



Blockstream

# Developing on c-lightning

# Goals

- Fast & Lightweight
- Flexible & Customizable
- Optimize for Powerusers

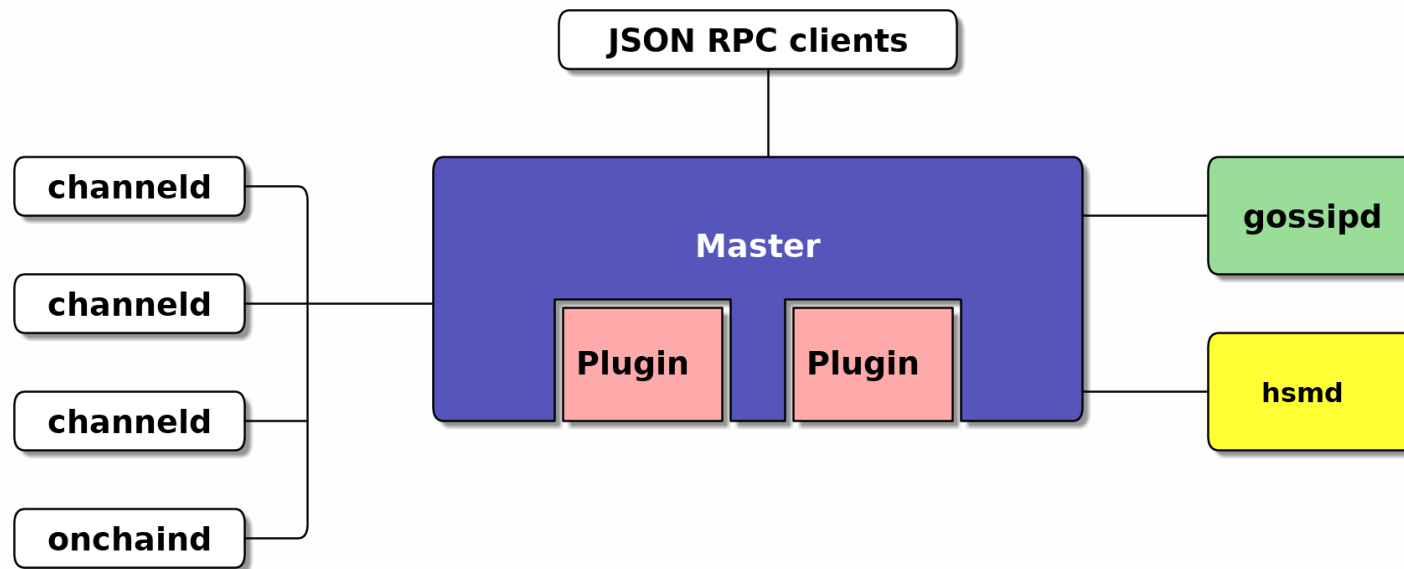
# TL;DR, I want my node now!

```
$ sudo add-apt-repository ppa:lightningnetwork/ppa  
$ sudo apt-get update  
$ sudo apt-get install lightningd
```

```
$ lightningd --network=bitcoin --log-level=debug
```

```
$ lightning-cli getinfo
```

# Architecture



# JSON-RPC Interface

UNIX Socket exposed at `$HOME/.lightning/lightning-rpc`

```
$ lightning-cli help  
$ lightning-cli getinfo
```

- Python: <https://pypi.org/project/pylightning/>
- Javascript: <https://github.com/ElementsProject/lightning-charge>
- Go: <https://github.com/cdecker/kugelblitz>
- Bash: socat

# Invoicing/Receiving Payments

```
$ lightning-cli invoice [msatoshi] [label] [description]
```

```
$ lightning-cli listinvoices  
$ lightning-cli waitinvoice [label]  
$ lightning-cli waitanyinvoice [last-pay-index]
```

```
$ lightning-cli deexpiredinvoice [maxexpiry]
```

# Sending Payments

```
$ lightning-cli pay [bolt11]
```

```
$ lightning-cli getroute [destination] [msatoshi] [riskfactor]  
$ lightning-cli sendpay [route] [payment_hash]  
$ lightning-cli waitsendpay [payment_hash]
```

```
$ lightning-cli listpayments
```

# Subdaemons

Subdaemons are tightly integrated with the master daemon, flexible but hard to code.

- Pluggable Bitcoin backend (bitcoind, spruned, neutrino)
- Routing service and gossip store
- Replace the HSMd with:
  - Hardware Wallet
  - Mobile App
  - ...



# Plugins (tbd...)

Customization made simple, you just read/write JSON.

- Autopilot
- Payment logic (hold HTLC until delivered, notify service, ...)
- Short-circuit payments
- External continuations
- ...



Blockstream

Thanks!  
Questions?

