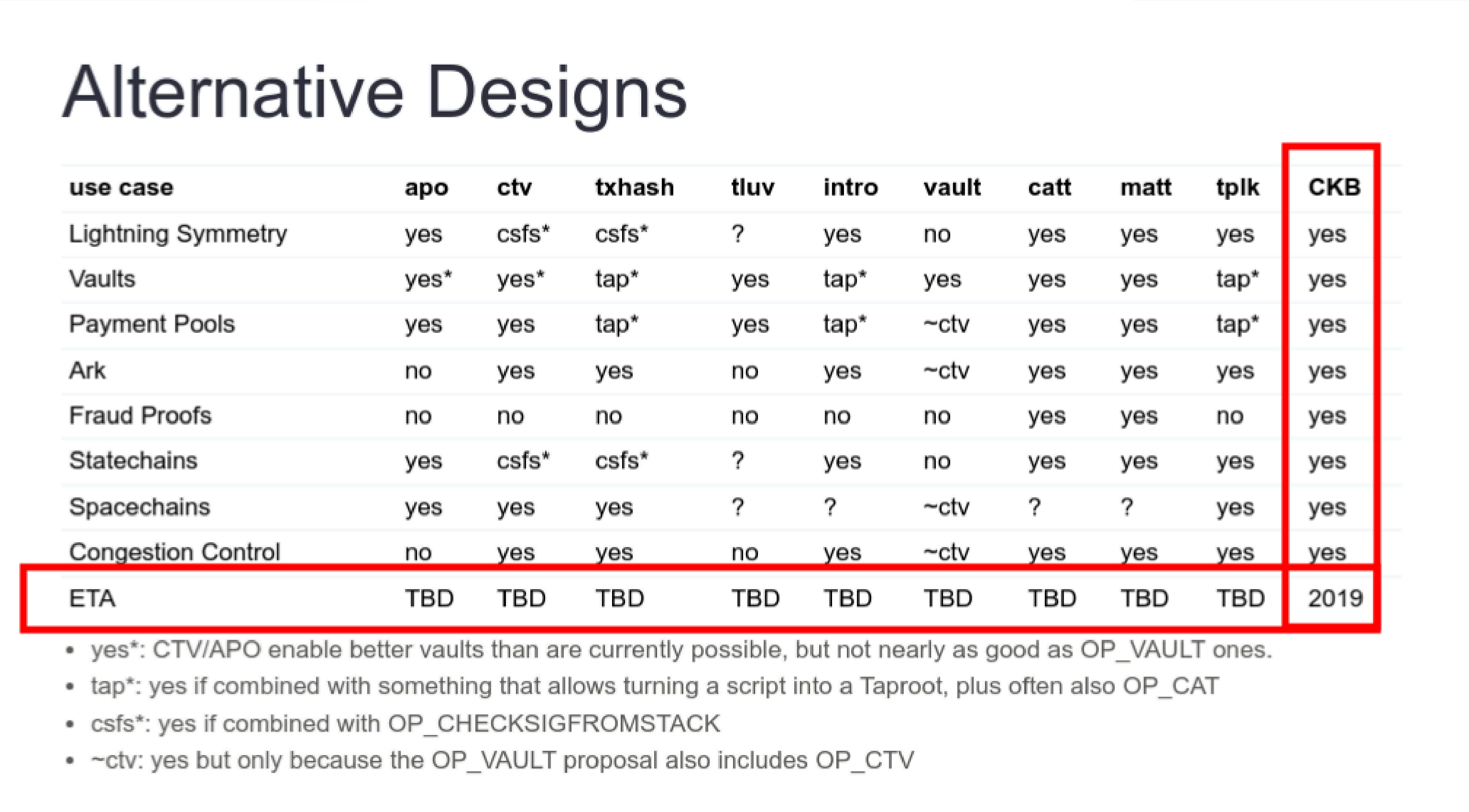


CKB VM Syscalls

VM Ver.	Syscall ID	C Function Name	Description
1	93	ckb_exit	Immediately terminate the execution of the currently running script and exit with the specified return code.
1	2061	ckb_load_tx_hash	Calculate the hash of the current transaction and copy it using partial loading.
1	2051	ckb_load_transaction	Serialize the full transaction of the running script using the Molecule Encoding 1 format and copy it using partial loading.
1	2062	ckb_load_script_hash	Calculate the hash of currently running script and copy it using partial loading.
1	2052	ckb_load_script	Serialize the currently running script using the Molecule Encoding 1 format and copy it using partial loading.
1	2071	ckb_load_cell	Serialize the specified cell in the current transaction using the Molecule Encoding 1 format and copy it using partial loading.
1	2081	ckb_load_cell_by_field	Load a single field from the specified cell in the current transaction and copy it using partial loading.
1	2092	ckb_load_cell_data	Load the data from the cell data field in the specified cell from the current transaction and copy it using partial loading.

They have always been there. Just too trivial to give a dedicated term.

And can CKB do that?



Alternative Designs

use case	apo	ctv	txhash	tluv	intro	vault	catt	matt	tplk	CKB
Lightning Symmetry	yes	csfs*	csfs*	?	yes	no	yes	yes	yes	yes
Vaults	yes*	yes*	tap*	yes	tap*	yes	yes	yes	tap*	yes
Payment Pools	yes	yes	tap*	yes	tap*	~ctv	yes	yes	tap*	yes
Ark	no	yes	yes	no	yes	~ctv	yes	yes	yes	yes
Fraud Proofs	no	no	no	no	no	no	yes	yes	no	yes
Statechains	yes	csfs*	csfs*	?	yes	no	yes	yes	yes	yes
Spacechains	yes	yes	yes	?	?	~ctv	?	?	yes	yes
Congestion Control	no	yes	yes	no	yes	~ctv	yes	yes	yes	yes
ETA	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	2019

- yes*: CTV/APO enable better vaults than are currently possible, but not nearly as good as OP_VAULT ones.
- tap*: yes if combined with something that allows turning a script into a Taproot, plus often also OP_CAT
- csfs*: yes if combined with OP_CHECKSIGFROMSTACK
- ~ctv: yes but only because the OP_VAULT proposal also includes OP_CTV

Request for fact-checking

- You are welcome fact-check my hasty conclusion above (it's backed by only over-confidence).
- I will not fix any inaccuracy in my slides, as CKB is easily fixable.

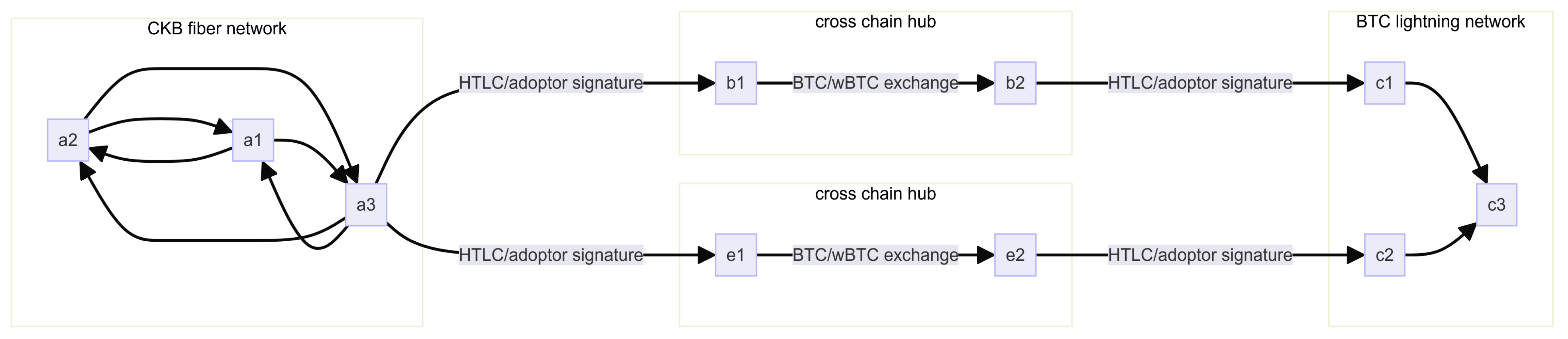
Introducing CKB Fiber Network (CFN)

```
    BTC      L1
   /   \
  /     \
LN      CKB  L2
  \     /
   \   /
    CFN    L2+L2=L3 or L4?
```

- LN: Instant, Infinitely Scalable P2P Payment System
- CKB: Unmatched Flexibility and Interoperability

Call this L_∞ instead of L_3 or L_4.

High level overview of CFN



- Same building blocks as lightning network (HTLC and revocation)
- Native multiple assets support (extremely versatile thanks to xUDT's extensibility)
- Cross-chain payment channel network (available now, made only possible by CKB-VM's flexibility)

Demo time

Too bad. We only have time to show some staged animations.

TODO: show some testnet transaction screenshots on the explorer websites.

Conclusion

CFN as of today

- Same security assumption as bitcoin lightning network
- Native multi-assets payment channel network
- Referenece implementation is now available with BTC cross chain support (WARNING: demonstratable only for now, lots for bugs to be squashed)
- Almost all the functionalities mentioned above have their repective RPC ready for integration

CFN as of tomorrow

- Achieve feature parity with bitcoin lightning network (watch tower, multiple-hop network)
- Rethink payment channel network with CKB's extensibility and programmbility
  - State channels with smart contract support
  - Highly-articipated lightning network features made possible by covenants (e.g. Non Interactive Channels)
  - And beyond

Join the force

Come and build.
Life is too short
for all the nicest BIPs
to land.

- https://github.com/nervosnetwork/cfn-node
- https://github.com/nervosnetwork/cfn-scripts