Super Validator Health — methodology

CONTENTS

1 The short version
2 Endpoint probes: what we send and what we expect
3 Our terms (not Canton or Splice definitions)
4 Endpoint probes: how the figures are calculated
5 Status reports: source data and formulas
6 Getting the data

1 The short version

Two independent sources, shown as two tabs. Only the first produces uptime and quorum figures.

TAB 1 · FROM THE LEDGER

Status reports

- Every SV app writes an on-ledger `SvStatusReport` about every 140 s.
- **Reporting**: latest accepted report under 5 min old.
- **Heartbeat %** = time reporting ÷ time as a DSO member.
- All uptime, quorum and gap figures come from here.

TAB 2 · OUR OWN REQUESTS

Endpoint probes

- Every 30 s: Scan, sequencer, BFT peer and SV app of each SV.
- Each answer is classified by status codes, content and response time (section 2).
- A refusal (401 / 403, PERMISSION_DENIED) is **restricted**, never a failure.
- **Diagnostics only**: a failed probe can be our route, so probes never enter the ledger figures.

The 5-minute rule and the probe thresholds are ours, not Canton or Splice definitions (section 3).

2 Endpoint probes: what we send and what we expect

Every 30 s, all SVs in parallel, four requests per SV. **Timeouts**: HTTP 10 s in total; gRPC up to 10 s to connect, then up to 10 s for the call. **Response time**: gRPC, the call alone; HTTP, including connection setup when a new connection is needed. Scan is timed through our proxy.

ENDPOINT	WHERE ITS URL COMES FROM (RE-READ EVERY 10 MINUTES)
Scan	`GET /api/scan/v0/dso` → `.sv_node_states[].contract.payload.state.synchronizerNodes[][1].scan.publicUrl`
Sequencer	the same node → `.physicalSynchronizers[]`, a list of [serial, entry] pairs → the entry of the active serial → `.sequencer.url`. Active serial: `GET /api/scan/v0/active-synchronizer-serial`
BFT peer	`GET /api/scan/v0/sv-bft-sequencers` on that SV's own Scan → `.bftSequencers[]` with `serialId` = active serial → `.p2pUrl`
SV app	not read from Scan: derived from the Scan host, `scan.<domain>` → `sv.<domain>`

Every answer becomes one of five results

RESULT	ON THE PAGE	COUNTED AS
answered	response time, green cell	answered; in the response times
slow · stale	response time in amber; amber cell	answered; in the response times; logged
failed	“down”; orange to red cell	failed; logged
restricted	lock; hatched grey cell	neither; left out of “Answering now”
unconfirmed	“unverified” or a dash; dotted cell	neither; stays in the “Answering now” denominator

2.1 Scan

GET `/api/scan/v0/events/latest-record-time` — HTTP · through our Scan proxy, pinned to this SV's Scan

REQUEST no parameters, no body

EXPECTED `200` `{"record_time":"2026-11-16T13:45:14.479003Z"}`

CODE	WHAT CAME BACK	WE RECORD
`200`	`record_time` at most 5 min behind our clock	answered · slow if over 3 s
`200`	`record_time` more than 5 min behind	stale "Scan data N s behind"
`200`	`record_time` missing or unreadable	answered freshness unknown
`other 2xx`	e.g. 204; freshness is not checked	answered
`401` `403`	access refused	restricted
`other 3xx · 4xx`	any other answer, e.g. 404 or 429	unconfirmed
`5xx`	error from the service or its ingress	failed
`Proxy`	not forwarded: our proxy's check found the Scan lagging	stale "Scan lags the network"
`Proxy`	not forwarded: our proxy's check of the Scan failed	failed "upstream is unhealthy"
`No answer`	timeout, refused or reset connection, DNS, TLS	failed

Proxy rows: our proxy's own health check had this Scan out of rotation, so no request reached it. Stale counts as answered; only the first row confirms fresh data.

2.2 Sequencer

POST `/grpc.health.v1.Health/Check` — gRPC over HTTP/2, TLS · on the SV's sequencer URL

REQUEST `content-type: application/grpc` · `te: trailers` · body `00 00 00 00 00` (empty request)

EXPECTED `gRPC 0` with `:status 200` and body `00 00 00 00 02 08 01` (health status SERVING)

CODE	WHAT CAME BACK	WE RECORD
`gRPC 0`	body: health status SERVING	answered · slow if over 3 s
`gRPC 0`	body: health status NOT_SERVING	failed
`gRPC 0`	body: UNKNOWN, empty or unreadable	unconfirmed
`gRPC 7`	PERMISSION_DENIED	restricted
`gRPC 16`	UNAUTHENTICATED; also any HTTP 401 / 403	restricted
`gRPC 12`	UNIMPLEMENTED: no health service at this URL	unconfirmed
`gRPC 14`	UNAVAILABLE; also any HTTP 5xx	failed
`other gRPC`	any other status, e.g. 5 NOT_FOUND	failed
`no gRPC status`	not gRPC, or an incomplete gRPC answer	unconfirmed
`No answer`	timeout, refused or reset connection, DNS, TLS	failed
`No answer`	timeout or reset, and the endpoint never answered us	unconfirmed

Priority: no answer → 7, 16, 401, 403 → 14, 5xx → the other rows. So an HTTP 503 without a gRPC status is failed.

2.3 BFT peer

POST `/grpc.health.v1.Health/Check` — gRPC over HTTP/2, TLS · on the SV's p2pUrl

REQUEST the same as 2.2

EXPECTED `gRPC 12` "Method not found": this port has no health service, so SERVING is not expected.

CODE	WHAT CAME BACK	WE RECORD
`gRPC 12`	`:status 200`, message starts "Method not found"	answered response time only
`gRPC 12`	any other UNIMPLEMENTED	unconfirmed
`gRPC 7`	PERMISSION_DENIED: **every MainNet SV today**	restricted
`gRPC 16`	UNAUTHENTICATED; also any HTTP 401 / 403	restricted
`gRPC 0`	a health status, should one ever come	as in 2.2
`gRPC 14`	UNAVAILABLE; also any HTTP 5xx	failed
`other gRPC`	any other status	failed
`no gRPC status`	not gRPC, or an incomplete gRPC answer	unconfirmed
`No answer`	timeout, refused or reset connection, DNS, TLS	failed
`No answer`	timeout or reset, and the endpoint never answered us	unconfirmed

Priority as in 2.2; the first gRPC 12 row is checked before the second.

"Answered" is a heuristic. gRPC 12 "Method not found" is what a grpc-java server replies, so a gRPC endpoint is reachable at the peer URL. Not verified: that the BFT node itself answered, peer-to-peer connectivity between SVs, authentication, ordering or block progress.

Colours in the BFT grid. Dark grey: answered (response time only). Hatched: restricted. Orange: two or more failed probes in that interval: the endpoint timed out, refused the connection or returned 5xx instead of its usual answer (87 probes, 23 Oct to 15 Nov). An orange column across SVs is a network event, not separate outages: on 27 Oct, 15:35 UTC, all endpoints of the six SVs hosted on Google Cloud failed for about two minutes.

Resolved: Five North 1, 27 Oct. Its endpoint answered us, then reset our connections while access was being restricted, then returned 403. We showed that as failed (red). Fixed: a refusal now closes an incident, and that stretch is recorded as restricted.

2.4 SV app

GET `/api/sv/version` — HTTPS · on sv.<sv-domain>

REQUEST no parameters, no body

EXPECTED `2xx` The body is not inspected.

CODE	WHAT CAME BACK	WE RECORD
`2xx`	any success	answered slow if over 3 s
`401` `403`	access refused	restricted
`other 3xx · 4xx`	any other answer, e.g. 404 or 429	unconfirmed
`5xx`	error from the service or its ingress	failed
`No answer`	timeout, refused or reset connection, DNS, TLS	failed
`No answer`	timeout or reset, and the endpoint never answered us	unconfirmed

3 Our terms not Canton or Splice definitions

TERM	RULE	READ IT AS
SV state “now”	In order. **Unknown**: our data is over 5 min old. **Down**: no accepted report for over 5 min. **Degraded**: reporting, but lag over 30 s, 2+ rounds behind, or a probed endpoint down or slow. **Up**: otherwise.	Down = the SV app's heartbeat is missing (two missed reports; they come about every 140 s). Participant, mediator and sequencer may still work. Degraded is the only place probes touch the Status reports tab.
Heartbeat %	`1 - (time in gaps ÷ time as a DSO member)`. Gap: from 5 min after a report to the next one.	Reporting, not transaction processing. Problems under 5 min leave no trace.
Period below the quorum	Fewer than 9 of 13 SVs reporting.	Not an outage: an SV app can stop reporting while its participant still confirms.
DSO activity	At least one report accepted in the previous 5 min.	Each accepted report needed the DSO's confirmation threshold.
Ledger incident	10 min or more between two reports, or mediator / participant lag over 300 s.	Listed as “Participant”: the signal, not the part that failed. Lag is self-reported.
Probe “failed”	5xx, UNAVAILABLE, NOT_SERVING or no answer. For Scan also our proxy's verdict.	Can be our route or our proxy: 647 of 702 failed Scan probes were the proxy's verdict. One failure is a blip; an incident needs two.
Restricted, unconfirmed	Refusals, and answers without a confirmed health status.	Neither up nor down: we do not guess.
Slow, stale	Probe: over 3 s. Endpoint now: median 3× its usual and 100 ms more. Stale: Scan data over 5 min behind.	Measured from Helsinki: includes the distance, and for Scan our proxy.

4 Endpoint probes: how the figures are calculated

ITEM	RULE
Counting	Per SV, endpoint, observer and UTC quarter hour, by result (normally 30 probes). Failed and slow probes are also kept one by one, with their HTTP / gRPC code.
Response time	Answered probes only. Shown: the median of the latest quarter hour. “Usual”: the median of the quarter-hour medians over 7 days.
Scan p50 / p95 / p99	All answered Scan probes of all SVs in the period. “≈” = read from daily histograms (periods starting over 13 days ago), within about 19%.
Endpoint state “now”	Latest quarter hour. **Down**: open incident, or over half of the probes failed. **Restricted / unconfirmed**: nothing answered. **Slow**: a probe failed, over half of the answers slow or stale, or median 3× its usual and 100 ms more. **Up**: otherwise.
Answering now	Up or slow ÷ not restricted. Unconfirmed stay in the denominator. “0 of 0”: every SV restricts that endpoint from us.
Probe incident	Opens on two failed probes with no good answer between; starts at the first. Closes on two good answers in a row; ends at the first. A refusal (403) counts as good.
Our own outage	Each round also calls a neutral control URL; if it fails, the round's direct probes are discarded. Scan probes are discarded when our proxy does not answer.

What we receive, MainNet, 23 Oct to 15 Nov 2026

ENDPOINT	PROBES	ANSWERED	FAILED	RESTRICTED	TYPICAL RESPONSE
Scan	847,821	99.92%	0.083%	0	59 ms
Sequencer	847,808	99.95%	0.052%	0	58 ms
BFT peer	847,073	10.0%	0.010%	89.9%	105 ms
SV app	847,808	99.91%	0.093%	0	59 ms

Our observer at our validator's address, which the SVs allowlist. BFT peer: 6 of 13 answered at first; all 13 return PERMISSION_DENIED since 31 Oct. No probe data 15 Nov 08:35 to 16 Nov 14:16 UTC (grey in the grid).

5 Status reports: source data and formulas

ITEM	RULE
Source	Every `Splice.DSO.SvState:SvStatusReport` on the ledger, with its record time (MainNet: since 24 Jun 2024). A contract without a status is not a heartbeat.
Fields used	`svName`, `status.createdAt`, `status.mediatorSynchronizerTime`, `status.participantSynchronizerTime`, `status.latestOpenRound.number`
DSO membership	From the `DsoRules` contract (its `svs` list) at each moment. Silence never ends membership.
Quorum	`f = floor((n - 1) / 3)`, `q = ceil((n + f + 1) / 2)`, as Splice computes it. With n = 13: f = 4, q = 9. Recomputed for the member count at each moment.
Time with ≥ q SVs reporting	Time with at least *q* members reporting ÷ time with data, computed over the exact interval boundaries, not by sampling.
Mediator / participant lag	`status.createdAt` minus the synchronizer time in the same report. Over 30 s: slow. Over 300 s: incident. A time at the epoch = unknown, never counted against the SV.

6 Getting the data

Open API, no key; also behind the **Download data** button on the page. Times are UTC.

```
GET https://cc.itrocket.space/api/sv-health/raw
    ?kind=probes&observer=validator-mainnet&service=scan&from=2026-10-23&to=2026-11-15&format=csv
```

KIND	ONE ROW IS	COLUMNS	KEPT SINCE
gaps	One stretch over 5 min without an accepted report from one SV	`sv, last_report_at, started_at, ended_at, next_report_at, seconds`	24 Jun 2024
incidents	One incident as the page lists it	`sv, component, started_at, ended_at, seconds, detail`	24 Jun 2024
probes	One SV, endpoint, observer and quarter hour: probes counted by result	`bucket, sv, service, observer, probes, ok, slow, stale, down, restricted, unverified, unreachable, median_ms`	23 Oct 2026
failures	One failed or slow probe	`at, sv, service, outcome, error, ms, observer, http, grpc`	last 90 days

`observer`: without it the API returns the rows of every observer, while the page shows one; section 4's figures are those of `validator-mainnet`. Also: `from` / `to` (UTC days, at most 31), `sv`, `service` (`scan`, `sequencer`, `bft`, `sv`), `source` (`ledger`, `probes`), `format` (`json`, `csv`, `tsv`, `ndjson`). At most 100,000 rows per request; the header `X-Next-From` says where to continue. The page's own figures: `GET /api/sv-health?range=24h|7d|30d|90d|1y|all`.