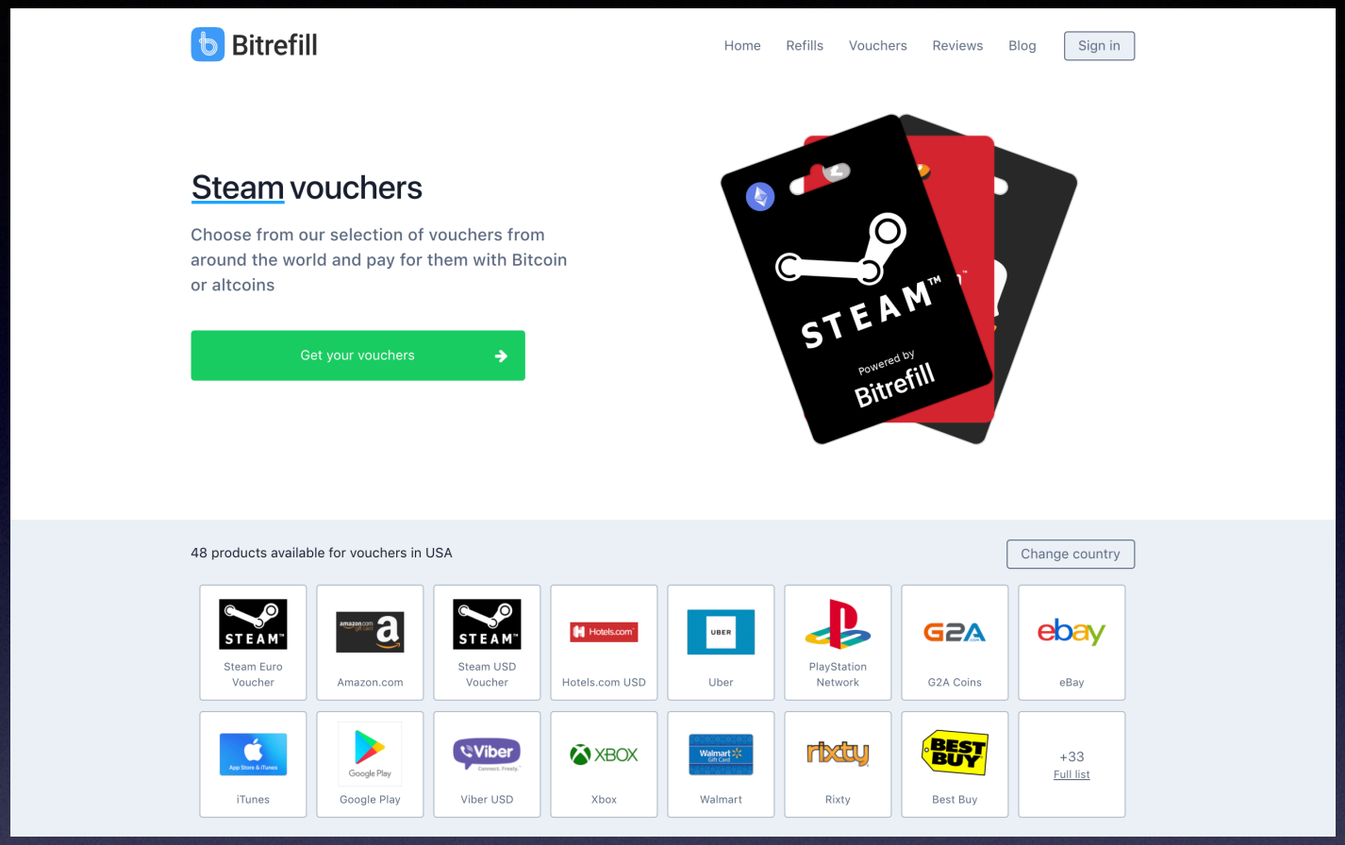
# Integrating Lightning Into Bitrefill

Justin Camarena

Chaincode Lightning Residency 2018

Bitrefill

# Bitrefill

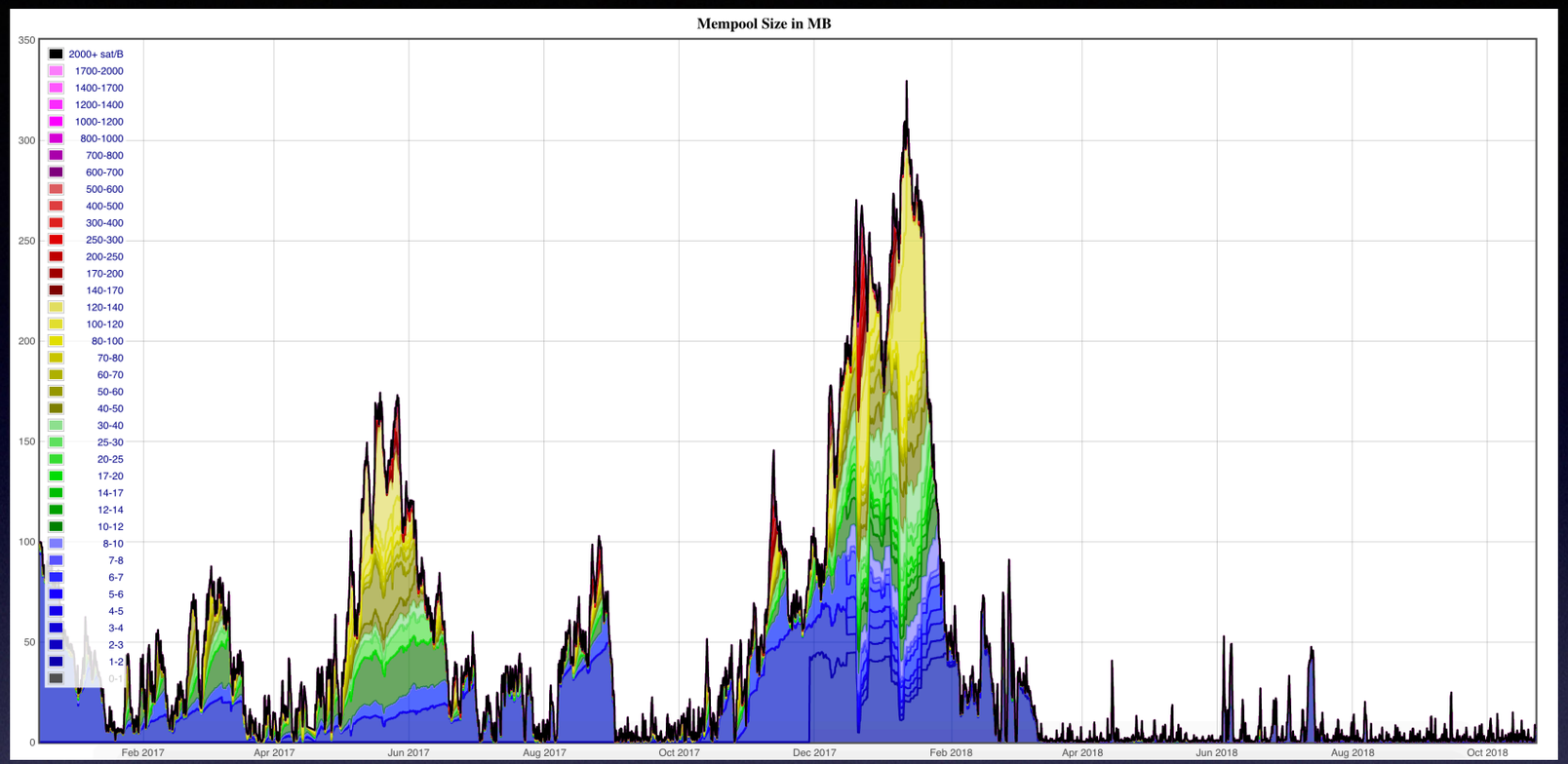



- Originally a prepaid phone credits seller available  in 107+ countries
- We were bitcoin only for many years founded in 2014
- We have recently expanded into other products such as gift cards

# The Path to Lightning Integration

- Originally joined Bitrefill as an intern a little more than a year to remove Bitrefill's dependence on Third-party bitcoin APIs
- Depend on our own bitcoind node as our source of truth for bitcoin orders
- Move refunds away from Copay
- Similar to btcpay server

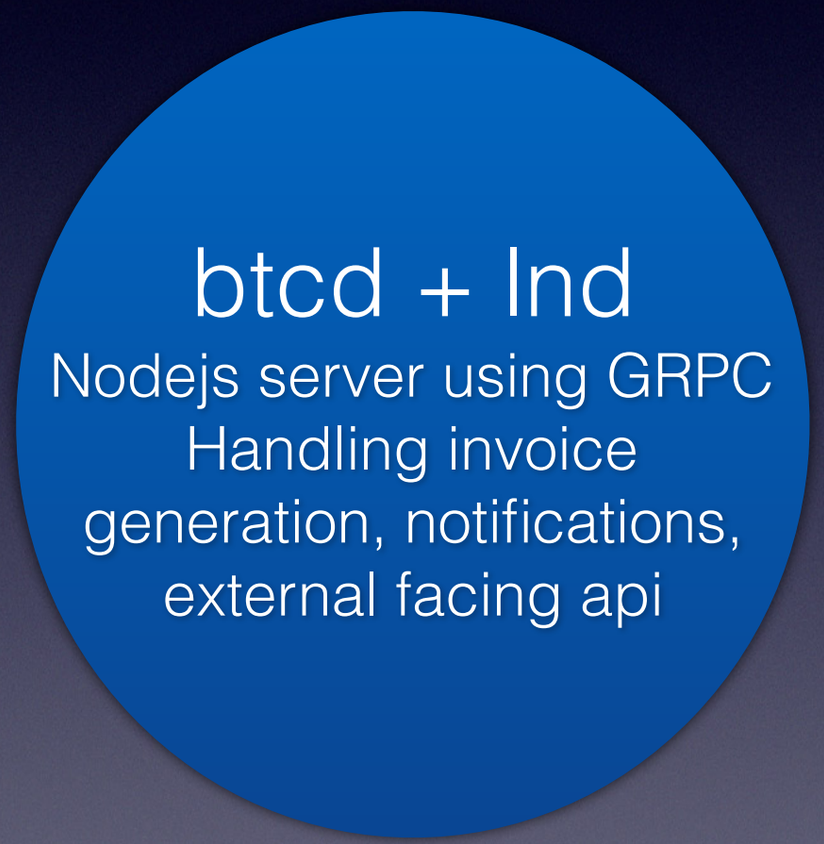
# Scaling Issues




- Merchants were hit quite hard due to fees spikes and backlogs
- Bitrefill was no exception especially due to having many low value products in some countries
- Centralized Approach: Accounts, zero fees, although we hold custody of funds
- Decentralized Approach: Being the first merchants to upgrade to segwit addresses for orders
- Accepting altcoins (Not a long term solution, low volume compared to bitcoin)
- Integrating lightning

# Actually Integrating Lightning

- Originally our backend went with BTCD + LND
- Why LND?
- Nodejs backend handling interaction with LND with GRPC
- Reused much of the work done to not depend on third-party bitcoin apis



# Actually Integrating Lightning

- Application Backend calls out to our bitcoin backend api server when generating a lightning order or deposit invoice
- Lightning Invoices: addInvoice  Params: memo, satoshi amount, expiry
- Memo: ‘Bitrefill 5bcfdce8e3a4580004abaf7e’

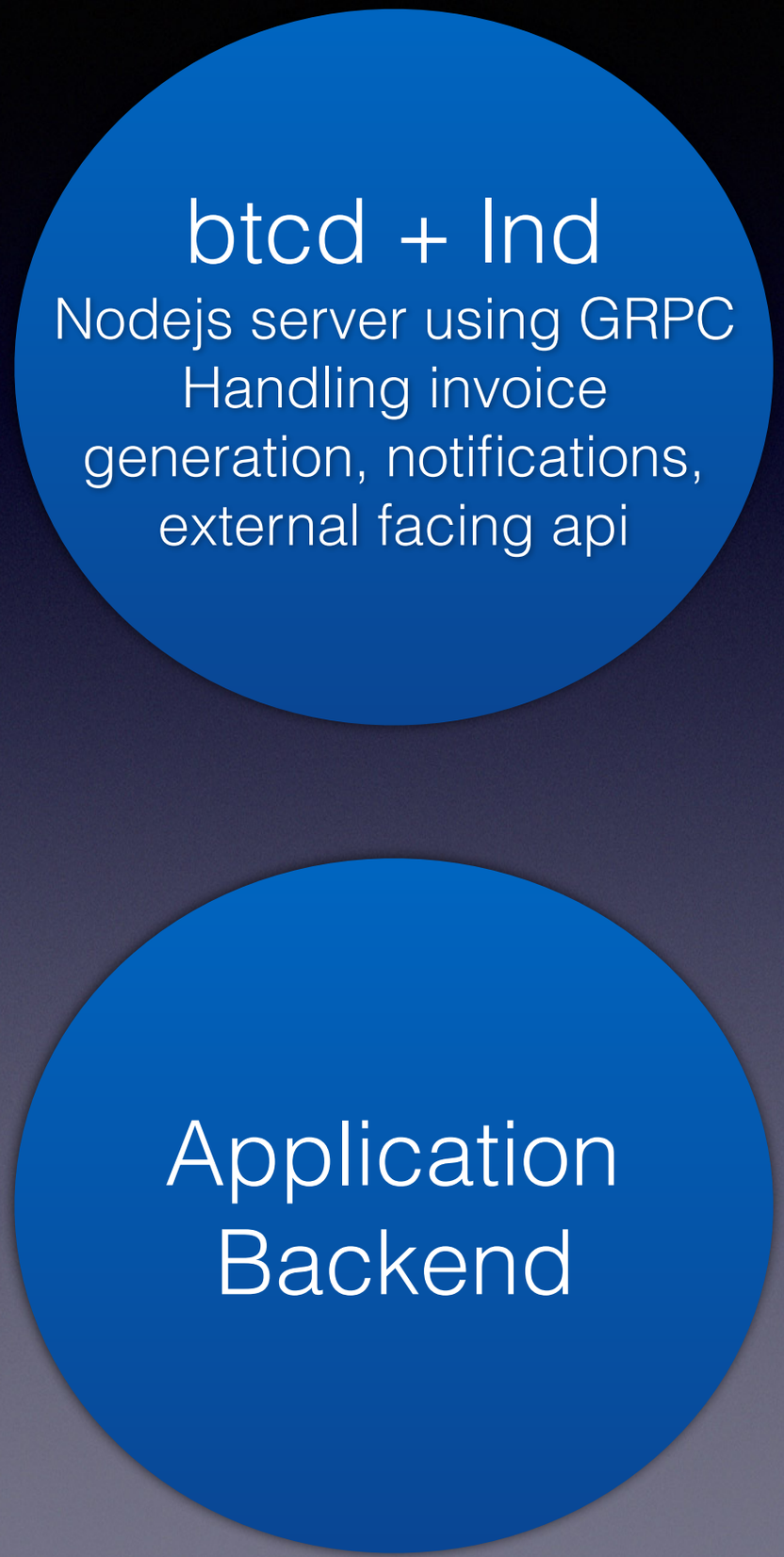
btcd + lnd
Nodejs server using GRPC
Handling invoice generation, notifications, external facing api

- Application backend keeps state of orders being paid, refunded
- Expiration is handled on the lightning end avoid which is great to avoid payments after expiration

Application Backend

# Actually Integrating Lightning

- SuscribeInvoice on our bitcoin backend service, send relevant notifications to our application backend resend notifications using listinvoices
- Important details I pass to our application backend web hook notification endpoint: memo, amount, settled, amt_paid_sat
- Grab order id from memo, verify amount paid matches amount set in application backend



# Deploying Lightning

- Originally testnet only
- This was before beta and shortly after segwit activated I had a local version with working lightning payments in August 2017
- Great for finding bugs with users testing on Bitrefill
- Experience Bitrefill with a real Merchant with fake refills
- Mainnet Launch March 15 to the public

# Challenges Integrating Lightning

- Stability
- Bugs
- Money Loss on Testnet
- Wallet Issues out of state (Ghost UTXOs w LND)
- Early tech with bugs expected
- Interoperability issues (channels force closing due to protocol disagreements, or fees) 150 153? ¯_(ツ)_/¯
- Three Lightning Networks for some time: Eclair Lightning, c-lightning, lnd lightning
- Little to no liquidity between different node implementations

# Lighting Nodes

- Take of merchants not being routing nodes
- Many devs believe Merchants should have private incoming channels

- Bitrefill is a routing node
- Merchants online already run 24/7, suited to route at the moment
- Centralized due to not having nodes freely able to add capacity
- At the will of other nodes raising routing fees to the merchant
- Our view currently, long term it may be better to have private incoming channels

# Liquidity Lightning

- Incoming capacity is not a problem, if anything we have more than enough incoming capacity
- If a service has demand and usage nodes will open channels and provide liquidity to earn fees
- Issues offloading funds due to not many if any exchanges supporting lightning deposits
- Avoid close user funded channels to avoid potential user having a bad experience, may also be another reason long term why merchants should avoid

- Have sweeped funds off our node many times in cases of users remotely closing channels, capacity is always fluctuating up and down

# Avoiding Dust UTXO

- Minimum incoming channel 0.01 bitcoin has helped avoid small channels that could more likely close with small local balances
- Can not actually avoid small utxos

- A Minimum helps limit the amount of channels incoming to Bitrefill which limits potential utxo count
- This also limits number of peers with better performance due to this

# Integrating Lightning Payments

- Focus on receiving lighting payments, and refunding on-chain initially

- Stealing funds via routing fees with refunds
- Receiving is easy, sending is hard and very dangerous!

# Fee Siphoning Attack

Attacker pays for service that fails and requires a refund


200000 satoshi payment from node 1

Each channel is at max capacity, preferably with more incoming capacity toward the attacker’s node 1

# Fee Siphoning Attack

Service must offer automatic refunds or offer sending from a service without paying routing fees


Bitrefill refunds 200000 satoshi payment with attacker directing it to node 2
Attacker raises routing fee rates and provides the only liquidity to node 2, node 2 presumably rejects any other nodes from creating channels to it

# Fee Siphoning Attack

- You must then repeat this scenario over and over, larger amounts yield bigger fees
- A well connected node 1 with plenty of incoming funds increase fund loss potential
- Can use channel opening services to obtain incoming capacity

# Mitigations

- Rate limit refunds to nodes but may cause problems for those using a shared custodial node
- Fee limits do not solve this issue only limit fund loss to a slower rate
- Must pass on routing fees to the user, refund x satoshis less and use that for routing fees
- Before any service/exchange integrate sending lightning, implement fee limiting
- Help create a best practices for sending lightning payments

# Other Work

- Automatically restarting LND after crashes with supervisord,
- Migrating to a new node using bitcoind instead of btcd
- Migration strategy liquidity wise moving to anew node
- Automatic wallet unlock using unlockWallet grpc endpoint
- Settled Index, resubmitting notifications the right way
- Moving away from a node running since beta, and having it as a backup
- Lightning Deposits
- Lightning Withdrawals and other issues sending via lightning

# There’s hope!

- Even with all the issues I have had most of these issues can be fixed or mitigated
- Many of them have already been fixed or are being worked on right now
- I’d rather run into these issues myself before others do at scale
- Lightning will only get better with recent improvement proposals and maturity of documentation and new developers being onboard