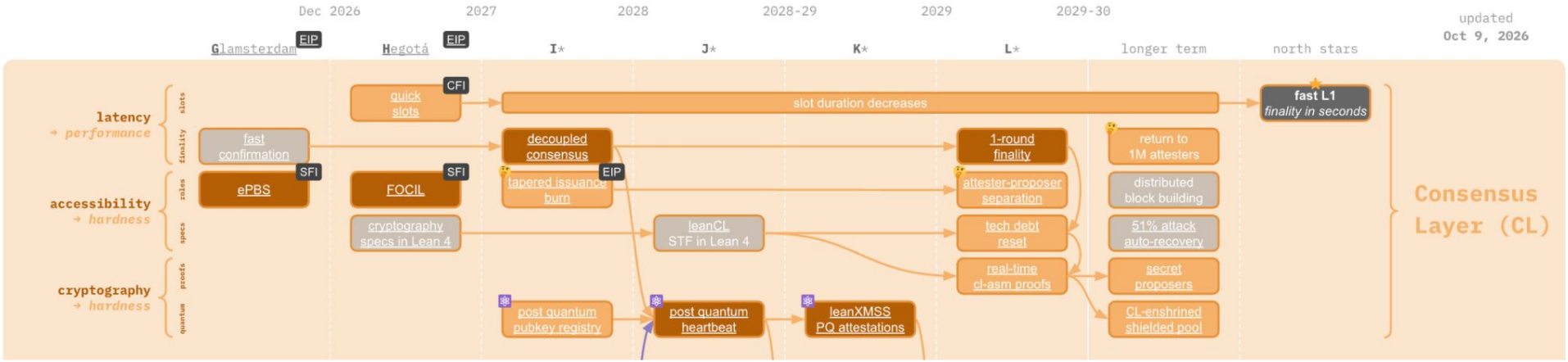
Questions

- What is the target time-to-finality in I*?
- What is the target time-to-finality at the PQ transition? (including attestations)
- What are blockers to having PQ attestestations (“full CL-PQ”) by J*?
- (Bonus) What is the long-term time-to-finality target?

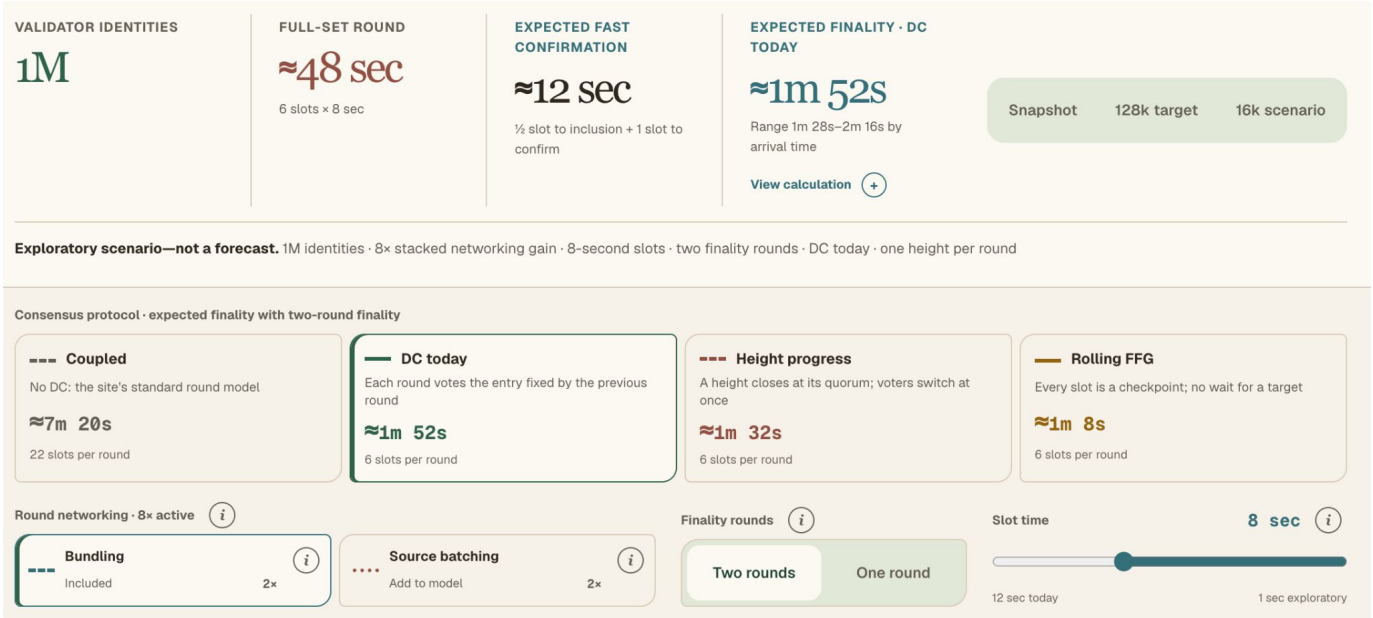
The outlook

- Strawmap: https://strawmap.org/



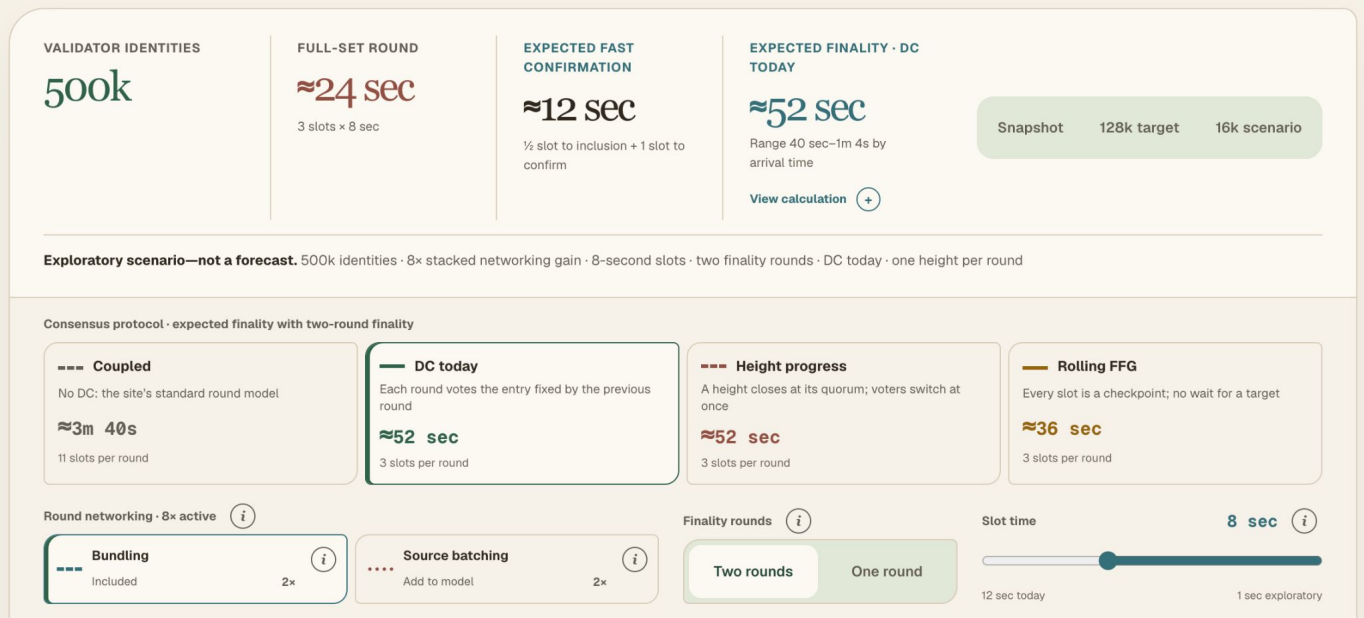
Decoupled consensus in I*

- Finality watch: https://finality.watch/?view=expert&protocol=dc-today



Decoupled consensus in I*

- Finality watch: https://finality.watch/?view=expert&protocol=dc-today



PQ questions

- How many max concurrent aggregators can we expect/do we want to expect?
  - I assume 128 aggregators at peak
    => 8 subnets per layer max if we want 16 aggregators per subnet
  - If we want more (for higher QoS, lower latency etc.) we need to convince ourselves some mechanism delivers the right number
- How many validators can we expect/do we want to expect?
  - We're floating the ability to support 1M.
    Reminder: We have 7-8k staking nodes on the network.
- How much engineering progress can we expect and by when?
  - Signature aggregation per second + SNARK aggregation per second
- PQ simulator: https://kamilsa.github.io/pq-sig-agg-model/
  - Must multiply results by two, unless one-round finality

PQ Signature Aggregation Time Model

Simplified model — ignoring gossipsub overhead & inter-node latencies

BEAMSIM Default | Optimistic | Conservative | Deep Tree

NETWORK

Total Validators: 1002496

Node Bandwidth: 50

SIGNATURE & SNARK

Signature Size (bytes): 1024

SNARK Size (KB): 220

Sigs Agg Rate (/s): 1200

SNARKs Agg Rate (/s): 5

THRESHOLDS

Subnet Sigs Threshold (%): 100

Global Threshold (%): 100

LAYER CONFIGURATION

Auto-optimize

Aggregation Layers: 3

- L1 — Subnets: 8
- L2 — Aggregators: 8
- Global — 1 aggregator: ×1

LAYER BREAKDOWN

	Download	Aggregate	
L1 — Subnet 8 aggregators	20.531 s 125312 sigs × 1.0KB	104.427 s 125312 sigs @ 1,200/s	124.958 s
L2 — Intermediate 1 8 aggregators · 1 children each	36.0 ms 1 SNARKs × 220KB (bottleneck)	200.0 ms 1 SNARKs @ 5/s	236.0 ms
L3 — Global 1 aggregator	288.4 ms 8 of 8 SNARKs × 220KB	1.600 s 8 SNARKs @ 5/s	1.888 s

TOTAL LATENCY **127.082 s** > slot time ✕

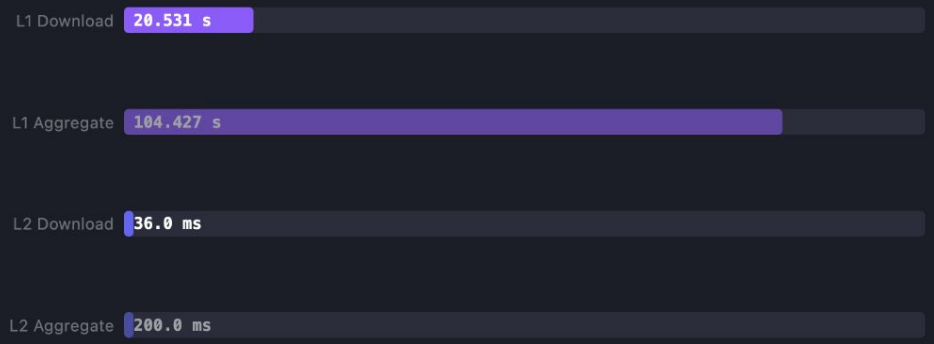
NETWORKING **20.856 s**

COMPUTE **106.227 s**

DERIVED VALUES

Validators / subnet	125,312
Sigs needed / subnet	125,312
Tree (nodes/layer)	8 → 8 → 1
SNARKs to global	8 of 8
Validators proved	1,002,496 (100%)

TIME BREAKDOWN



PQ Signature Aggregation Time Model

Simplified model — ignoring gossipsub overhead & inter-node latencies

BEAMSIM Default | Optimistic | Conservative | Deep Tree

NETWORK

Total Validators: 128512

Node Bandwidth: 50

SIGNATURE & SNARK

Signature Size (bytes): 1024

SNARK Size (KB): 220

Sigs Agg Rate (/s): 1200

SNARKs Agg Rate (/s): 5

THRESHOLDS

Subnet Sigs Threshold (%): 100

Global Threshold (%): 100

LAYER CONFIGURATION

Auto-optimize

Aggregation Layers: 3

- L1 — Subnets: 8
- L2 — Aggregators: 8
- Global — 1 aggregator: ×1

LAYER BREAKDOWN

	Download	Aggregate	
L1 — Subnet 8 aggregators	2.632 s 16064 sigs × 1.0KB	13.387 s 16064 sigs @ 1,200/s	16.019 s
L2 — Intermediate 1 8 aggregators · 1 children each	36.0 ms 1 SNARKs × 220KB (bottleneck)	200.0 ms 1 SNARKs @ 5/s	236.0 ms
L3 — Global 1 aggregator	288.4 ms 8 of 8 SNARKs × 220KB	1.600 s 8 SNARKs @ 5/s	1.888 s

TOTAL LATENCY **18.143 s** > slot time ×

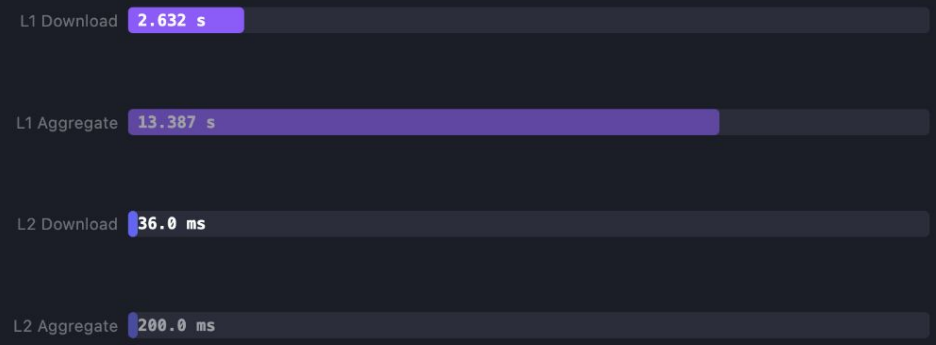
NETWORKING 2.956 s

COMPUTE 15.187 s

DERIVED VALUES

Validators / subnet	16,064
Sigs needed / subnet	16,064
Tree (nodes/layer)	8 → 8 → 1
SNARKs to global	8 of 8
Validators proved	128,512 (100%)

TIME BREAKDOWN



What are blockers to having PQ attestations in J*?

- **Thesis:** We should at least consider it.
  Timelines to PQ remain uncertain + very clear community desire to see this prioritised.

- Validator set size
  - Would we have to increase time-to-finality?
  - Do we have enough aggregators?
- Are we sufficiently happy with the cryptography?
- Are we happy with VRF/whatever we need for finality?
  - Do we need something here? We can just rotate attesters deterministically?

What should be on the table, imo, for *really* fast finality (< 10s)

- We may not need any of these. We'd know only by setting targets for finality.
  But if we do want some of this stuff, we may want to introduce early(er).
- Small stakers are not pivotal for finality. Do not design for their participation, without excluding it either.
  We can have a permissive set, but not having fast finality is also a security tradeoff.

- **Attester cap:** Limit the number of attesters. Best to introduce progressively?
- **Attester-includer separation:** Attesters are includers too, but more includers exist.
- **Increased attester requirements:** This is a role paid by the protocol. Large attesters already incur linear cost (e.g., more blob custody). Full nodes remain cheap.
- **Directed attester topology:** Attesters are assumed more professionalised.