The Lightning Network: More than just payments

William Casarin jb55@jb55.com, nostr:
32e1827635450ebb3c5a7d12c1f8e7b2
b514439ac10a67eef3d9fd9c5c68e245

June 16, 2022

Overview

- The lightning network at the base level is an ultra-secure way of sending information between two computers. It is more secure than how we access the web today.

Overview

- The lightning network at the base level is an ultra-secure way of sending information between two computers. It is more secure than how we access the web today.
- How can we utilize this to build interesting apps that can talk to lightning and bitcoin nodes directly.

Overview

- The lightning network at the base level is an ultra-secure way of sending information between two computers. It is more secure than how we access the web today.
- How can we utilize this to build interesting apps that can talk to lightning and bitcoin nodes directly.
- Why would you want to do this, and what kinds of applications are possible.

Internet communication in the modern age: The Web

- Typically internet commerce is done through the world wide web, which is a suite of protocols such as HTTP, TLS, etc

Internet communication in the modern age: The Web

- Typically internet commerce is done through the world wide web, which is a suite of protocols such as HTTP, TLS, etc
- There's a problem: depending on the web is a security liability. The web can never be truely secure due to the centralized nature of certificate authorities.

Internet communication in the modern age: The Web

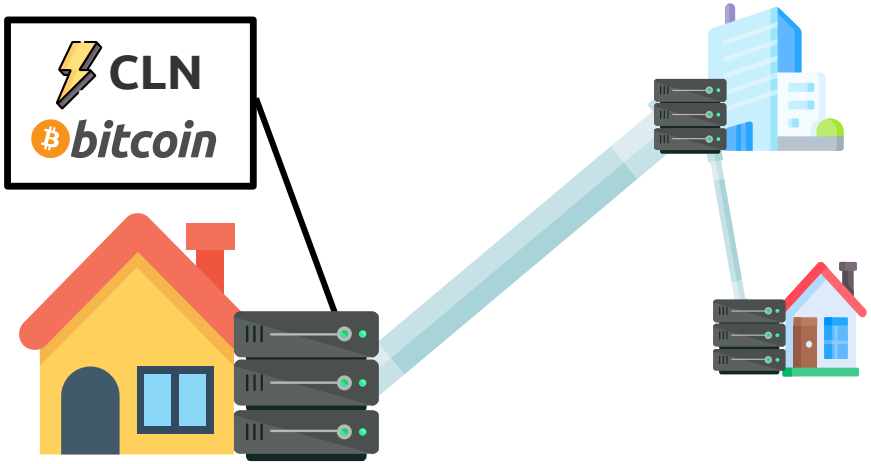
- Typically internet commerce is done through the world wide web, which is a suite of protocols such as HTTP, TLS, etc
- There's a problem: depending on the web is a security liability. The web can never be truely secure due to the centralized nature of certificate authorities.
- Intermediary certificate authorities can forge certificates, allowing interception of traffic. This isn't just theoretical, this has actually happened by state actors in Iran to snoop on gmail users.

Internet communication in the modern age: The Web

- Typically internet commerce is done through the world wide web, which is a suite of protocols such as HTTP, TLS, etc
- There's a problem: depending on the web is a security liability. The web can never be truely secure due to the centralized nature of certificate authorities.
- Intermediary certificate authorities can forge certificates, allowing interception of traffic. This isn't just theoretical, this has actually happened by state actors in Iran to snoop on gmail users.
- Due to the sensitive nature of internet money, it deserves a better protocol for secure communications that is simpler and can't be intercepted.

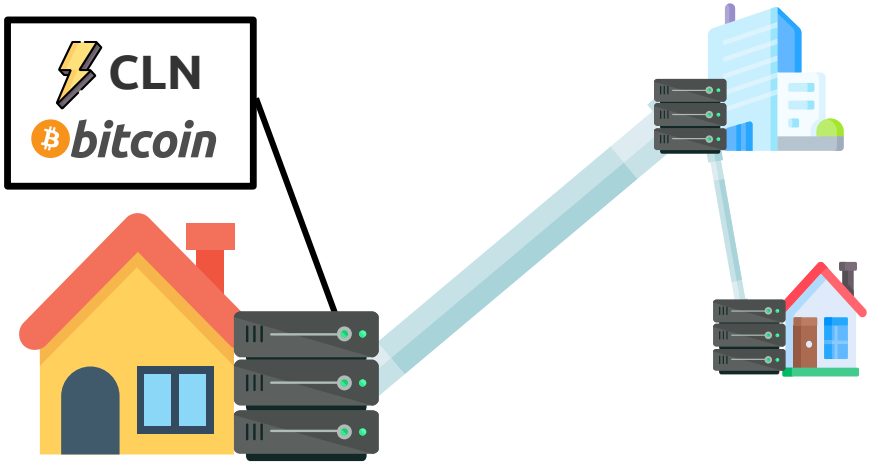
Enter the lightning network

- The Lightning Network doesn't build on the web, it is a distinct internet protocol with it's own security properties.



Enter the lightning network

- The Lightning Network doesn't build on the web, it is a distinct internet protocol with it's own security properties.
- You can connect to a lightning node in a similar way you would connect to a web server, so why not just communicate to a node directly to list products and pay for things?



Aside: Importance of running nodes

- Due to the nature of bitcoin, privacy, and the importance of holding your own keys, the ideal way to use bitcoin as a business or user is running your own nodes and having control of your keys on your own hardware.

Aside: Importance of running nodes

- Due to the nature of bitcoin, privacy, and the importance of holding your own keys, the ideal way to use bitcoin as a business or user is running your own nodes and having control of your keys on your own hardware.
- With bitcoin-core and a lightning node you can keep your transactions completely private and you don't have to outsource anything. You have a working bank + visa stack within your own organization or home.

Aside: Importance of running nodes

- Due to the nature of bitcoin, privacy, and the importance of holding your own keys, the ideal way to use bitcoin as a business or user is running your own nodes and having control of your keys on your own hardware.
- With bitcoin-core and a lightning node you can keep your transactions completely private and you don't have to outsource anything. You have a working bank + visa stack within your own organization or home.
- Now that we agree that running node is the best way to interact with the Bitcoin economy, what would be the best way to iteract with them?

Lightning as the communication layer for the Bitcoin Economy

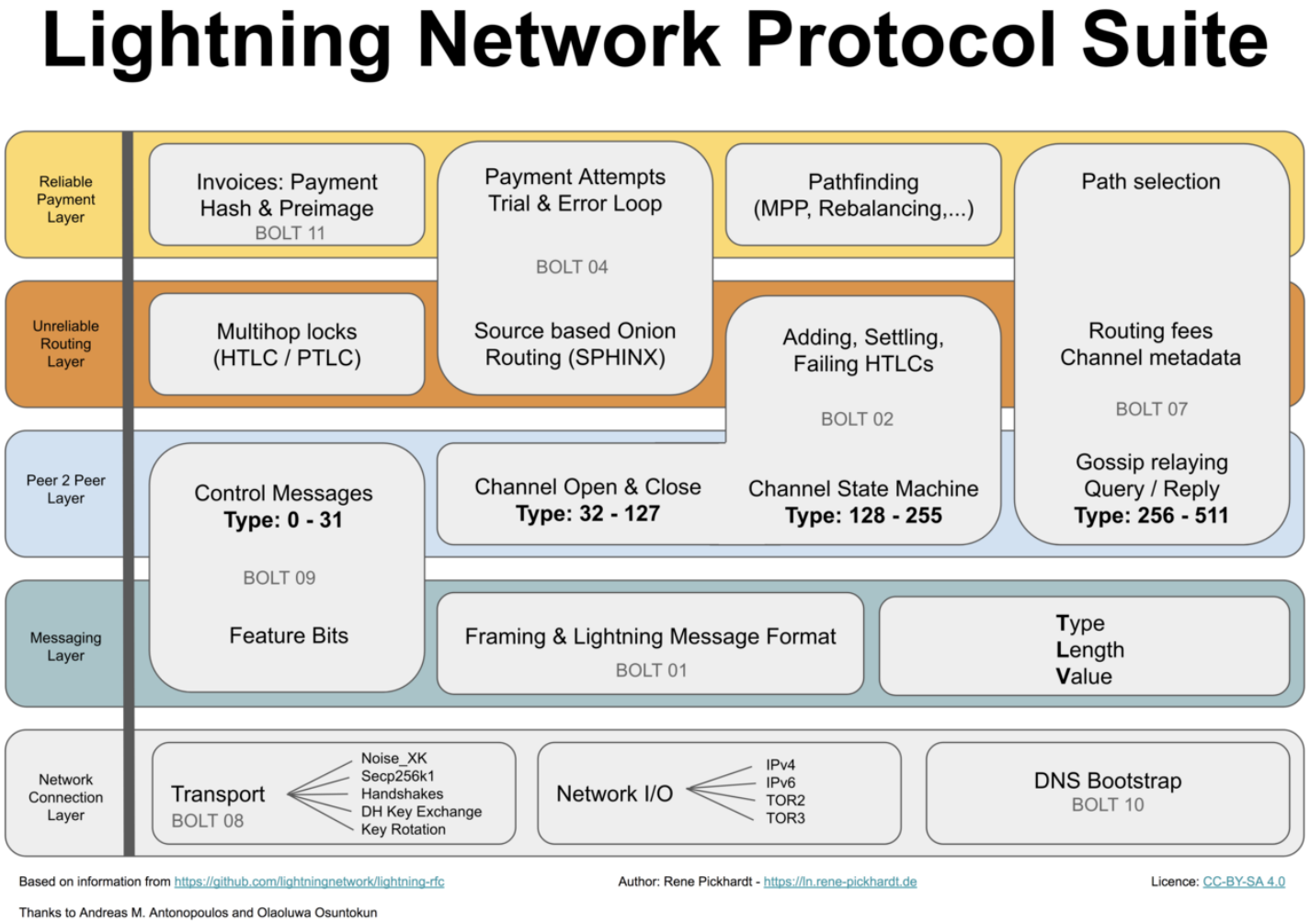
- We can utilize the lightning network itself as a means to interact with the bitcoin economy without relying on web servers, web services, and custodial web platforms.

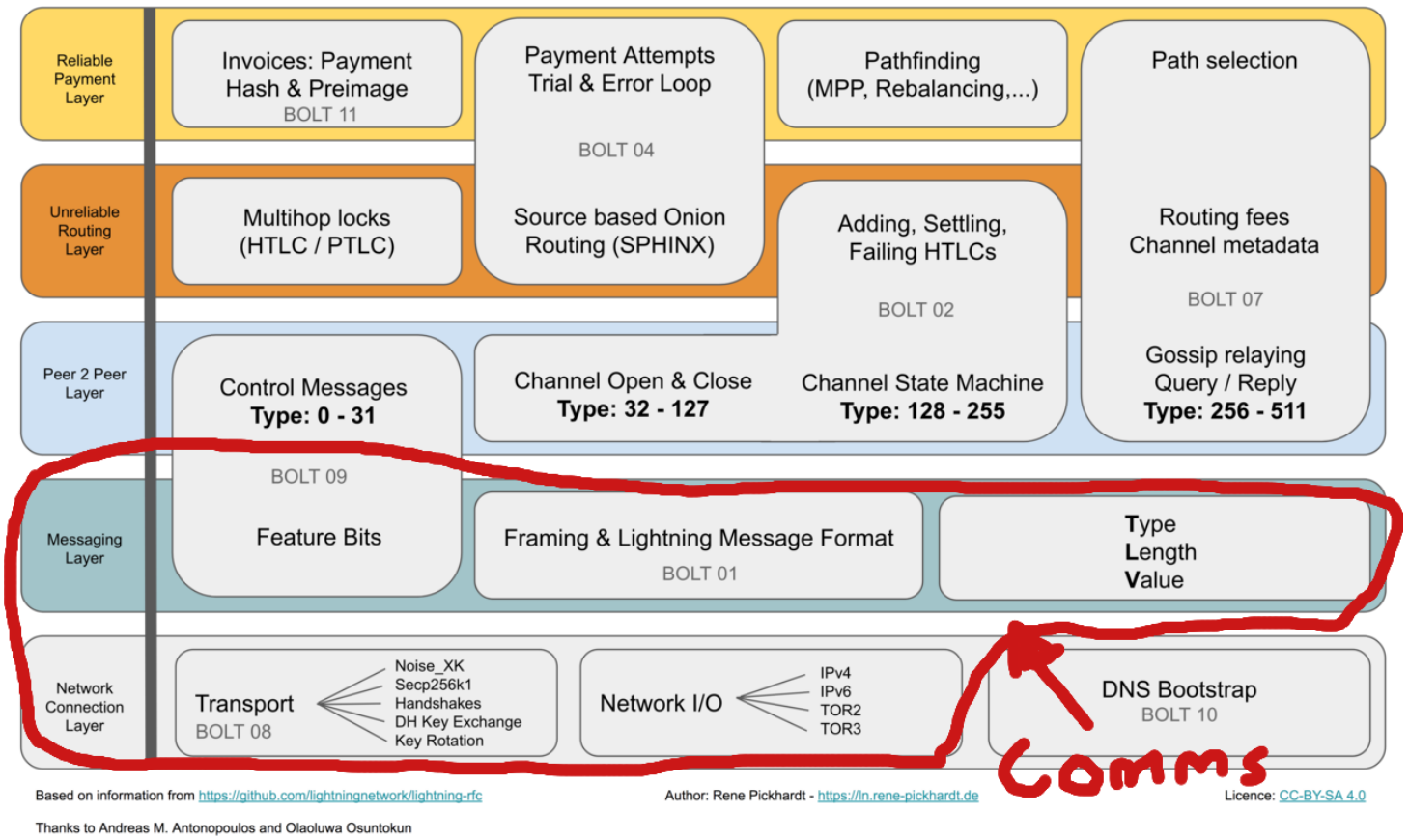
Lightning as the communication layer for the Bitcoin Economy

- We can utilize the lightning network itself as a means to interact with the bitcoin economy without relying on web servers, web services, and custodial web platforms.
- We can reduce the software burden for business and users when running nodes this way. All you need is to plug in a small bitcoin computer that provides standard APIs accessible over lightning, then you could have client software that talks to this API to provides dashboards, point of sales, wallets, anything you would need.

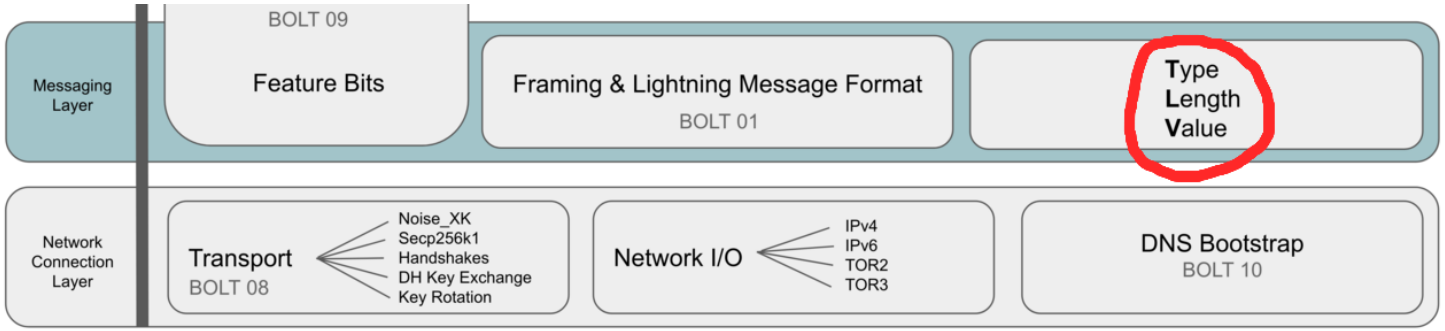
Lightning as the communication layer for the Bitcoin Economy

- We can utilize the lightning network itself as a means to interact with the bitcoin economy without relying on web servers, web services, and custodial web platforms.
- We can reduce the software burden for business and users when running nodes this way. All you need is to plug in a small bitcoin computer that provides standard APIs accessible over lightning, then you could have client software that talks to this API to provides dashboards, point of sales, wallets, anything you would need.
- This shifts the burden of development into client applications instead of pushing it onto the business owner who just wants to sell products, or a user who just wants to use their CLN node as a wallet.


Lightning Network Protocol Suite
Reliable Payment Layer
Invoices: Payment Hash & Preimage
BOLT 11
Payment Attempts Trial & Error Loop
BOLT 04
Pathfinding (MPP, Rebalancing,...)
Path selection
Unreliable Routing Layer
Multihop locks (HTLC / PTLC)
Source based Onion Routing (SPHINX)
Adding, Settling, Failing HTLCs
BOLT 02
Routing fees
Channel metadata
BOLT 07
Peer 2 Peer Layer
Control Messages
Type: 0 - 31
BOLT 09
Feature Bits
Channel Open & Close
Type: 32 - 127
Channel State Machine
Type: 128 - 255
Gossip relaying
Query / Reply
Type: 256 - 511
Messaging Layer
Framing & Lightning Message Format
BOLT 01
Type
Length
Value
Network Connection Layer
Transport
BOLT 08
Noise_XK
Secp256k1
Handshakes
DH Key Exchange
Key Rotation
Network I/O
IPv4
IPv6
TOR2
TOR3
DNS Bootstrap
BOLT 10
Based on information from https://github.com/lightningnetwork/lightning-rfc
Author: Rene Pickhardt - https://ln.rene-pickhardt.de
Licence: CC-BY-SA 4.0
Thanks to Andreas M. Antonopoulos and Olaoluwa Osuntokun

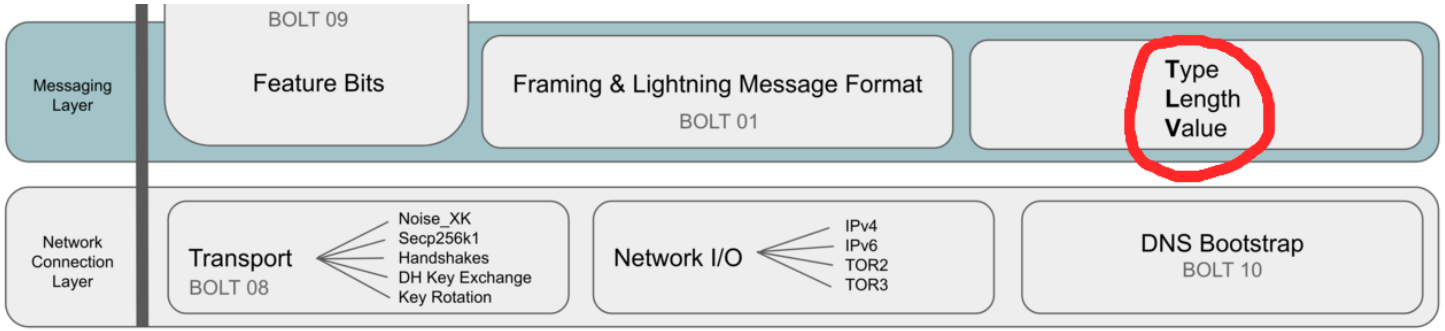

Lightning Network Protocol Suite
Reliable Payment Layer
Invoices: Payment Hash & Preimage
BOLT 11
Payment Attempts Trial & Error Loop
BOLT 04
Pathfinding (MPP, Rebalancing,...)
Path selection
Unreliable Routing Layer
Multihop locks (HTLC / PTLC)
Source based Onion Routing (SPHINX)
Adding, Settling, Failing HTLCs
BOLT 02
Routing fees
Channel metadata
BOLT 07
Peer 2 Peer Layer
Control Messages
Type: 0 - 31
Channel Open & Close
Type: 32 - 127
Channel State Machine
Type: 128 - 255
Gossip relaying
Query / Reply
Type: 256 - 511
Messaging Layer
BOLT 09
Feature Bits
Framing & Lightning Message Format
BOLT 01
Type
Length
Value
Network Connection Layer
Transport
BOLT 08
Noise_XK
Secp256k1
Handshakes
DH Key Exchange
Key Rotation
Network I/O
IPv4
IPv6
TOR2
TOR3
DNS Bootstrap
BOLT 10
Comms
Based on information from https://github.com/lightningnetwork/lightning-rfc
Author: Rene Pickhardt - https://ln.rene-pickhardt.de
Licence: CC-BY-SA 4.0
Thanks to Andreas M. Antonopoulos and Olaoluwa Osuntokun

TLVs



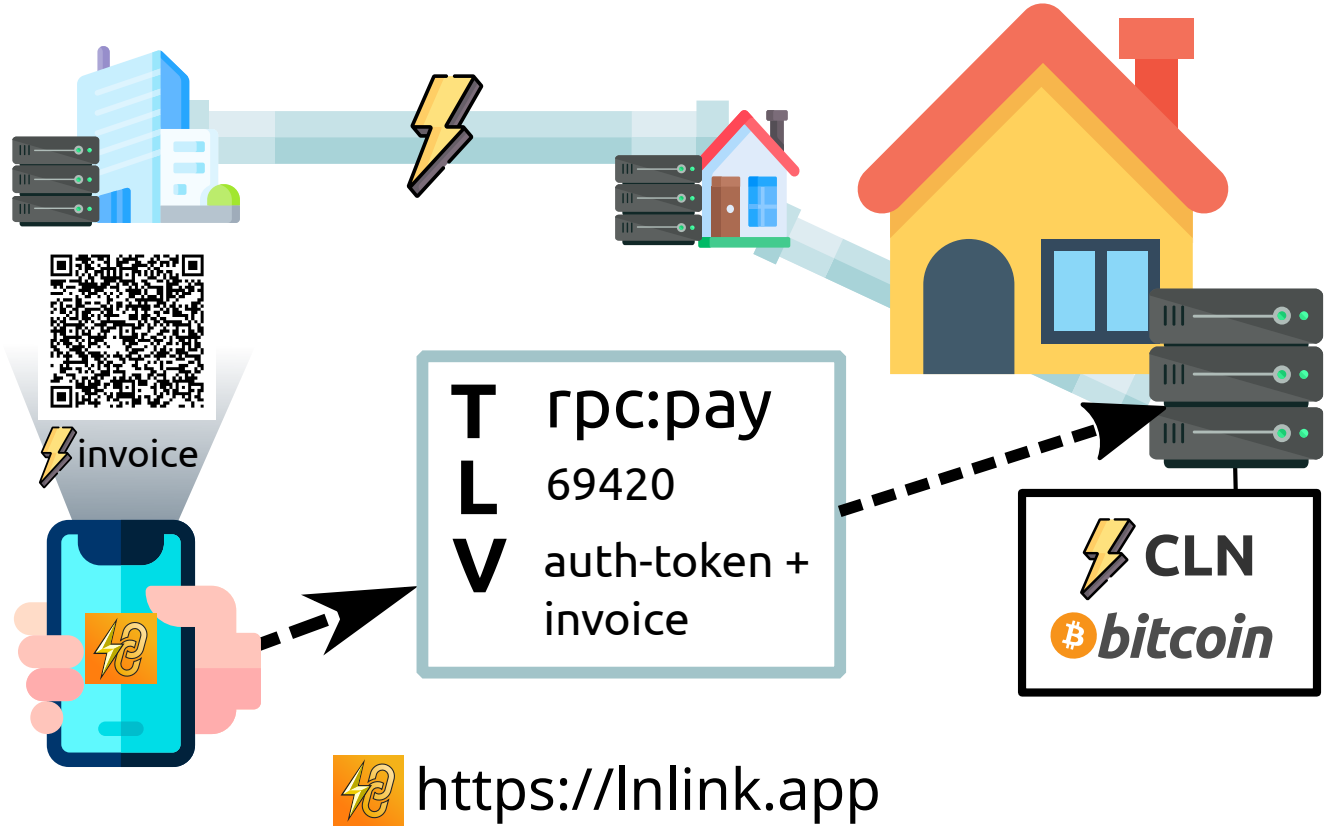
- Packets on the lightning network are composed of three data elements: type, length, and value

TLVs



- Packets on the lightning network are composed of three data elements: type, length, and value
- By including custom data in these packets, we can talk to a lightning node as if we were making a web request.

Example: Paying invoices with your node



Benefits of a payment setup like this

- No web dependency. This all works without interaction with the complex and not-as-secure web stack.

Benefits of a payment setup like this

- No web dependency. This all works without interaction with the complex and not-as-secure web stack.
- Low latency connection. Just a plain TCP port so unlikely to be blocked anywhere, which is important if you want reliable payments on the go.

Benefits of a payment setup like this

- No web dependency. This all works without interaction with the complex and not-as-secure web stack.
- Low latency connection. Just a plain TCP port so unlikely to be blocked anywhere, which is important if you want reliable payments on the go.
- Manage your lightning node channels from your phone, access your point-of-sale from your phone or anywhere.

Benefits of a payment setup like this

- No web dependency. This all works without interaction with the complex and not-as-secure web stack.
- Low latency connection. Just a plain TCP port so unlikely to be blocked anywhere, which is important if you want reliable payments on the go.
- Manage your lightning node channels from your phone, access your point-of-sale from your phone or anywhere.
- No Tor needed, no VPN needed, which are sometimes blocked on certain networks.

Example: Crowdfunding without a web server

Damus Android crowdfund

If ya'll help crowdfund me an android phone I can start working on an android version

Donations

This is a bolt12 offer, you can pay this with a CLN node. Otherwise press the button to get a bolt11 invoice.

REQUEST BOLT11 INVOICE

Total: **3,371,451.625 / 3,000,000 sats goal (1.1e+2%)**

Recent Donors

Note	Amount	
we got ya. @SN	16,537 sats	28 days ago

Example: Crowdfund code

```
async function make_request(method, rune, params) {
    const LNSocket = await lnsocket_init()
    const ln = LNSocket()

    ln.genkey()
    await ln.connect_and_init("03f3c108ccd536b8526841f0a5c58212bb9e6584a1eb493080e7c1cc34f82dad71",
"wss://cln.jb55.com:443")

    const {result} = await ln.rpc({ rune, method, params })

    ln.disconnect()
    return result
}

function fetch_tipjar_summary() {
    const rune = "b3Xsg2AS2cknHYa6H94so7FAVQVdnRSP6Pv-1WOQEBc9NCZtZXRob2Q9b2ZmZXItc3VtbWFyeQ=="
    return make_request("offer-summary", rune, {
        offerid: "2043536dfec68d559102f73510927622812a230cfdda079e96fccbfe35a96d11",
        description: "@damus-android",
        limit: 5
    })
}

function make_invoice(description) {
    const rune = "LZwGZJO7wZgmoScFQb5reZ0Ii8qPKCeUfTb-UcbDxWw9MTImbWV0aG9kPWludm9pY2U="
    description = (description && `${description} @damus-android`) || "@damus-android donation"
    return make_request("invoice", rune, {
        msatoshi: "any",
        label: `damus-android-${new Date().getTime()}`,
        description: description
    })
}
```

Example: Manage your bitcoin-core node

1 2 3: lnlink-core

Bitcoin Wallet

Transactions | Descriptors | ☑ Masked

Transactions

Date	Amount	Label	Type
3 days ago	🙈	shakepay-2022-07-13	receive
11 days ago	🙈	shakepay-2022-06-05	receive
21 days ago	🙈	3NL...	send
29 days ago	🙈	375...	send
1 months ago	🙈	shakepay-2022-05-08	receive
2 months ago	🙈	shakepay-2022-04-11	receive
2 months ago	🙈	35G...	send
2 months ago	🙈	3Me...	send
3 months ago	🙈	shakepay-2022-03-08	receive

Example: Manage your bitcoin-core node

- bitcoin-core is not designed to be accessed from an external network, typically due to the web concerns as described in previous slides.

Example: Manage your bitcoin-core node

- bitcoin-core is not designed to be accessed from an external network, typically due to the web concerns as described in previous slides.
- Lightning provides a secure way to access your node remotely. From a mobile app or http webpage, you can generate addresses, get notifications when you receive funds, manage your bitcoin wallet from anywhere, create onchain transactions, etc.

Insocket

- All of these examples use lnsocket: a C, Go and javascript library for sending and receiving lightning TLVs. On mobile the browser, or server-to-server.

Insocket

- All of these examples use lnsocket: a C, Go and javascript library for sending and receiving lightning TLVs. On mobile the browser, or server-to-server.
- server-to-server use cases of lnsocket are also interesting. For example, you can use this to create an lnurl proxy server enables lnaddresses for your CLN node. This server can be hosted anywhere and doesn't require you to run a web server at home.

Insocket

- All of these examples use lnsocket: a C, Go and javascript library for sending and receiving lightning TLVs. On mobile the browser, or server-to-server.
- server-to-server use cases of lnsocket are also interesting. For example, you can use this to create an lnurl proxy server enables lnaddresses for your CLN node. This server can be hosted anywhere and doesn't require you to run a web server at home.
- In the browser, a connection is established via websockets. It can connect directly to your CLN node (which has built-in websocket support). You can have a completely secure end-to-end encrypted session with just http, no TLS/green lock needed!

Insocket

- All of these examples use lnsocket: a C, Go and javascript library for sending and receiving lightning TLVs. On mobile the browser, or server-to-server.
- server-to-server use cases of lnsocket are also interesting. For example, you can use this to create an lnurl proxy server enables lnaddresses for your CLN node. This server can be hosted anywhere and doesn't require you to run a web server at home.
- In the browser, a connection is established via websockets. It can connect directly to your CLN node (which has built-in websocket support). You can have a completely secure end-to-end encrypted session with just http, no TLS/green lock needed!
- https://github.com/jb55/lnsocket

CLN plugins

- Many of these examples require custom CLN plugins, ideally in the future we could rely less on these and just have CLN come pre-installed with a bunch of standard RPCs (remote procedure calls) that you can call.

CLN plugins

- Many of these examples require custom CLN plugins, ideally in the future we could rely less on these and just have CLN come pre-installed with a bunch of standard RPCs (remote procedure calls) that you can call.
- For example all examples require a plugin called commando which allows you to call CLN's jsonrpc over lightning itself. Perhaps one day CLN could have commando available via a config option, or provide a cross-node standard for lightning RPC.

CLN plugins

- Many of these examples require custom CLN plugins, ideally in the future we could rely less on these and just have CLN come pre-installed with a bunch of standard RPCs (remote procedure calls) that you can call.
- For example all examples require a plugin called commando which allows you to call CLN's jsonrpc over lightning itself. Perhaps one day CLN could have commando available via a config option, or provide a cross-node standard for lightning RPC.
- The bitcoin-core example required me to write a plugin that exposes bitcoin-core's jsonrpc as CLN rpc. This allowed me to call bitcoin's jsonrpc via commando over lightning.

Conclusion

- Lightning enables way to securely access bitcoin and lightning nodes over the internet with low latency and with high security, while at the same time making it simpler for people to interact with their nodes without being a hardcore computer nerd.

Conclusion

- Lightning enables way to securely access bitcoin and lightning nodes over the internet with low latency and with high security, while at the same time making it simpler for people to interact with their nodes without being a hardcore computer nerd.
- It enables developers to create apps that interact with lightning and bitcoin nodes without having to force people running nodes to install any custom software which increases complexity and maintenance burden.

Conclusion

- Lightning enables way to securely access bitcoin and lightning nodes over the internet with low latency and with high security, while at the same time making it simpler for people to interact with their nodes without being a hardcore computer nerd.
- It enables developers to create apps that interact with lightning and bitcoin nodes without having to force people running nodes to install any custom software which increases complexity and maintenance burden.
- It looks like Lightning could become the *communications layer* for the Bitcoin economy.

Thanks

The End