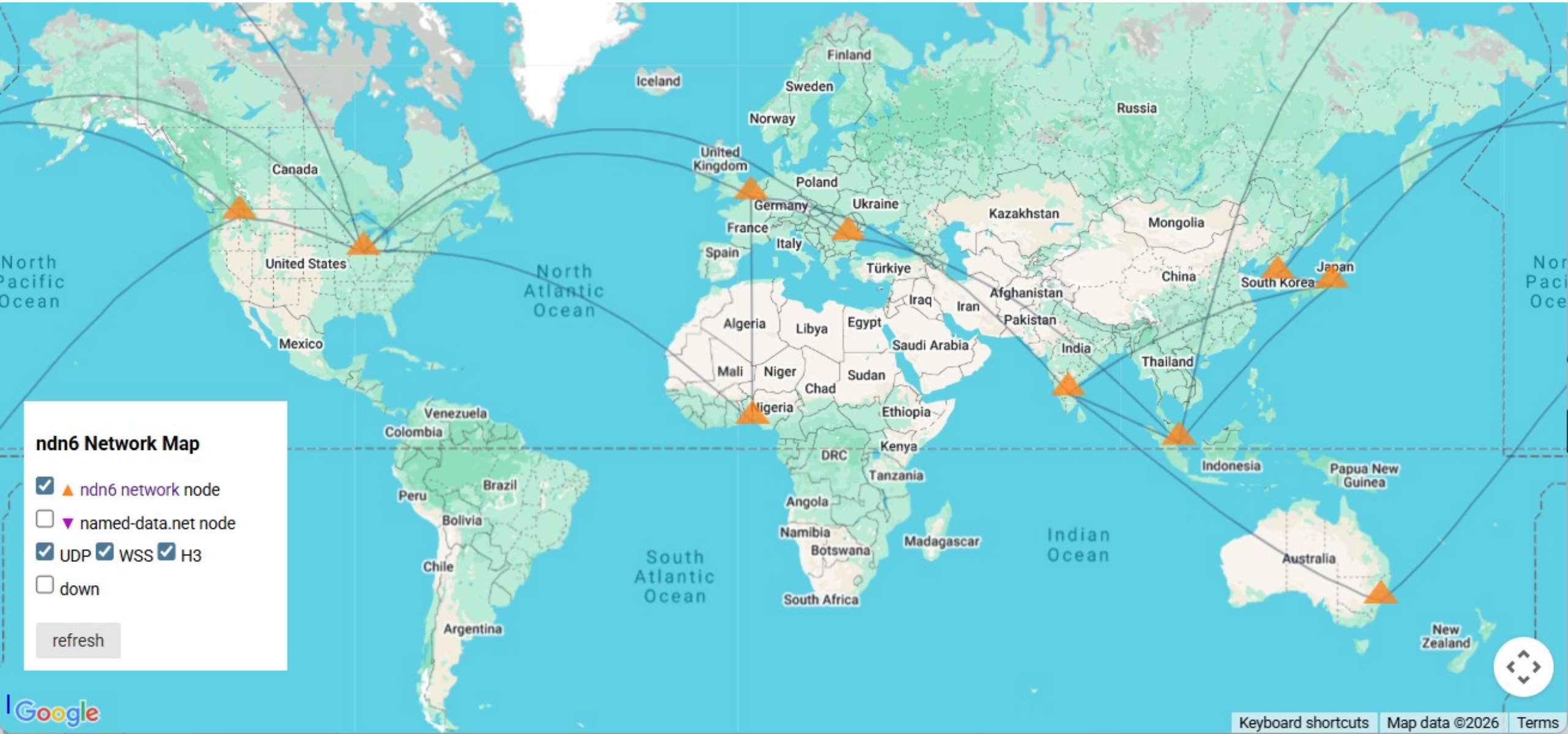
Why Ownly does not Work on the ndn6 Network? A Decade of Operational Gaps in Trust and Routing

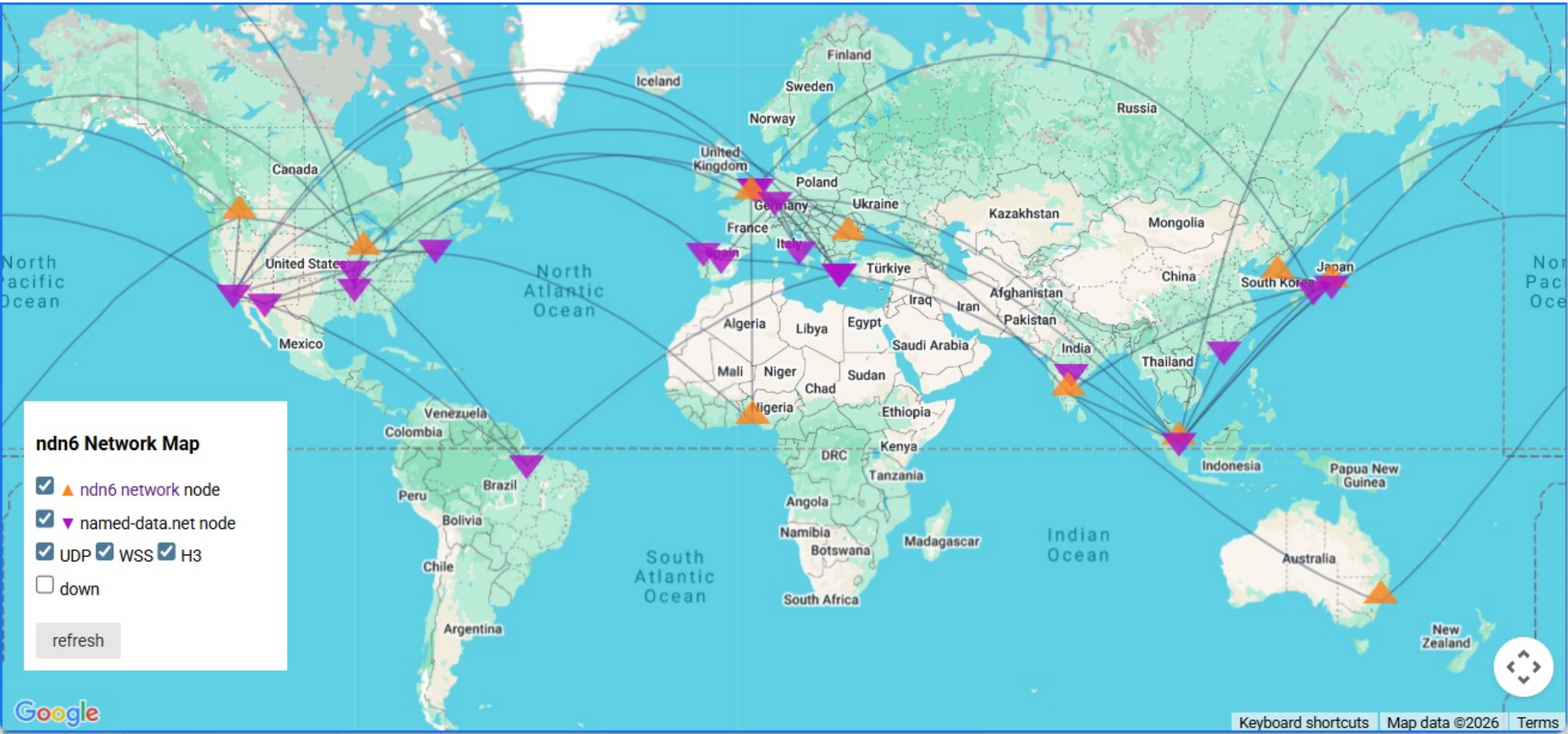
Junxiao Shi – @yoursunny

NDN Community Meeting 2026-05-19

ndn6: Independently Operated NDN Network



ndn6: Interconnected with Global NDN Testbed



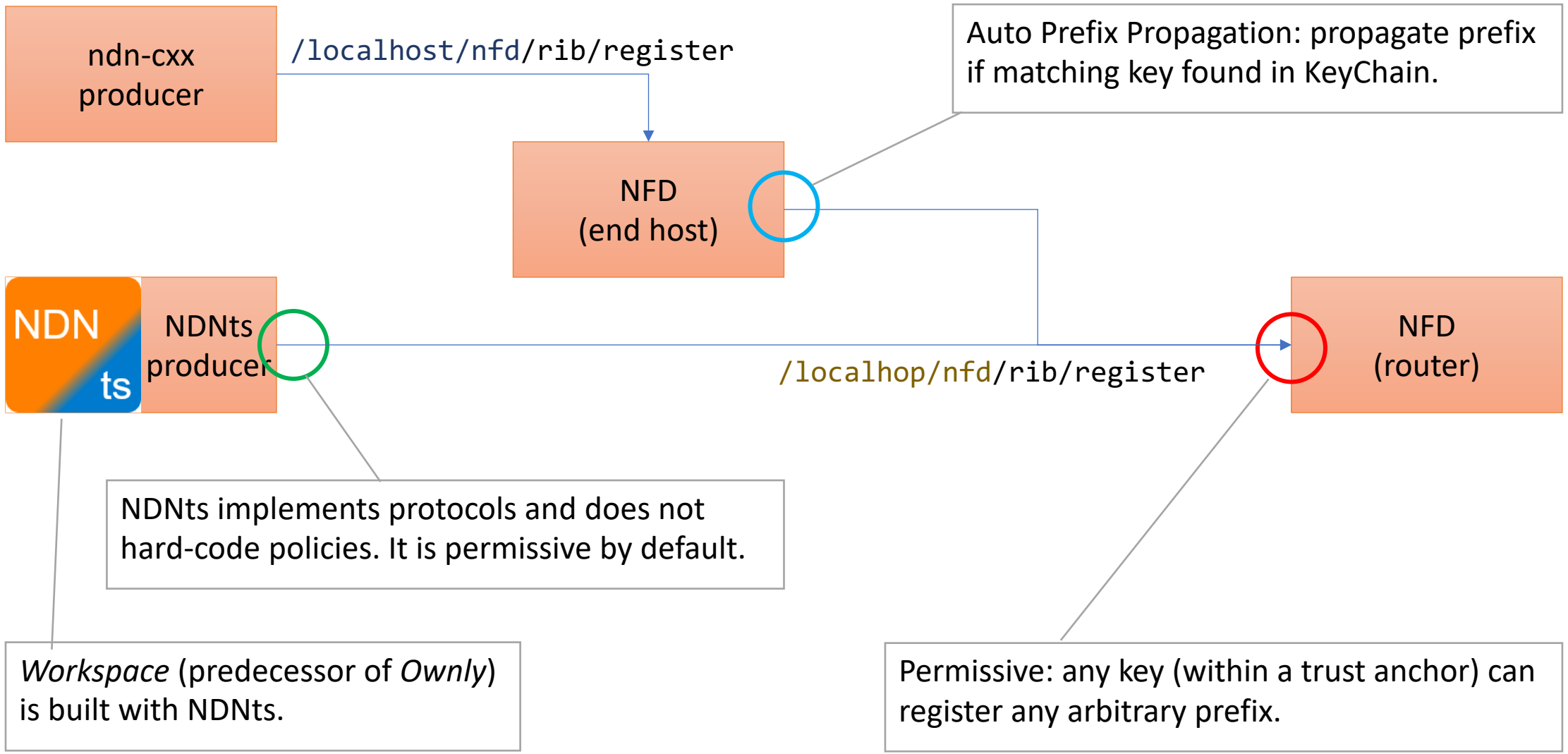
Testbed *vs.* ndn6 Network

	Global NDN Testbed	ndn6 Network
operator	UCLA and Memphis students	@yoursunny
active nodes	19 servers in universities	10 low end VPSs in commercial DCs
node-node tunnels	UDP over IPv4	UDP over IPv6
transport protocols for users		
UDP / TCP	✅	❌
WebSocket	✅	✅
WebTransport	❌	✅
node software and operator tooling		
deployment method	Docker Compose + Jinja2 template	bash + jq + Docker Compose + infoedit
forwarder + routing	NFD + NLSR (link-state mode)	NFD + NLSR (link-state mode)
site cert issuance	HTTP CA with IP address authentication	bash scripts over SSH
user cert issuance	automated NDNCERT CA	manual

Zero-Trust in Prefix Registration

A Decade of #2856

Prefix Registration & Propagation



The "Skeleton Key" Problem

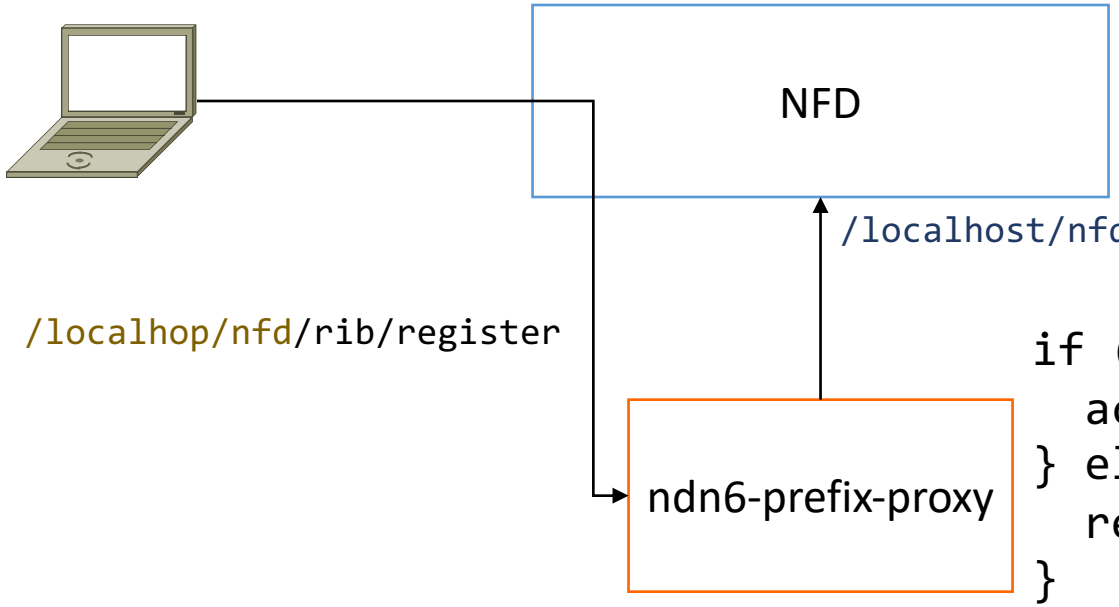
- Given my certificate identity name: /ndn/arizona/shijunxiao:
  - Expected behavior:
    - I can register /ndn/arizona/shijunxiao/files
    - I cannot register /ndn/ucla/adam/files
  - Actual behavior:
    - I can register /ndn/arizona/shijunxiao/files
    - I can also register /ndn/ucla/adam/files ⚠ prefix hijacking
- Why NFD accepts skeleton key?
  - Prefix registration command is encoded as a ControlParameters TLV.
  - ValidatorConfig schema cannot drill into this sub-TLV.

➢ NFD Feature #2856: Confine registered prefix within identity

  - created 2015-June, unresolved for a decade

ndn6 Network Enforces Prefix Confinement

- I built an "enforcer" outside NFD: ndn6-prefix-proxy.
  1. Listen on `/localhop/nfd` command prefix.
  2. Manually decode the command and enforce prefix confinement.
  3. Pass accepted command to NFD `/localhost/nfd`.




```
if (keyLocator.isPrefixOf(params.getName())) {
  accept(); // forward to /localhost/nfd
} else {
  reject(403);
}
```

The Sydney Incident (2024)

1. A *Workspace* user traveled to Australia.
2. Router discovery service guided the user to ndn6's Sydney router.
   - Global NDN testbed does not have a router in Oceania.
3. *Workspace* application sent a prefix registration command:
   - Prefix: `/ndn/irl-workspace/20240525/root/ndn/ucla/lixia`
   - Signed by: `/ndn/ucla/lixia`
4. The command was rejected due to prefix confinement violation.

- *Ownly* inherits the same namespace design.

The Conflict and The Solution Space

	intention	philosophy	prefix registration policy
global NDN testbed	"UCLA experiment"	trust is assumed	anything wanted by application
ndn6 network	"NDN Internet"	trust is a crypto proof	rigid prefix confinement

"I think the network should adapt to app requirement, not the opposite". ~Ownly developer

Sure, but we can have a trade-off:

- Applications should be allowed to keep their namespace designs.
- Network should not suffer from prefix hijacking attacks.

Proposal: Prefix Grant Trust Schema (PGTS)

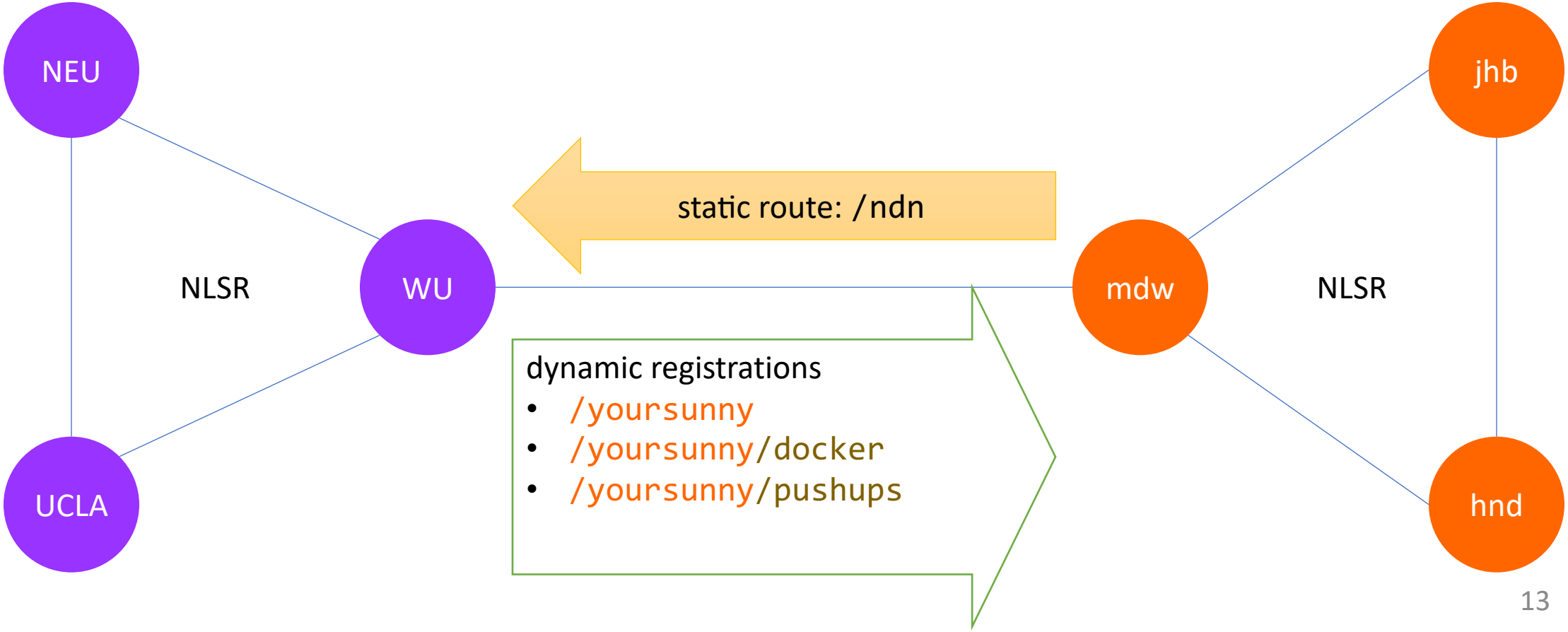
```
Data
  Name: /ndn/ownly.named-data.net/ndncomm2026
        /32=prefix-grant/v=1/seg=0

  Content:
    // Light VerSec trust schema (compiled):
    #user_prefix: #wksp/proj/#user <= #user_cert

  SignatureInfo:
    KeyLocator: /ndn/ownly.named-data.net/ndncomm2026/…

  SignatureValue: …
```

Inter-Domain Routing Policy

A Decade of Policy-Blind Routing

ndn6 – Testbed Peering

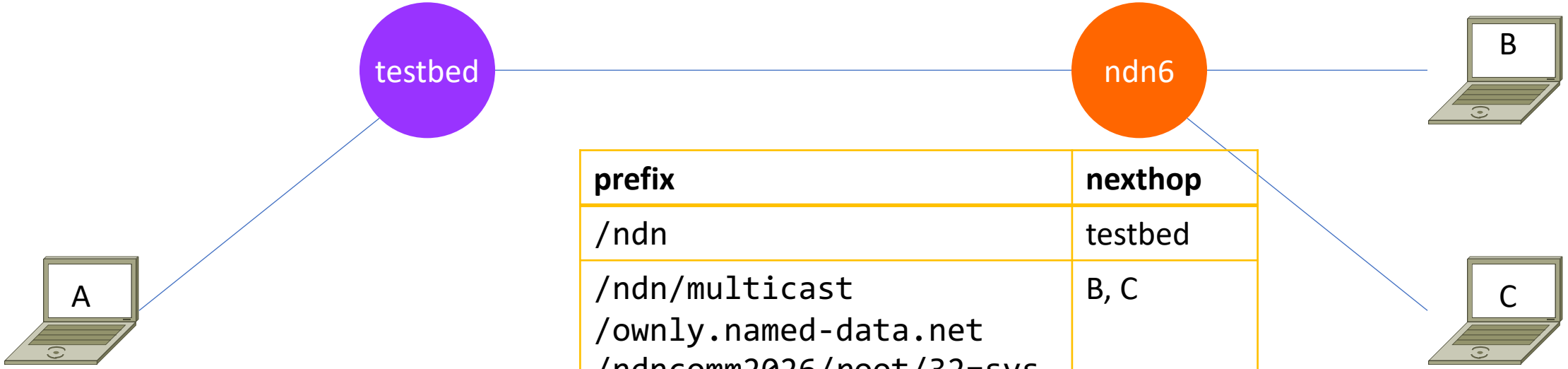
- `ndn6-register-prefix-remote`: **prefix export** only.
  1. Retrieve NLSR NameLSA dataset find names starting with `/yoursunny`.
  2. Send prefix registration commands under `/localhop/nfd/rib/register`



Split-Brain State Breaks State Vector Sync

1. B sends Sync Interest.
2. Longest Prefix Match – Sync Interest goes to C only.
3. A never receives the Sync Interest.

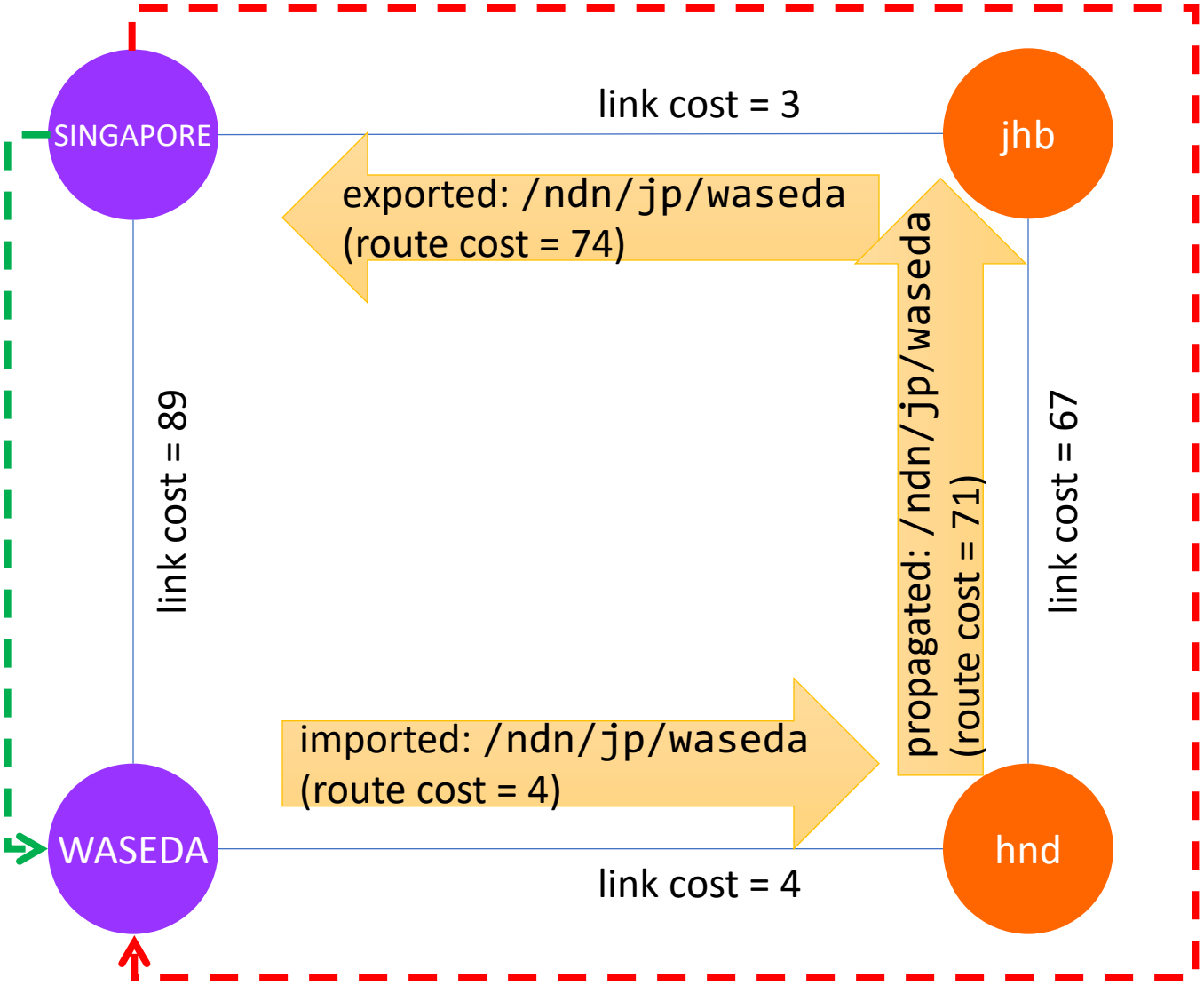
- **Prefix import** needed for routing consistency.



prefix	nexthop
/ndn	testbed
/ndn/multicast /ownly.named-data.net /ndncomm2026/root/32=svs	B, C

Bidirectional Import+Export is Unsafe

route cost = 89
valley-free route



route cost = 74
valley-free violation
/ route leak

BGP Large Community

AS200690
IPv6-only
LowEndASN

```
import filter {
  reject_bogon_routes();
  reject_bogon_asns();
  reject_out_of_bounds_routes();
  reject_rpki_invalid();
  enforce_first_as(52025);
  reject_blocklist();
  bgp_large_community.delete([(200690, *, *)]);
  bgp_local_pref = 80; # pathvector:localpref
  honor_graceful_shutdown();
  bgp_large_community.add((200690,101,52025));
  bgp_large_community.add((200690,101,0));
  accept;
};
```

imports tagged as "provider"

```
export filter {
  remove_private_asns();
  if ((200690,100,0) ~ bgp_large_community) then accept;
  if (net ~ LOCALv6) then {
    bgp_large_community.add((200690,100,0));
    bgp_large_community.add((200690,100,10));
    bgp_large_community.add((200690,100,12));
    accept;
  }
  reject;
};
```

export only "customer" routes

adapted from AS200690 BGP session with Paradox Networks Limited

Proposal: Policy Tagging in NLSR & ndn-dv

```
; NLSR: Name LSA
PrefixInfo = PREFIX-INFO-TYPE TLV-LENGTH
             Name Cost *PolicyTag

; ndn-dv: Prefix Data
PrefixOpAdd = PREFIX-OP-ADD-TYPE TLV-LENGTH
              Name Cost *PolicyTag

PolicyTag = POLICY-TAG-TYPE TLV-LENGTH Name
```

- Operator's peering tool can read the policy tags and make decisions.

Q&A + Links

- Download the slides: https://talks.ndn.today
- In-depth analysis on yoursunny blog:
  - https://yoursunny.com/t/2026/Ownly-ndn6-2856/
  - https://yoursunny.com/t/2026/Ownly-ndn6-policy/
- Peer with AS200690: https://peerwith.me/200690
- Peer with ndn6 network: https://yoursunny.com/p/ndn6/