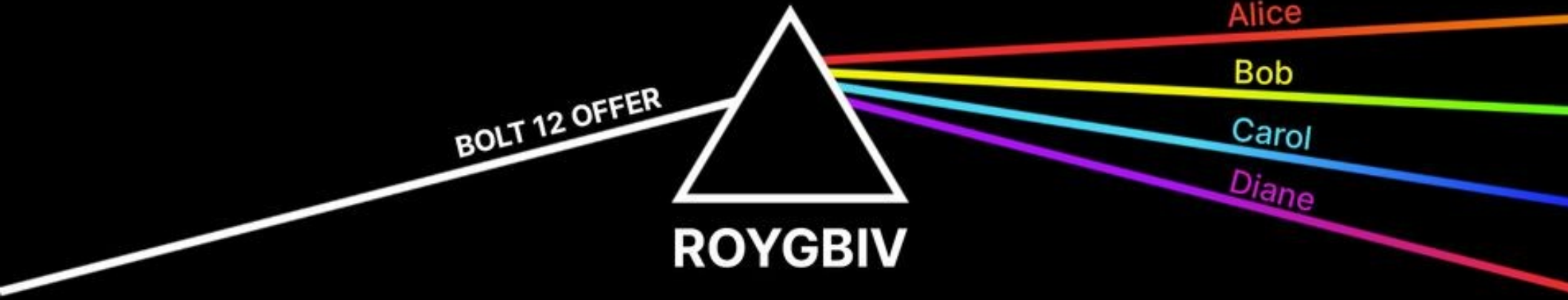
BOLT12

- Reusable Payment Codes – necessary for prism accounting!
- Receiver Privacy – Route Blinding (hide receiver node pubkey)
- Issue unique BOLT12 offers for each “line-of-accounting.”
- No web server required. Eliminates unnecessary dependencies
- Recurrence and payout triggers (e.g., amount)
- Withdraw Offers (recipient of sats can initiate withdrawl)



lnplay.guide/showcase/tabconf2024

tip.

farscapian.com/

/about

/qualifications

/portfolio

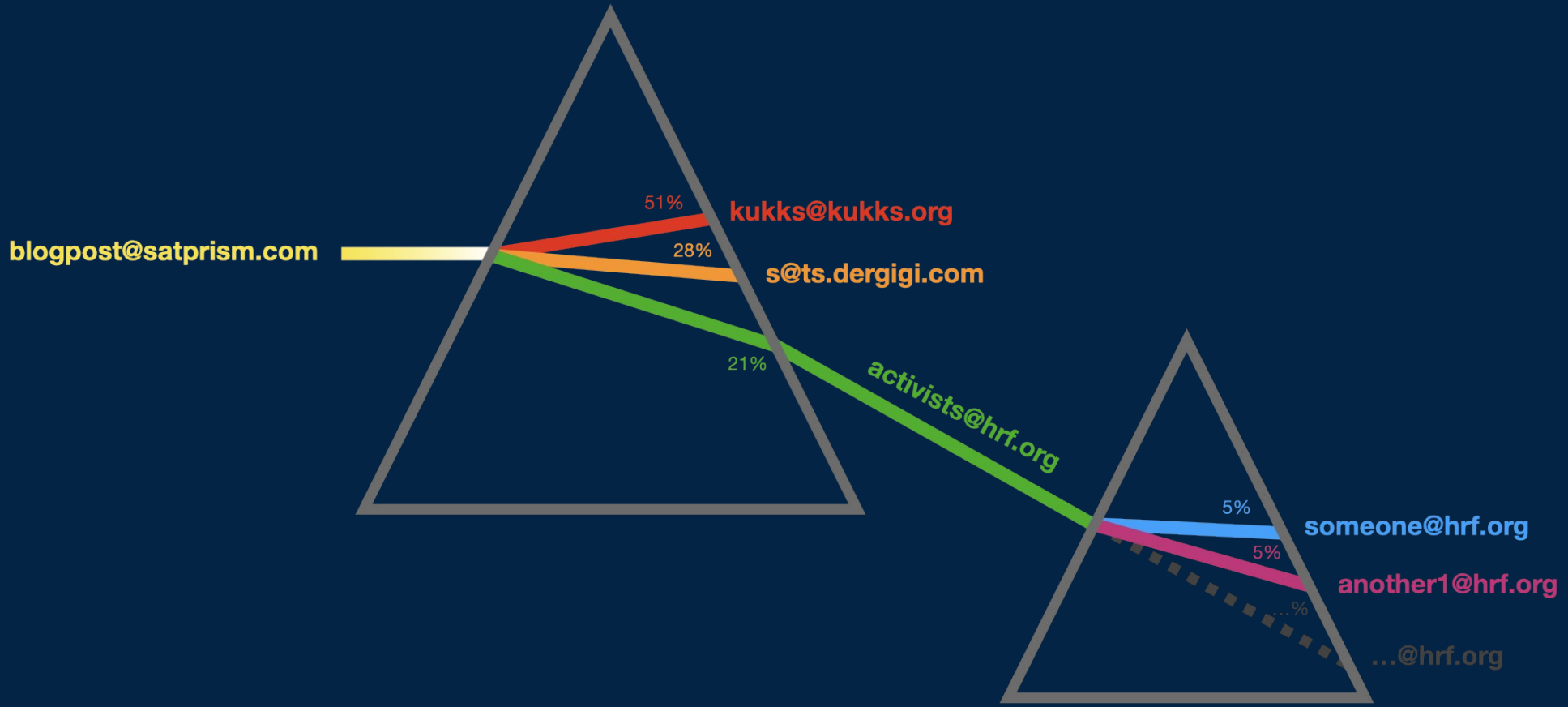
Agenda

- Prisms: What is a Prism? Why BOLT12?
- roygbiv.guide: a BOLT12-Prism-native Website
- LNPlay Prism Simulation featuring Clams Remote

What is a Lightning Prism?

What is a Lightning Prism?

- A [LN] Prism is a construct that allows for “lightning address value split workflows”
  - Prism identified by address (LN Address, BOLT12 offer, pubkey for keysend, etc)
  - Has **one or more** members (acts as a proxy)
  - Another prism can be one of the recipients (composable)
  - Splits are defined programmatically


blogpost@satprism.com
51%
kukks@kukks.org
28%
s@ts.dergigi.com
21%
activists@hrf.org
5%
someone@hrf.org
5%
another1@hrf.org
...%
...@hrf.org

Why are Prisms interesting?

- Prisms are composable
  - Have the ability to interconnect and integrate different parts together to build a whole (like legos).
- Prisms can help automate your sats flows.
  - Set payout criteria
- Prisms can buffer your sats flows.
  - Useful for offline nodes or liquidity issues
  - Make payouts more consistent by adding payout_threshold

Prism Composability

- **Composability** - idea that a component (prism) can be combined and recombined in various ways to create new functionality.

Prism Use Cases

- **Value4Value Websites*** - that split tips among the posts various authors, editors, reviewers, etc.
- **Shopping Carts** – construct an offer backed by a Prism that splits payments to item offers, taxes, fees, profits, etc.
- **Stock Ownership** – split is like shares which entitle you to a percentage of overall profits of sats flows.
- **Employer Matching** – employee contributions to a prism (which can be line-of-accounting) can be matched: e.g., total outlays are 110% of income amount.
- **Communal Tip Jars** – create a Prism Offer that splits tips among serving staff, bar staff, kitchen, etc.
- **Band Prisms*** – Create a prism that splits tips among band members.


PLINKO
1
2
3
4
5
6
7
8
9
10

Definitions

- **Prism** – a construct on Lightning Network for splitting incoming payments.
- **Prism Policy** – a JSON document that represents prism policy.
- **Member** – each prism has ONE OR MORE members
- **Split** – a float representing member's relative percentage of the total outlays.
- **Binding** – bindings are created when you apply a prism to a BOLT12 offer. At this point, we track outlays (state) and treat sats on the offer as a flow.
- **Payout** – when a prism is are sent sats
- **outlay_msat** – an amount to be spent on the destination of a member.
- **payout_threshold** – payouts are only triggered when outlays exceed this value .
- **destination** – a BOLT12 offer (e.g., 'lno…') representing the member payee (remote/local).

bolt12-prism API

./lightning-cli.sh help | grep prism

prism-addbinding prism_id [offer_id]
prism-create members [description] [outlay_factor]
prism-delete prism_id
prism-deletebinding [offer_id]
prism-list [prism_id]
prism-listbindings [offer_id]
prism-pay prism_id [amount_msat] [label]
prism-setoutlay [offer_id] [member_id] [new_outlay_msat]
prism-update prism_id members

lightning-cli prism-list


```
{
  "prisms": [
    {
      "prism_id": "8594cf28d393bff1d1dd7577731cc6300119fdeab03bd20a86290b34730c9a21",
      "description": "Band Prism",
      "timestamp": 1729605449,
      "outlay_factor": 0.6667,
      "prism_members": [
        {
          "member_id": "328e37134bdd92870c19b65907d0f4be265110f2eb27f28d89270897a2daf5b6",
          "description": "Drummer",
          "destination": "lno1qgsqvgnwgcg35z6ee2h3yczraddm72xrfua9uve2rlrm9deudeun05rxuvfc",
          "split": 1.0,
          "fees_incurred_by": "local",
          "payout_threshold_msat": 5000000
        },
        {
          "member_id": "1bd50184ed88136a9630b7a5711071b5b807c44e089d5fb6a122da272338d9a9",
          "description": "Guitarist",
          "destination": "lno1qgsqvgnwgcg35z6ee2h3yczraddm72xrf6cup0829ywruqg77h4zmpzs8xthx",
          "split": 1.0,
          "fees_incurred_by": "remote",
          "payout_threshold_msat": 5000000
        }
      ]
    }
  ]
}
```

lightning-cli prism-listbindings


```
{
  "bolt12_prism_bindings": [
    {
      "offer_id": "b47a195e3b768933308b93ce16f3fa2036e1b6ed80c202f4e7ff0354aff3ab24",
      "prism_id": "0e38b824f6bc6e0c8946fa2ca5c68fd28757dfaebc27e8b632250b81e09aea82",
      "timestamp": 1729910682,
      "member_outlays": [
        {
          "member_id": "813fe892cc0f4bf41fd6301c606dcebd47247735b92152129b6f42f3c4d7a",
          "outlay_msat": 0
        },
        {
          "member_id": "16033ce59d0750f638ec30d182680d7a8e3a45db7c2ad18311d126f7be19",
          "outlay_msat": 0
        },
        {
          "member_id": "a48a8b351386ee8fb9a4336daf46402c4e5b5fc3f9d5d21e495f434ef0ed8",
          "outlay_msat": 0
        }
      ]
    }
  ]
}
```

Plugin workflow 1

- CLN Plugin listens for “invoice_payment” event
- Label: “offer_id-invoice_id”

JSON

```
{
  "invoice_payment": {
    "label": "unique-label-for-invoice",
    "preimage": "0000000000000000000000000000000000000000000000000000000000000000",
    "msat": 10000,
  }
}
```

Plugin Workflow 2

- Plugin checks if there is a binding
- If so, calculate and increase member outlays according to split
- Outlays remain UNTIL a payout succeeds.
- Outlay decreases only when payout succeeds!
  - Total amount depends on who is paying for fees!

Public/Private Prisms

- Gigi: “Users should have a way to see how payments are split that is both easy to understand and verify”.
  - e.g., NIP-33 – replaceable events.
- BOLT12-Prisms are private by default.
- You can make them public via Lightning Network (e.g., commando+rune) or by exposing outlay/prism info via HTMX or JSON, etc.

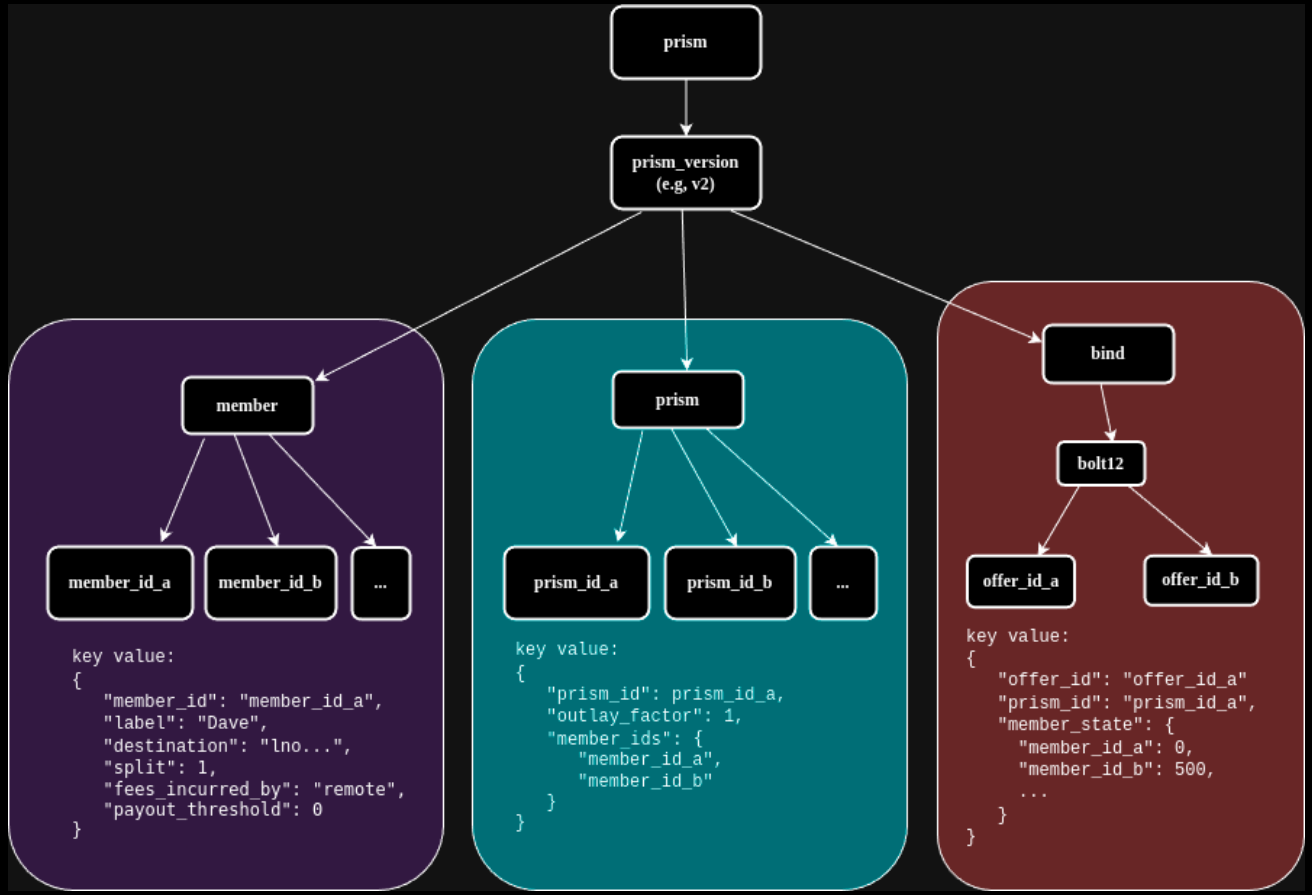
Plugin Design Goals

- ATOMICITY in Prism payouts are NOT DESIRABLE nor doable (without protocol support).
  - Payout fees (are known only AFTER the payment succeeds)
  - Hard to split 100 sats when fees aren’t known ahead of time
- Outlay tracking helps deal with offline nodes or unpayable invoices (e.g., due to liquidity).

Prism DB Structure

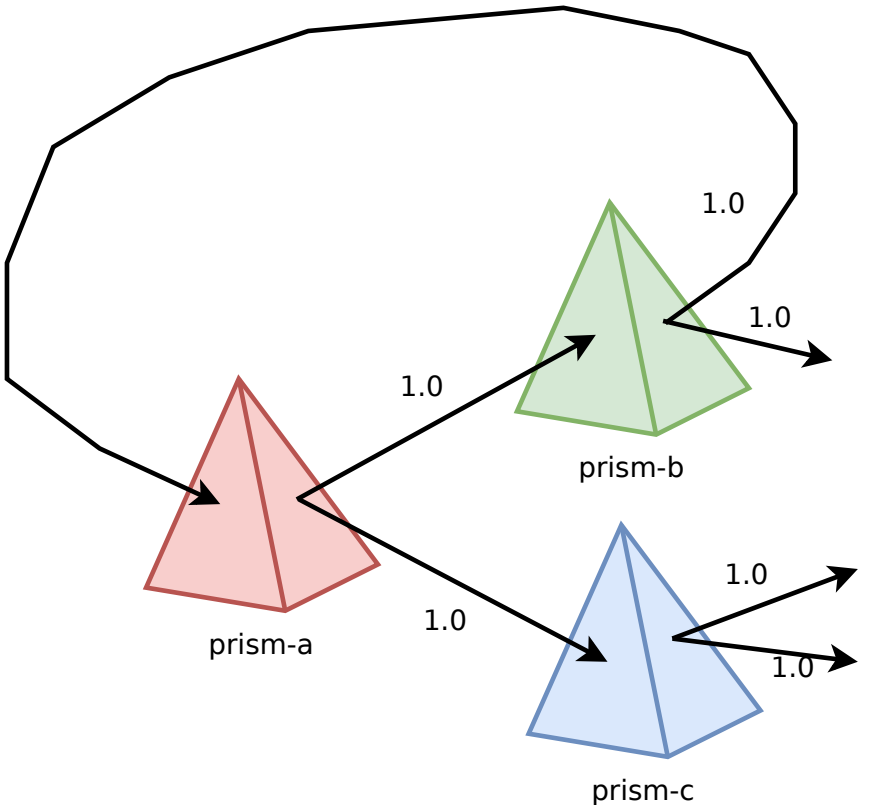
"prism_binding_key": [
"prism",
"v2.1",
"bind",
"bolt12",

"f1e8ea8617a24b321d8de0
a41c1177b106beab578fff29
a844d0a39727660cd8"
],



Loops

- If we view sats as flows like packets on the Internet, then
- Very important to avoid loops!
  - Fees dis-incentivize Loops.
  - Locally we must ensure the all Prisms form a DAG.
  - Other mitigation includes “TTL” on sats?
  - Need: Prism Explorer

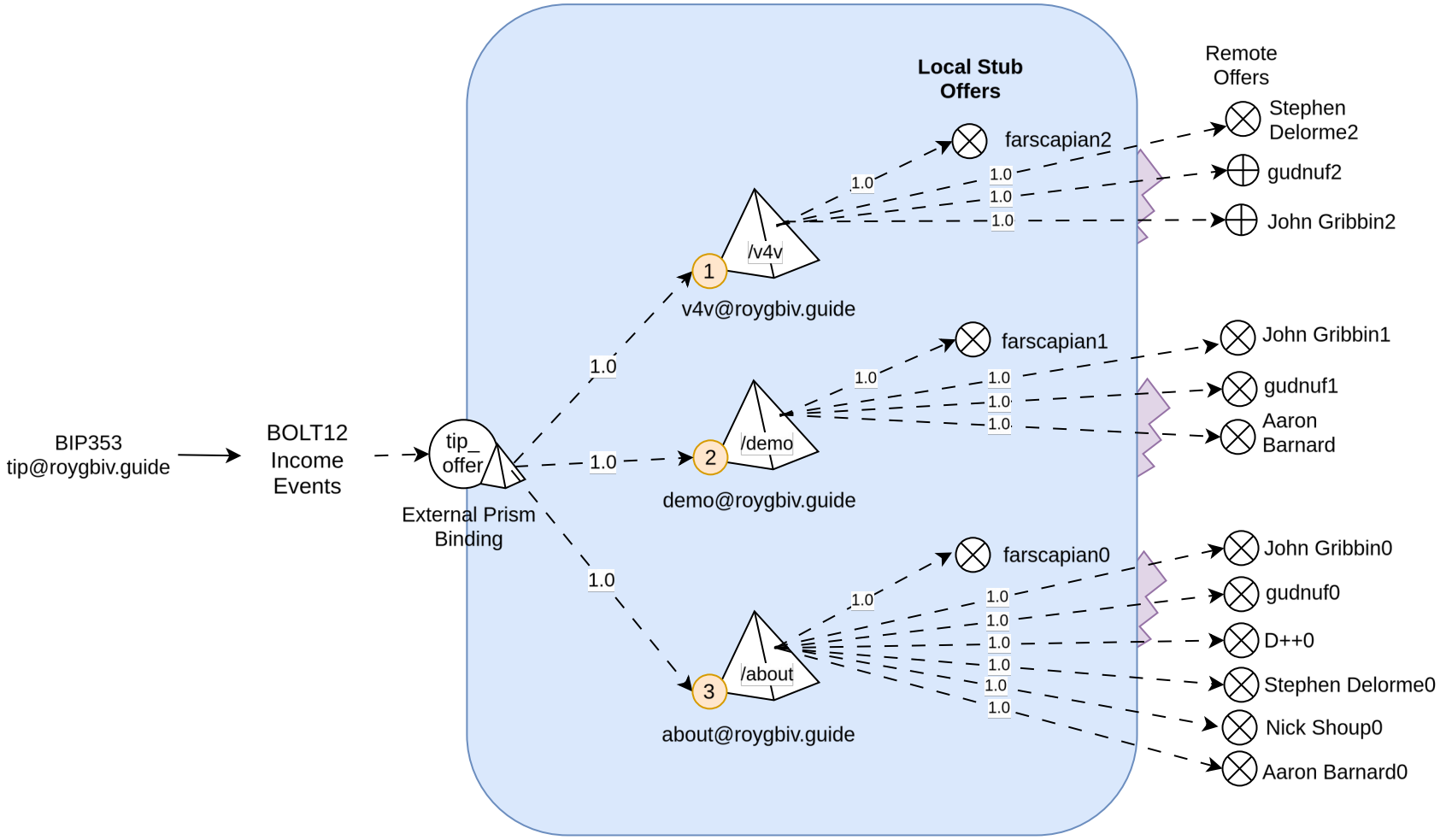


Ideas to implement

- Support **BIP353** & **npub** destinations
- Scheduled Outlay Clearing
  - Right now, outlay clearing occurs only when there’s an income event and when destination node is online and payable
- Issue BOLT12 Withdraw offers to clear outlays
  - Notify destination over nostr?
- Explicit Pay-to-self
  - Preventing local loops

The problem of self-pay

- Assume there’s a prism.
- One of the members of the prism hosts the Prism (i.e., runs the Lightning Node).
  - In other words, one of the “payouts” should go to the prism host.
- Solution: **pay-to-self**
  - Implicit: we retain sats by reducing total outlays
  - Explicit: we create a member with a destination (offer) that we issued*



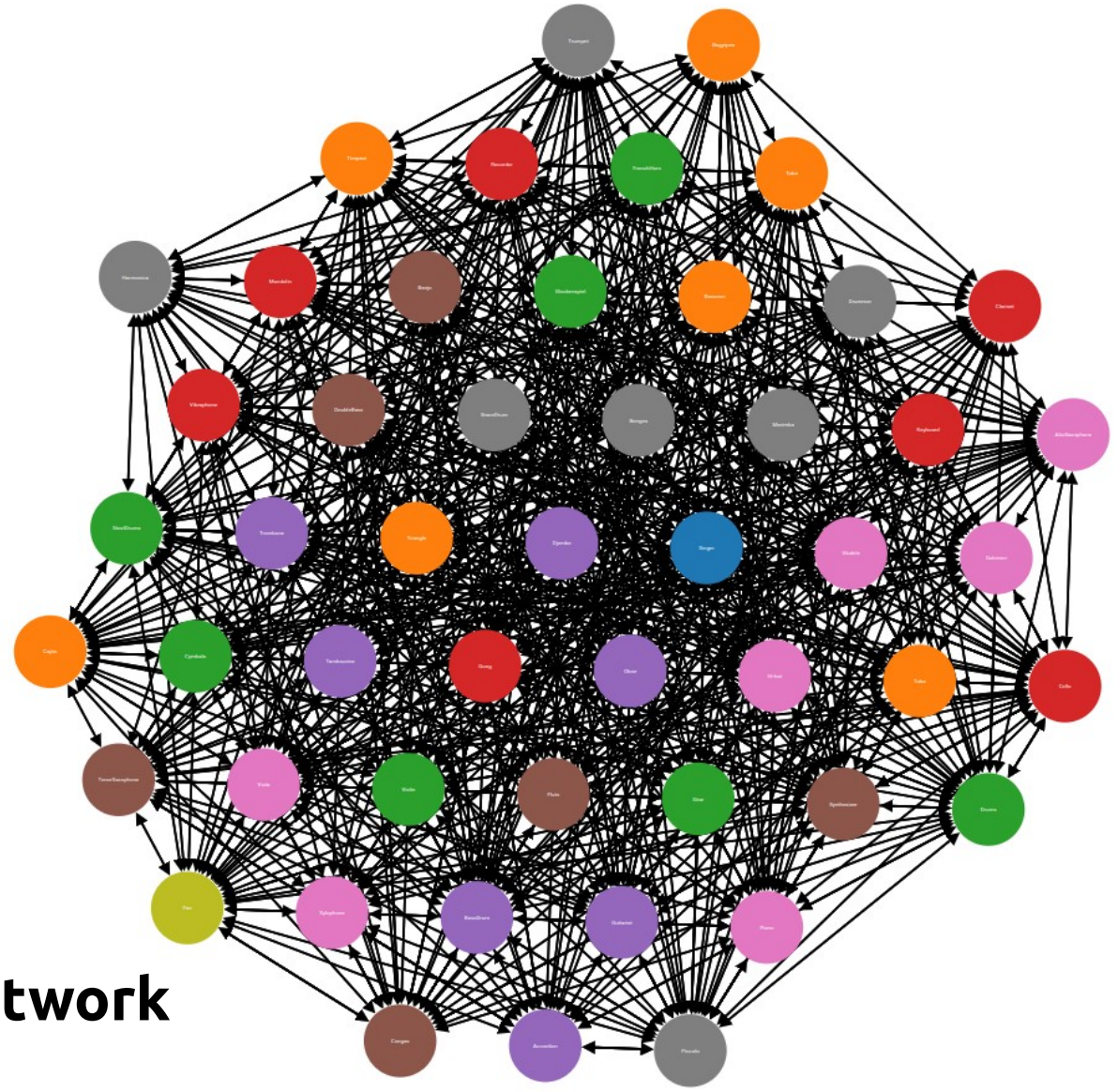
Local Core Lightning Node
running the prism plugin

LNPlay Prism Simulation

- LNPlay is a Lightning Network Simulator (FOSS)
- Clams Remote just added support for BOLT12-Prisms :)
- Leave feedback send email to feedback@roygbiv.guide

lnplay.guide/prism

Channel Graph



Lnplay.network

Prism Ambassadors

- **Randy Naar** – Works on Blockstream Greenlight
- **Michael Evans** – created lnplay.network and created PoC for Clams Remote Prism feature
- **Vake** – Top V4V Booster
- **Stephen DeLorme** – ATL Bitlab / Bitcoin Design Community

For each group:

- Each group is composed of a Prism Ambassador, a Bob, and the rest are prism members.
- Bob creates and hosts the Band Prism, but first he needs to **create a new offer** (call it the Band Prism Offer). This will be the offer that when paid to by the Prism Ambassador, triggers the band prism.
- Every member in each group **creates an Any Offer**. Show the QR code so it can be scanned by Bob as he constructs the prism.
- On Bob, **create the Band Prism policy** by scanning the any offers on the members. Bob should set the outlay_factor to 0.8% as he's taking a 20% cut of the proceeds (implicit pay-to-self of 20%).
- Bob **binds the Prism to the Band Prism Offer** he created earlier.
- The Prism Ambassador **pays to the Band Prism Offer** hosted on Bob.

feedback@roygbiv.guide

Who to thank

- Core Lightning (for implementing pay-to-self, and generally for the excellent plugin architecture), specifically Vincenzo Palazzo and Rusty Russell.
- Michael Evans and John Gribbin for adding support for the BOLT12-Prism plugin in Clams Remote.
- And the Clams Remote project for making this simulation possible
- Farscapian & gudnuf & daywalker (tests) for the prism plugin repo.
- Stephen Delorme – created/redesigned bolt12.org
- CLBOSS Project for automated channel management