Beyond Purdue: The Data Plane

Concept V1.0 | 3 November 2026 | Petr Roupec, BM Global | INTERNAL: draws on principles from unfiled work

1. Purpose

This note describes the data plane: the L3 Data nodes and the names that find them. It is a concept for discussion and a bench prototype, not a design. Names, addresses and plant names are examples (plant.example, RFC 2606).

The note follows the order of the works: the laws first, then data in independent nodes, then names and aliases. Everything else is a result of these three.

2. What we learned from the sources

Source	What it says	What we take
EP3734376 B1, Hovorka, Kudelka, Roupec (granted 2021)	Logic is taken out of the devices; devices only set and read ports. A shared automation context holds the knowledge and distributes it. Bridges convert signals (volts to degrees). Each end device has a digital twin. An electrician maps devices, no programmer.	The data plane is the plant context. Edge adapters are the bridges. One node per signal or asset is the twin. Deployment without programmers. Commands are not taken over: they stay on the L2 in lane.
1001 Meaning Tags with Historical Data V2 and 110 Deployment Plan (2024)	Autonomous meaning-tag nodes with their own history, edge conversion, DNS with SRV and TXT, aliases, KKS, hierarchical IPv6.	The node, the aliases and the hierarchy stay. The heavy parts change (section 20).
Meaning Tags for Conversational AI	A meaning tag is text from the operator manual. Value changes send messages into the context. Universal tag names.	Every node carries its meaning text and shouts meaning events, ready for the AI and the operator.
Beyond Purdue #058 Meaning Tags	Process close to the source. Raw data only, no historian compression.	History is raw and lossless, kept in the node.
DEWA data conversion, layer scripts (R)	Layer 0 raw data, layer 1 physical daily events, layer 2 unusual for this tag, layer 3 plant-wide reasoning, layer 4 LLM interface. The event generator writes tokens and an LLM explanation per tag and day.	The pre-analysis runs inside each node; the AI receives ready explanations.
Daily notes 27 October 2026	Two planes. Names identify, never authorize. Assets are nodes.	Names carry identity; keys authorize writes into the data plane only.
Unfiled one-way work (internal)	Principles only: one-way by construction, the sender need not know the receiver, repetition replaces acknowledgement, the receiver counts sequence numbers, opposite directions share nothing.	Every message in the data plane is a one-way shout. No technical detail of the disclosures is used.

3. The laws first

The laws of authority decide what each level is allowed to do. The data plane is L3: it processes, stores and presents data on demand, and records every command, permit, lock and bypass. It grants no access and takes no action. Protection stays hardwired, and a lock outranks every level. Everything in this note obeys these laws; nothing in it gives the data plane authority over the process.

4. Data in independent nodes

The second idea: data lives in independent nodes, not in one big database. Each node has a meaning. Everything that follows is a result of this. The technology is technology people already know and work with: DNS, IPv6, multicast, UDP and OPC UA.

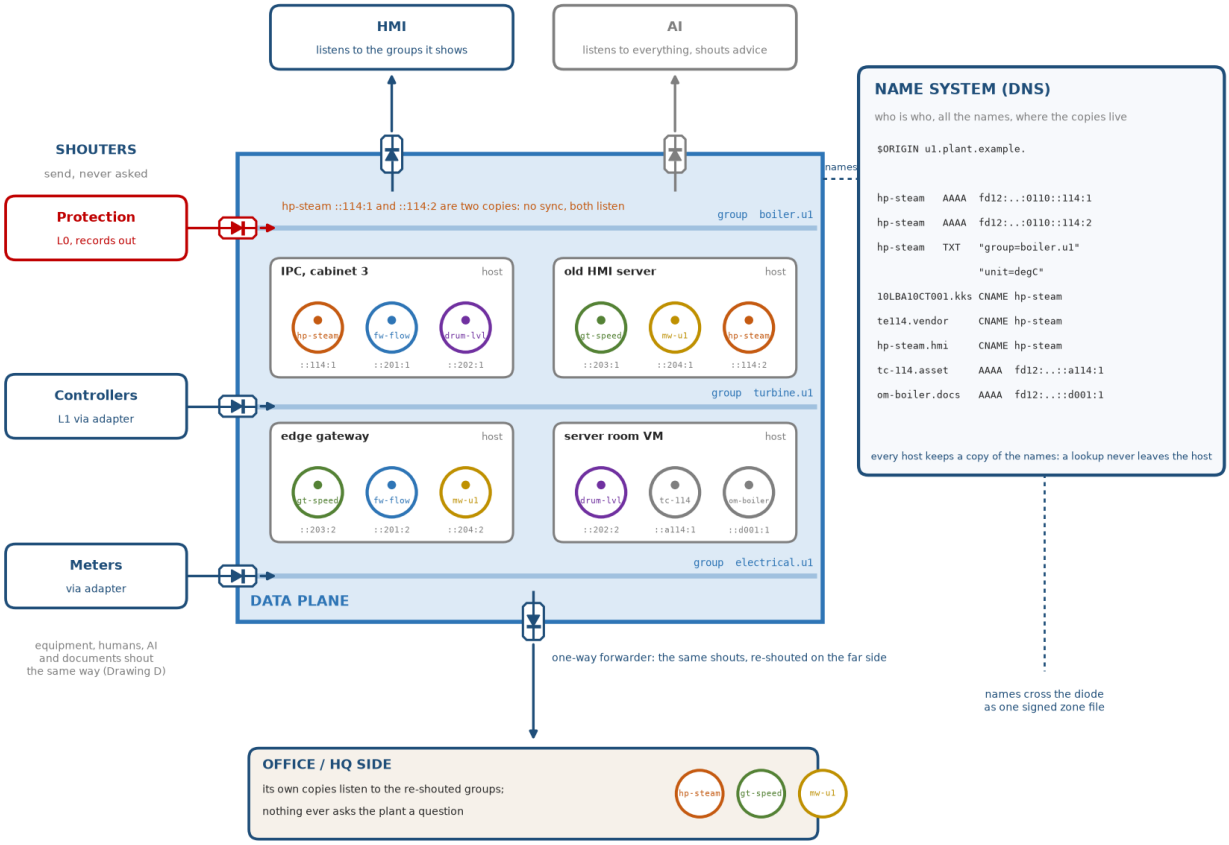
The data plane is the context of the power plant. It contains only messages, and every message is a one-way shout: nobody asks, nobody answers, nobody acknowledges. Each node listens for its own data, keeps it, and shouts what it knows.

There are two main components:

- **Names.** Standard DNS. It is already developed, works everywhere, and is distributed by design. It gives one thing as many names as people need (KKS, vendor tag, HMI name, meter number, manual reference) and points them all to the same node.
- **Data.** Very small, independent nodes. Each one holds the real-time value, the history, the alarms and the meaning of one signal, asset or document. It has its own IPv6 address and runs on any host in the company, with at least two copies.

Drawing E. The data plane: names and data

Two components: the name system says who is who; the nodes hold the data. Everything moves as one-way shouts.

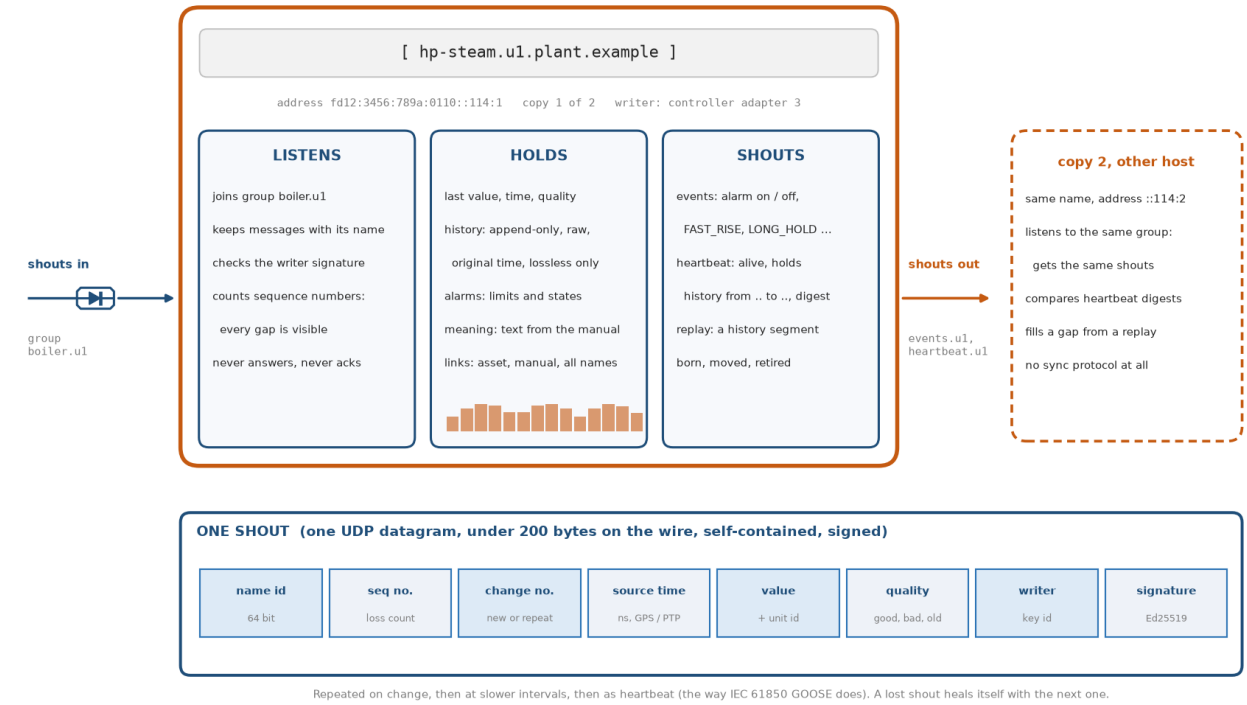


Drawing E. Shouters send through diodes into system groups; small nodes live on any host, two copies each; DNS holds the names; names cross the diode as one signed zone file.

5. The node

A node is one small process (or one task inside a shared host process), a few megabytes at most. Many nodes run on one host. Its parts:

- **Identity.** Canonical name, IPv6 address, the key of the writer it accepts.
- **Listens.** Joins the multicast group of its system and keeps only messages with its own name. It checks the writer signature and counts sequence numbers, so every gap is visible.
- **Holds.** Last value with time and quality; append-only raw history with the original source time; alarm limits and states; meaning text from the operator manual; links to the asset, the manual and all its other names.
- **Shouts.** Events (alarm on and off, FAST_RISE, LONG_HOLD), a heartbeat with what it holds (history from and to, digest), history segments for whoever needs them, and born, moved or retired.



Drawing F. One node: it listens, holds and shouts. Copy 2 on another host hears the same shouts. One shout is a single signed UDP datagram.

6. Names and aliases

Naming and asset management is always the big issue. That is why the names come right after the node. Standard DNS is used (RFC 1034, https://www.rfc-editor.org/rfc/rfc1034) because people already read names like these, DNS has aliases, and a name is updated in three ways: automatically when a node is born, by a person, or by a shout. All three end as one signed change from the registrar. For each node:

- **Canonical name** in a hierarchy: signal, system, unit, plant, for example hp-steam.u1.plant.example.
- **AAAA records,** one per copy (RFC 3596, https://www.rfc-editor.org/rfc/rfc3596).
- **Aliases** as CNAME records: KKS, vendor tag, HMI name, meter number, OPC UA name. Any of them finds the same node.
- **TXT and URI records** for the group, the unit, the writer key and links to the manual (URI, RFC 7553, https://www.rfc-editor.org/info/rfc7553).
- **Browsing.** HMI and AI list what exists with DNS service discovery (RFC 6763, https://www.rfc-editor.org/rfc/rfc6763), with multicast DNS on a bench (RFC 6762, https://www.rfc-editor.org/rfc/rfc6762).
- **Where DNS runs.** Every host runs a small standard DNS server with the full zone. All lookups go to the host itself, so a lost network does not lose names. 100,000 names at about 100 bytes each is about 10 MB (estimate).
- **Who keeps it.** One registrar holds the master list, a hidden primary running in two copies like every node. Only the registrar signs changes.
- **How it is kept: by shouts.** The registrar shouts every signed change with a serial number on the names group. A small listener beside each DNS server applies it. The whole zone is repeated slowly as a carousel, so a host that missed a serial catches up without asking. Standard pull transfer with NOTIFY (RFC 1996, https://www.rfc-editor.org/rfc/rfc1996) would work inside one side, but not through a diode; shouts give one mechanism on both sides.
- **Readers notice nothing.** HMI, AI and tools make ordinary DNS lookups.
- **Signed.** The zone is signed with DNSSEC (RFC 9364, https://www.rfc-editor.org/info/rfc9364) and carries a zone digest (ZONEMD, RFC 8976, https://www.rfc-editor.org/rfc/rfc8976.html). Across a diode the whole zone is shouted as one file and checked on arrival.

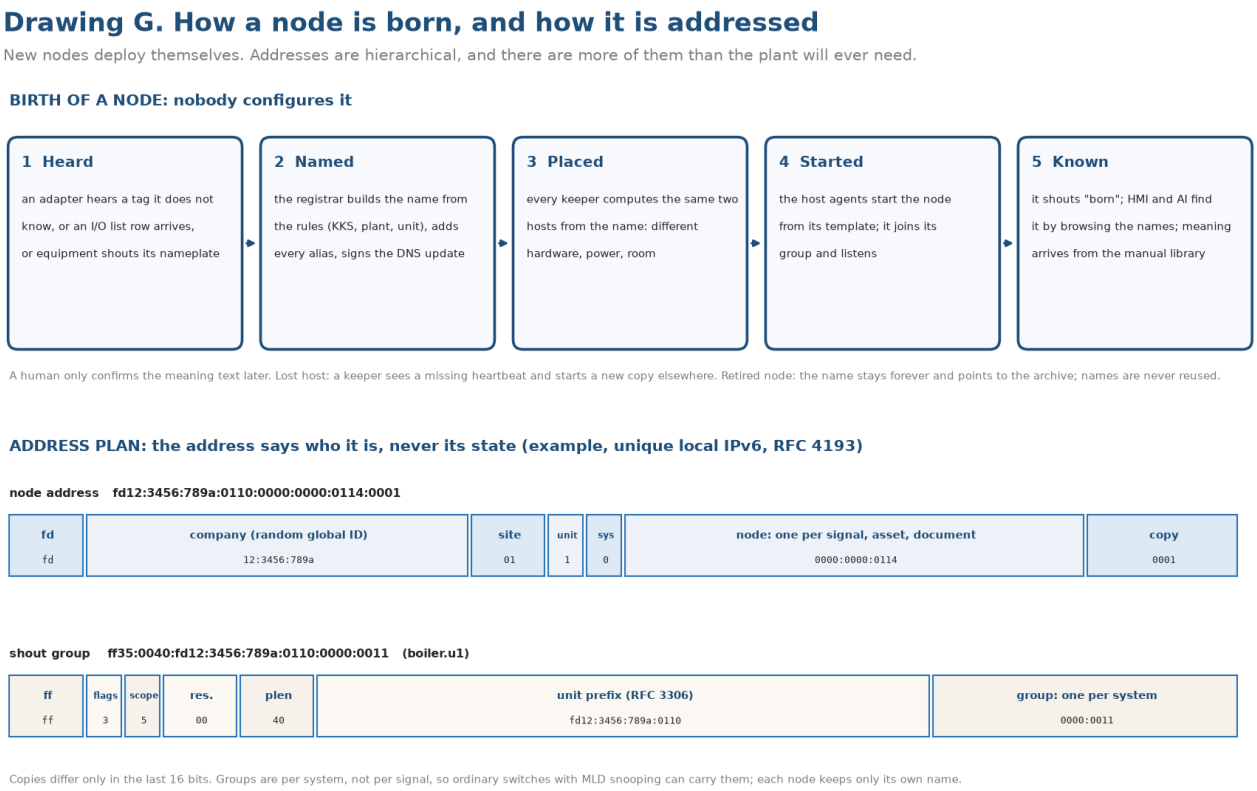
Example zone lines:

```
hp-steam.u1.plant.example.    AAAA  fd12:3456:789a:0110::114:1
hp-steam.u1.plant.example.    AAAA  fd12:3456:789a:0110::114:2
hp-steam.u1.plant.example.    TXT   "group=boiler.u1" "writer=ctl-adapter-3"
10LBA10CT001.kks.plant.example.  CNAME  hp-steam.u1.plant.example.
te114.vendor.plant.example.      CNAME  hp-steam.u1.plant.example.
asset-4711.cmms.plant.example.   CNAME  dd01.u1.plant.example.
```

7. IPv6 addresses

Unique local addresses (RFC 4193, https://www.rfc-editor.org/rfc/rfc4193): fd, a random 40-bit company ID, 16 bits for site, unit and system, 48 bits for the node and 16 bits for the copy. Groups use unicast-prefix multicast addresses (RFC 3306, https://www.rfc-editor.org/rfc/rfc3306), one per system. The address says who a node is, never its state. This corrects the 2024 idea of alarm subnets: an alarm is an event, not a place.

Groups are per system, not per signal, because ordinary switches keep a limited number of groups for MLD snooping (MLDv2 RFC 3810, https://www.rfc-editor.org/rfc/rfc3810; snooping RFC 4541, https://www.rfc-editor.org/rfc/rfc4541). The limit must be tested on the real switches.



Drawing G. A node is born without people, and its address follows the plant hierarchy. Copies differ only in the last 16 bits.

8. At least two copies

Why two copies of everything? Because one copy is never enough. Redundancy is not an option added later; it is how every node exists.

- **Co-listening replaces sync.** Both copies join the same group and hear the same shouts, so they hold the same data without talking to each other.
- **At least two copies, to be really redundant.** Always on different hardware, power supplies and rooms. Losing one copy loses nothing. Three for important signals.
- **Gaps.** Copies compare the digest in each other's heartbeat. A copy with a gap takes the missing segment when the other shouts it on a repair group. This happens only within one side of a diode.
- **Moving.** A node can move to another host and keep its address and name.

9. Messages are shouts

- **Self-contained.** One UDP datagram per value: name id, sequence number, change number, source time (GPS or PTP), value and unit, quality, writer key id, signature (Ed25519, RFC 8032, https://www.rfc-editor.org/rfc/rfc8032). Under 200 bytes on the wire.
- **Repetition instead of replies.** A change is shouted at once, then repeated at growing intervals, then as heartbeat. IEC 61850 GOOSE works the same way: multicast, no acknowledgement, a state number and a sequence number, a

time allowed to live (https://cdn.selinc.com/assets/Literature/Publications/Technical%20Papers/6921_IEC61850Network_MS_20190712_Web.pdf, https://pmc.ncbi.nlm.nih.gov/articles/PMC7926645/). A lost shout heals itself with the next one.

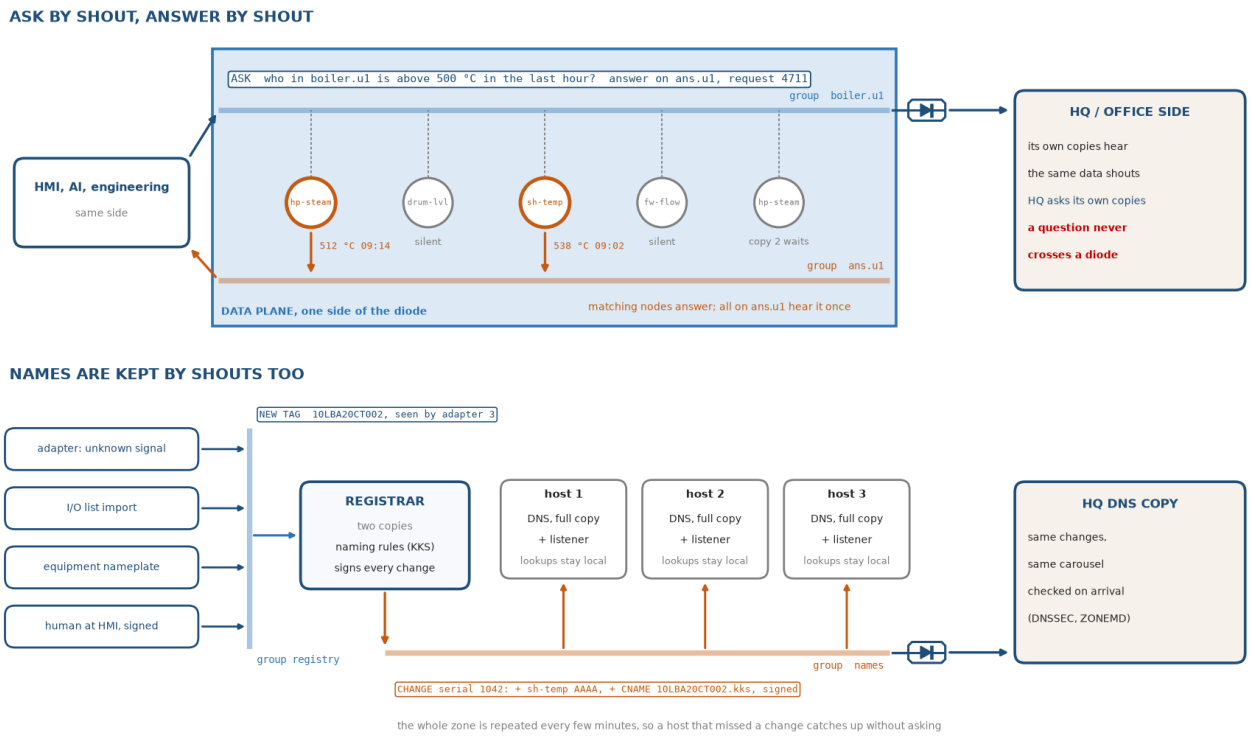
- **Large objects.** History segments, manuals and zone files are shouted with forward error correction, as FLUTE (RFC 6726, https://www.rfc-editor.org/rfc/rfc6726) with RaptorQ (RFC 6330, https://www.rfc-editor.org/rfc/rfc6330), repeated in a carousel.
- **Proof stays local.** The sender keeps its own record of what it shouted; the receiver keeps its own count. Nobody asks the other.

10. Asking a node

Every node has an API, but the API is made of shouts. A question is a shout, and the answer is a shout. Nobody replies to anybody.

- **The question.** The asker shouts to the node name on its system group, for example: who in boiler.u1 is above 500 degrees C in the last hour, answer on ans.u1, request 4711.
- **The answer.** Only the nodes that match shout an answer, on the answer group. Everyone listening there hears it once; the HMI and the AI asking the same thing get one answer. A node answers an identical request only once and limits its answer rate.
- **Copies.** The same rule that placed the copies decides which copy answers. The other stays silent and answers only if it does not hear the first one within a short time.
- **Kinds of question.** Value now; history range; events; meaning text; a calculation at the source (time-weighted average, time above a limit, number of changes).
- **Read only.** A question can only read. It never changes a node and gives no authority over the process.
- **Diodes.** Questions need a way back, so they work only on one side of a diode. Behind a diode the reader asks its own local copies, which the forwarder keeps filled. A question never crosses a diode.

Drawing H. Questions, answers and names, all as shouts



Drawing H. Top: a question is shouted on the system group, only matching nodes answer on the answer group, copy 2 waits. Bottom: anyone shouts new tag, the registrar signs the change and shouts it on the names group; every host DNS applies it; the whole zone is repeated; HQ gets the same through the diode.

11. The node thinks before it is asked

Every node pre-analyses its own data where the data is, with the layer scripts of the DEWA data conversion work (R, but any language works). The answer for the AI is ready before anyone asks.

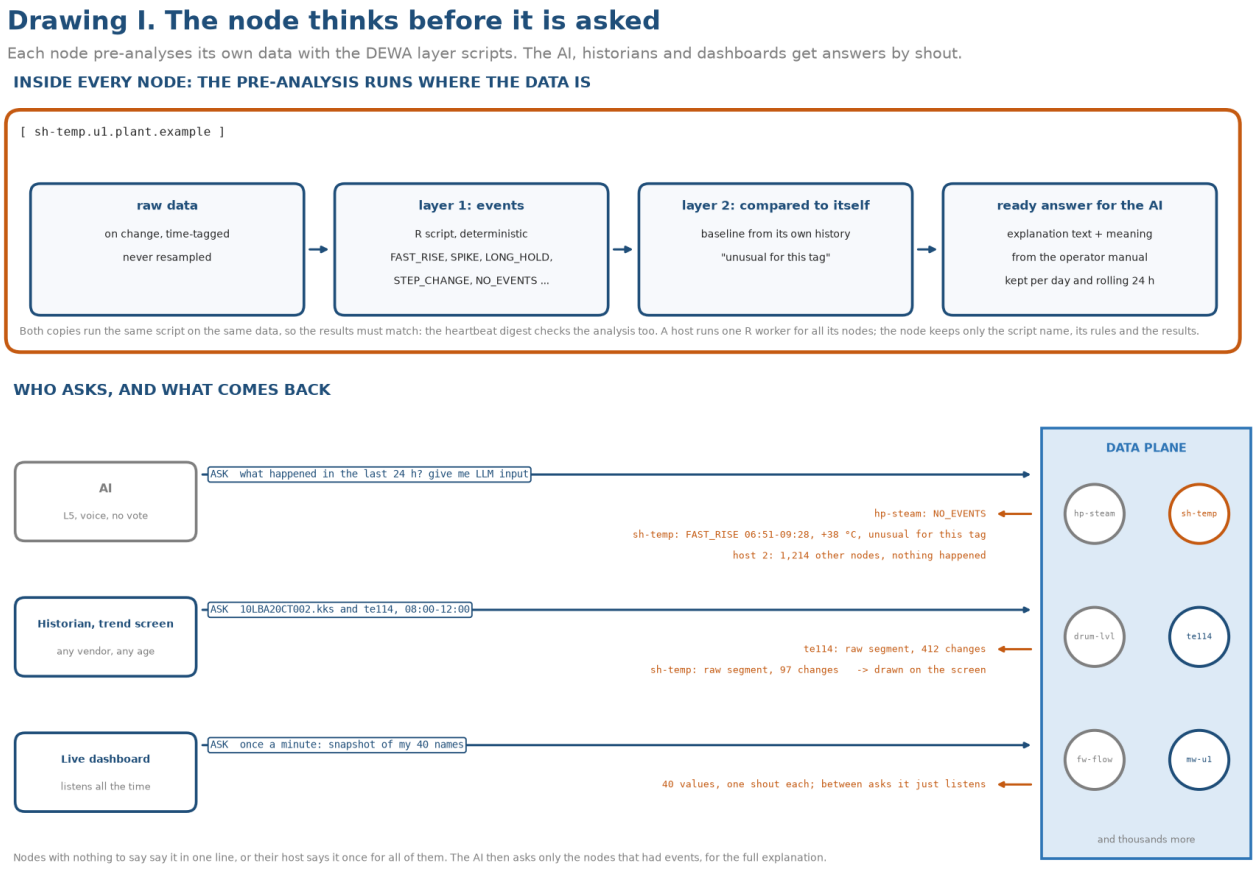
- **Layer 1, events.** L1-Daily Event Generator V2.R reads raw time-tagged data with irregular timestamps and produces deterministic tokens: FAST_RISE, FAST_DROP, SPIKE, STEP_CHANGE, REGIME_SHIFT, LONG_HOLD, BURST_RATE,

KURTOSIS_SHIFT, SKEWNESS_SHIFT, RANGE_BREACH, NO_EVENTS (https://drive.google.com/file/d/1BZliGdPo0lvfey1UrfXyU7OQ_o1uGNrS).

- **Layer 2, compared to itself.** The node holds its own history, so it is the natural place for the baseline: unusual for this tag, compared with itself.
- **The ready answer.** The script already writes an explanation text for an LLM: GLOBAL_ID, DAY, SUMMARY, DETECTED_TOKENS, EXPLANATION, EVENT_LINES, INTERPRETATION_HINT. The node adds the meaning text from the operator manual and keeps it per day and for the rolling last 24 hours.
- **Raw only.** Events come from raw data, never from resampled data (Beyond Purdue #058).
- **Copies check the analysis.** Both copies run the same script on the same data, so the results must match; the heartbeat digest covers the results too.
- **Small nodes stay small.** An R runtime is large, so a host runs one R worker for all its nodes in turn. The node keeps only the script name and version, its rules and its results.
- **Layer 3 and 4 stay with the AI.** Plant-wide reasoning across tags (correlation, narratives) is the AI, at L5: voice, no vote.

Who asks, and what comes back:

- **AI.** Shouts: what happened in the last 24 hours, give me LLM input. Nodes with events shout their explanation; nodes with nothing say NO_EVENTS in one line, or their host says it once for all of them. The AI then asks only the nodes that had events for the full text.
- **Historian or trend screen, any vendor.** Shouts: these tags, by any of their names, from 08:00 to 12:00. Raw segments come back as shouts and are drawn on the screen. An old historian can also simply listen and keep its own copy.
- **Live dashboard.** Listens to the shouts all the time. Once a minute (or on opening a screen) it shouts for a snapshot of its names, to fill whatever it missed.



Drawing I. Top: inside every node the raw data goes through layer 1 events and layer 2 baseline to a ready answer for the AI. Bottom: the AI, a historian and a live dashboard ask by shout and get answers by shout.

12. Time

- **The source clock matters most.** The shouter (protection, controller adapter, meter) stamps each value. The node stores the time and never re-stamps it.
- **Time quality in every shout,** as in IEC 61850: clock failure, clock not synchronized, accuracy, leap seconds known (https://infosys.beckhoff.com/content/1033/tf6510_tc3_iec61850/5648686859.html). Grid operators require all of

them on every timestamp (RTE, https://assets.rte-france.com/prod/public/2022-02/%5BRte-Com%5D%20RSPACE%20-%20Exigences%20d'impl%C3%A9mentation%20IEC%2061850%20-%20Ind.4.1CC.pdf).

- **Two times stored.** Source time and arrival time. Neither is ever rewritten.
- **Own time source per side.** PTP needs two ways and cannot cross a diode, so each side has its own GPS, preferably two masters with holdover.
- **Desynchronization is detected, not hidden.** A node watches the gap between source time and arrival time for each writer and shouts clock suspect when it drifts beyond a limit. Two copies on different hosts cross-check: if their arrival times disagree, the host clock is at fault. Sequence numbers keep the order even when a clock jumps. Values from a bad clock keep their flag for ever.

13. Deployment without people

The steps are in Drawing G: an adapter hears an unknown tag, an I/O list row arrives, or equipment shouts its nameplate; the registrar builds the name from the rules (KKS), adds the aliases and signs a DNS update (RFC 2136, https://www.rfc-editor.org/rfc/rfc2136); every keeper computes the same two hosts from the name and the host list (rendezvous hashing, so they agree without talking); host agents start the node from a template; the node shouts born. Everything on this path is a shout: new tag on the registry group (from an adapter, an I/O list import, an equipment nameplate, or a human at the HMI, each signed), the signed change on the names group, born from the node. Moved and new alias work the same way. Meaning text is filled from the manual library and marked unconfirmed until a human confirms it. A lost host is noticed by missing heartbeats and the copy is started elsewhere. A retired name is kept forever and points to the archive.

14. Writers and readers

As in Drawing D: equipment, machines, humans, AI and documentation write, each through its own one-way gate, each signed with its own key. Names identify; keys authorize writing into the data plane only. Nothing in the data plane authorizes an action on the process.

15. The data diode, the enabler

The data diode is the enabler. Light travels one way, so data and commands cannot mix. The levels hold because the diode holds. A passive diode has no power, no firmware and nothing to configure, so it is as predictable as a relay or a padlock (NIST glossary, https://csrc.nist.gov/glossary/term/data_diode; NIST SP 800-82r3, https://csrc.nist.gov/pubs/sp/800/82/r3/final; IEC 62443-3-3 restricted data flow, https://webstore.iec.ch/en/publication/7033).

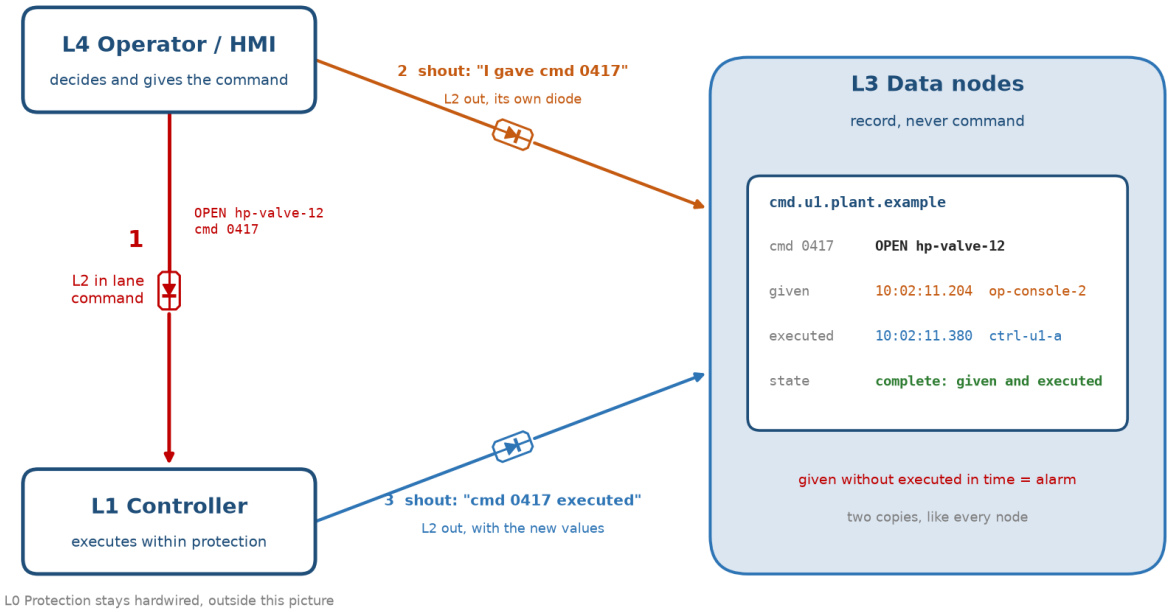
- **The diode as a named node.** Every diode is an asset and an L3 node, for example dd01.u1.plant.example, with aliases by level, asset number and cabinet. Its node holds type (passive or active, passive by preference), direction and channel (out carries data and records, in carries commands), what it carries, and its plant state. Replacing a diode is maintenance, released for work like any other equipment. Only the sending side writes the record; an out diode and an in diode are never linked in one record.
- **A longer wire.** For a shout the diode is only a longer wire. A forwarder on the receiving side hears the group and shouts it again; the zone file crosses the same way. Headquarters keeps its own copies of the nodes and its own copy of the names. Nothing on the far side ever asks the plant.

16. Commands are recorded

A command takes one path only: the L2 in lane from L4 to L1. It never passes through L3. But L3 must know about every command, so each command produces two shouts into L3, each through its own one-way gate (Drawing J).

Drawing J - Commands are recorded

One command, two shouts, one record. Commands still never pass through L3.



L1 does not repeat the command. It only shouts what it executed. Nothing goes back from L3.

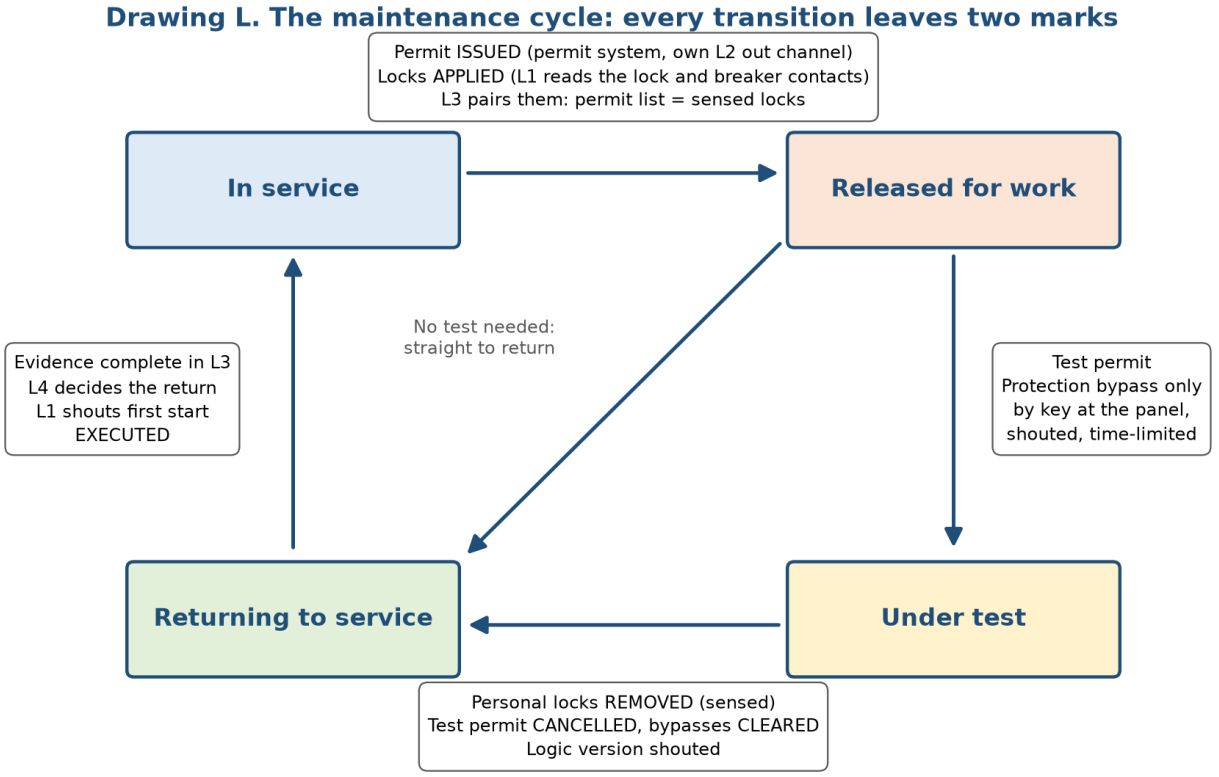
Drawing J. The command goes from L4 to L1 on the L2 in lane. L4 shouts given through its own diode, L1 shouts executed on its data-out gate. L3 pairs them into one record and raises an alarm if executed never comes.

- **Given.** When the operator sends the command, L4 shouts "I gave command 0417" through its own diode into L3: command number, target, requested action, console, source time and signature.
- **Executed.** L1 does not repeat the command. Once it has acted, it shouts only "command 0417 executed" on its normal data-out gate, with the new values and its own source time.
- **One record.** The command node (cmd.u1.plant.example, two copies like every node) pairs the two shouts by command number. A command given but not reported as executed within its time is an alarm. L3 sends nothing back and gives no command.
- **Refused.** L1 is also allowed to refuse a command and shout refused. How L1 checks who is allowed to command (public keys, authentication nodes) is kept for later (section 21).

17. Maintenance and isolation

People work inside the plant, so maintenance is part of operation and is designed in from the start. Isolation is there because of maintenance, and the lock sits in L0: it holds a machine still while someone is inside it. The data plane locks nothing. It hears and records (Drawing L).

- **State as a meaning.** Each equipment node carries its plant state in its meaning: in service, released for work, under test or returning to service. The HMI shows it; the AI reads it.
- **Permit nodes.** A permit is its own node (for example ptw.0193.u1.plant.example), written by the permit system through its own one-way gate and linked by name to the equipment it covers.
- **Lock contacts are shouted.** L1 reads the breaker position and the lock or key contacts and shouts them like any other value, for example 11MBN12AP001 released for work.
- **Reconciliation.** L3 pairs the permit, the sensed locks and any bypass. Humans declare, hardware confirms, L3 reconciles. A mismatch is an alarm, never a correction. L3 sends nothing back.
- **Emergency stop.** It stops a machine; only the lock keeps it stopped (ISO 13850). The data plane records both.



Drawing L. The four plant states. Every transition leaves two marks in L3: what a person declared and what the hardware confirmed. A mismatch is an alarm, never a correction.

18. What stays out

Commands as commands. They stay on the L2 in lane, separate, as in the levels of authority; the data plane only records them (section 16). Protection stays hardwired. The data plane holds and serves; it has no authority.

19. Economics

- **No licence per tag.** Open standards (DNS, IPv6, UDP multicast), commodity hosts, including old servers and cabinet PCs that are already there.
- **Shouts only on change.** Values are shouted and stored only when they change beyond their delta, as time-tagged data, the principle used in the DCS archives. Silence is not loss: the heartbeat and the time allowed to live tell a valid steady value from a dead source.
- **Storage.** One record is about 16 bytes (time, value, quality). In the DEWA example in Beyond Purdue #058, one tag had 97 raw changes in a day: about 1.5 kB per day, 0.6 MB per year. If 10,000 signals average 1,000 changes a day, that is about 160 MB per day, about 58 GB per year per copy (assumption, to be replaced by real plant counts).
- **Network.** The same assumption gives about 120 change shouts per second for the whole plant. Heartbeats then dominate, so one heartbeat per host carries the digests of all its nodes.
- **Averages.** With on-change data a plain sample mean is biased; nodes answer averages with the time-weighted average (internal note Time Average Value).

20. Changes from the 2024 design

2024	Now	Why
Central DNS on CockroachDB	Standard DNS, a local copy on every host	Already distributed; works offline; crosses a diode as a file
HTTP and gRPC queries	One-way shouts	Works through a diode; no server to overload
Kubernetes and InfluxDB per node	Tiny nodes, many per host, append files	Small, cheap, runs on old hardware
TLS and OAuth2	Signed messages	A signature survives a diode; a session does not
Blockchain logging	Hash-chained history compared between copies	Same proof, no consensus

2024	Now	Why
Alarm subnets	Alarm events	An address is identity, not state
Queries and replies	Ask by shout, answer by shout	No session; works on both sides of a diode
Zone transfer by pull	Zone changes and full zone by shout	One mechanism through diodes

21. Open points

- **Switch group limits** for MLD snooping on the real plant switches.
- **Time.** PTP or GPS at the shouters; quality marked when time is lost.
- **Writer keys:** how they are issued, rotated and shouted.
- **Naming rules:** who owns them, KKS mapping, the governance of aliases.
- **Storage sizing** from real change counts per signal class, and history retention.
- **Analysis in the node:** size of the R worker per host, script versions in every result, layer 2 baseline rules per signal class.
- **Command authorization:** refused commands, and how L1 checks who is allowed to command: public keys and authentication nodes.
- **Permit system interface:** which permit or maintenance system exports permits, and in which format.
- **Question rules:** answer rate limits, which copy answers, size of answers.
- **Bench prototype:** one adapter, 100 signals, three hosts, one diode. Kill a host and check that the copy is started elsewhere and that the headquarters copy keeps working.

22. Changes in V1.0

Reordered to follow the works: the laws first, then data in independent nodes, then names and aliases, then copies and everything else. New: the laws section; why at least two copies; the three ways a name is updated; the data diode as the enabler and as a named node.

23. References

- EP3734376 B1, System, computer system and method for controlling distributed actuators: https://patents.google.com/patent/EP3734376A1/en
- IEC 61850 GOOSE networks (SEL): https://cdn.selinc.com/assets/Literature/Publications/Technical%20Papers/6921_IEC61850Network_MS_20190712_Web.pdf
- GOOSE retransmission and timing: https://pmc.ncbi.nlm.nih.gov/articles/PMC7926645/
- DEWA L1-Daily Event Generator V2.R (internal): https://drive.google.com/file/d/1BZliGdPo0lvfey1UrfXyU7OQ_o1uGNrS
- DEWA Data transformation, layered approach (internal): https://drive.google.com/file/d/1OFXDsorPG5gUVN05wVHzTuT3BTLhzL8L
- IEC 61850 time quality format (Beckhoff): https://infosys.beckhoff.com/content/1033/tf6510_tc3_iec61850/5648686859.html
- RTE IEC 61850 implementation requirements (time quality): https://assets.rte-france.com/prod/public/2022-02/%5BRte-Com%5D%20RSPACE%20-%20Exigences%20d'impl%C3%A9mentation%20IEC%2061850%20-%20Ind.4.1CC.pdf
- ISO 14118:2017, prevention of unexpected start-up: https://www.iso.org/standard/66460.html
- ISO 13850:2015, emergency stop function: https://www.iso.org/standard/59970.html
- IEC 61511 ed.2 clause 16.2, bypass log (ABB summary): https://library.e.abb.com/public/52dc61cd2284462c8f66bdcafb1fefcb/11%20C%20Bratt%20ABB%20changes%20in%20IEC61511.pdf
- EN 50110-1:2023, operation of electrical installations: https://standards.iteh.ai/catalog/standards/clc/1b0a0556-70ff-4d0f-a322-ca2d7d05af90/en-50110-1-2023
- RFC 1034 DNS concepts: https://www.rfc-editor.org/rfc/rfc1034
- RFC 1996 DNS NOTIFY: https://www.rfc-editor.org/rfc/rfc1996
- RFC 2136 DNS dynamic update: https://www.rfc-editor.org/rfc/rfc2136
- RFC 2606 Reserved example names: https://www.rfc-editor.org/rfc/rfc2606
- RFC 3306 Unicast-prefix multicast: https://www.rfc-editor.org/rfc/rfc3306
- RFC 3596 AAAA records: https://www.rfc-editor.org/rfc/rfc3596
- RFC 3810 MLDv2: https://www.rfc-editor.org/rfc/rfc3810
- RFC 4193 Unique local IPv6: https://www.rfc-editor.org/rfc/rfc4193
- RFC 4291 IPv6 addressing: https://www.rfc-editor.org/rfc/rfc4291
- RFC 4541 MLD snooping: https://www.rfc-editor.org/rfc/rfc4541
- RFC 6330 RaptorQ: https://www.rfc-editor.org/rfc/rfc6330
- RFC 6726 FLUTE: https://www.rfc-editor.org/rfc/rfc6726
- RFC 6762 Multicast DNS: https://www.rfc-editor.org/rfc/rfc6762
- RFC 6763 DNS service discovery: https://www.rfc-editor.org/rfc/rfc6763
- RFC 7553 URI record: https://www.rfc-editor.org/info/rfc7553

- RFC 8032 Ed25519: https://www.rfc-editor.org/rfc/rfc8032
- RFC 8976 ZONEMD: https://www.rfc-editor.org/rfc/rfc8976.html
- RFC 9364 DNSSEC: https://www.rfc-editor.org/info/rfc9364
- NIST glossary, data diode: https://csrc.nist.gov/glossary/term/data_diode
- NIST SP 800-82r3, Guide to OT Security: https://csrc.nist.gov/pubs/sp/800/82/r3/final
- IEC 62443-3-3, system security requirements: https://webstore.iec.ch/en/publication/7033