WireTrace

WHITE PAPER

Deep Protocol & Asset Intelligence

Why the industry's twenty-year investment in reading packets has stopped producing better answers, and what has to change in the layer above it.

Category: DPAI - Deep Protocol & Asset Intelligence
Applies to: OT, IT, IoT and IoMT networks
Audience: CISOs, network and OT architects, security operations leads
Published: August 2026 · **Version:** 1.0

EXECUTIVE SUMMARY

Deep Packet Inspection was one of the most successful ideas in network security. It is also finished as a source of competitive advantage. Every serious vendor can read a packet. What separates them now is what they build on top of it, and the industry has no shared name for that layer. This paper proposes one, defines it precisely, and argues that it is where the next decade of network defence is decided.

Deep Protocol & Asset Intelligence (DPAI) is the discipline of turning protocol evidence into durable, reconciled knowledge about the things on a network: what each asset is, what it does, where it physically sits, how it behaves, and what its exposure means in context. DPI answers *what is in this packet.* DPAI answers *what is on my network, and what should I do about it.* Those are different questions, and the second one is not solved by inspecting packets harder.

The argument in four steps:

1. **The bottleneck moved.** Encryption, ephemeral addressing and device sprawl mean that payload visibility is falling while the number of things that need identifying is rising. Vendors respond by adding parsers. That is optimising the part of the pipeline that is no longer the constraint.
2. **The unit of defence is the asset, not the packet.** Every consequential security decision - segment it, patch it, isolate it, accept the risk - is made about a device. Packets are evidence. Assets are the subject.
3. **Asset knowledge is a reconciliation problem, not a detection problem.** One device presents differently to passive capture, SNMP, an active probe and a human operator. Producing one trustworthy record requires an explicit model of identity, provenance and precedence. Most platforms have this implicitly, which is why their asset counts drift.
4. **Physical grounding is the missing dimension.** Logical topology tells you what talks to what. It does not tell you which cabinet to walk to. A reconciled physical model closes the gap between an alert and an action.

The claim this paper makes. DPI is rung three of a seven-rung climb, and the industry has been polishing rung three since 2004. DPAI is the name for rungs five through seven taken as one engineering problem rather than three separate features. Vendors who treat it as a bolt-on will keep shipping high-fidelity packet decoding attached to an asset inventory nobody trusts.

CONTENTS

1 · THE END OF DPI AS A DIFFERENTIATOR

Deep Packet Inspection earned its place. Before it, a firewall decided using an IP address, a port and a protocol number. DPI let a device read the payload, identify the application regardless of port, and match signatures against content. It made application-aware policy possible and it made a generation of security products viable.

Its problem is not that it stopped working. Its problem is that it finished. Reading a packet is now a solved engineering task with mature open implementations, and the marginal value of the next parser is approaching zero for three independent reasons.

Encryption removed the payload

The overwhelming majority of traffic on a modern enterprise network is encrypted, and TLS 1.3 with encrypted client hello removes even the metadata that made passive application identification easy. A platform whose intelligence lives in payload inspection loses capability every year by default. A platform whose intelligence lives in behaviour, timing, peer sets and asset context does not.

The interesting protocols were never the hard part

In OT and IoMT the protocols that matter are mostly unencrypted and mostly well documented. Modbus, DNP3, S7comm, EtherNet/IP, BACnet, DICOM and HL7 are readable. Decoding them is a matter of engineering effort, not insight. Every credible vendor decodes them. Parser count as a marketing metric measures how long a team has been working, not how much a customer will learn.

The volume problem inverted

Twenty years ago there were fewer devices generating less traffic, and the challenge was seeing inside it. Today a mid-size hospital or plant has tens of thousands of endpoints, many of them unmanaged, unpatchable and invisible to agents. The challenge is no longer seeing inside the traffic. It is knowing what produced it and whether that thing is supposed to exist.

> *"We can decode every field in the exchange. We still cannot tell you with confidence how many infusion pumps you own." That sentence describes most deployments in the market, and it is a rung-five failure that no amount of rung-three excellence repairs.*

The symptom customers actually report

Ask an operator running a mature packet-inspection platform what frustrates them, and the answers cluster in a revealing way. None of them are decoding complaints.

- The asset count changes when nothing changed on the network.
- One physical device appears three times under three identities.
- An alert fires on a broadcast address, a link-local address, or the organisation's own DNS resolver, and there is no way to make it stop being wrong rather than merely silenced.
- A device is flagged offline when it is running fine and simply stopped answering a poll.
- An alert names an IP address, and locating the physical device behind it takes a day.
- Something the operator personally corrected last month is wrong again.

Every one of these is an asset-intelligence defect. Every one of them survives a perfect packet decoder. That is the case for naming the layer where they live and engineering it deliberately.

2 · THE MATURITY LADDER

Network visibility has a well-understood progression. Each rung subsumes the one below it, which is why DPI and DPAI are not rival products. They are two positions on one climb, and the argument of this paper is simply that the industry has been standing on rung three for a long time.

1	**Packet filtering** IP, port and protocol allow or deny. Stateless.	
2	**Stateful inspection** Tracks connection state across packets. Knows whether a session is open, not what it means.	
3	**Deep Packet Inspection** Reads payloads, headers and signatures. Commodity capability since roughly 2010.	INDUSTRY PLATEAU
4	**Application awareness** Identifies the application regardless of port. Degrades as encryption spreads.	
5	**Deep Protocol Intelligence** Reconstructs sessions, semantics and intent. Understands that a write to a specific register at a specific time is a control action with consequences.	DPAI
6	**Behavioural intelligence** Per-device baselines, peer-set modelling and deviation detection. Survives encryption because it does not depend on payload.	DPAI
7	**Asset, risk and threat intelligence** Continuous, correlated posture per asset: identity, exposure, criticality, physical location, and a defensible score an operator can argue with.	DPAI

The important boundary is between rung two and rung five, and it is easy to miss because both involve the word "state". Classic stateful inspection tracks *connection* state: is this TCP session open. It does not reconstruct the conversation's *meaning*. Semantic reconstruction is what separates reading a letter from understanding a negotiation, and it is an architecturally different problem: it requires memory that outlives the flow, and identity that outlives the address.

Why this matters commercially. A vendor at rung three plus a spreadsheet export is selling a sensor. A vendor at rungs five through seven is selling an operating picture. Buyers cannot tell them apart from a feature list, because both list the same protocols. Section 10 gives twelve questions that separate them in under an hour.

THE ARCHITECTURAL TELL

There is a reliable structural signature. Rung-three work is stateless, per-packet, and belongs in compiled code on a hot path measured in nanoseconds. Rung five and above is stateful, cross-flow, cross-source and cross-time, and belongs in a correlation layer measured in seconds to minutes. These are such different workloads that they are almost always different codebases.

So the question to ask a vendor is not "do you do deep inspection". It is *where does the asset record live, and what reconciles it*. If the answer is that the sensor emits events and a database accumulates them, the platform is at rung three with storage. If there is a named layer whose job is to decide what a thing *is*, and it can explain its own decisions, the platform is doing DPAI whether or not it calls it that.

3 · DEFINING DPAI

Deep Protocol & Asset Intelligence is the continuous derivation of trustworthy asset knowledge from protocol evidence. It treats the asset as the unit of analysis, protocol behaviour as the primary evidence, physical placement as a first-class attribute, and the provenance of every assertion as part of the record rather than an implementation detail.

Read the name literally, because each part is load-bearing. **Deep** means semantic, not merely decoded. **Protocol** means the evidence base is how devices actually converse, which is available whether or not payloads are readable. **Asset** names the subject: the output is a device record, not an event stream. **Intelligence** means derived and defensible - a conclusion with a stated basis, not a field copied out of a frame.

Note what the definition excludes. DPAI is not a synonym for network detection and response, and this paper is not proposing a rebranded SIEM. It says nothing about ticketing, case management, playbook automation or endpoint control. Those are consumers of asset intelligence. Conflating them is how the category would become meaningless.

THE FOUR PILLARS

1 · Protocol semantics

Understand what an exchange *means* in its own domain. A function code is not a number; it is a read, a write, a stop, or a firmware transfer, with a different consequence in each case.

2 · Asset identity and reconciliation

An explicit model for merging what passive capture, SNMP, active probing and human operators each claim about the same device, with declared precedence and preserved overrides.

3 · Physical grounding

Every asset carries a place: site, building, floor, room, rack, port. Discovered cabling reconciled against recorded intent, with drift surfaced rather than hidden.

4 · Evidence discipline

Every assertion carries its source, its confidence and its age. Evidence that cannot support a conclusion is rejected at the boundary, with a reason, not silently accumulated.

A FALSIFIABLE TEST

Categories that cannot be tested become marketing. Here is a test any platform can be held to. Pick one device on a live network. Ask the platform six questions and require it to answer without a human interpreting for it.

#	QUESTION	WHAT A DPAI ANSWER LOOKS LIKE
1	What is this device?	A type and a role, separated from its capabilities. Bridging traffic does not make something a switch.
2	How do you know?	A named source per attribute, with rank. Passive observation, SNMP, active probe and operator override are not interchangeable and must not be averaged.
3	Where is it?	A physical location to rack and port, with a flag if the discovered wiring disagrees with the record.
4	What is normal for it?	A per-device baseline: peers, protocols, volumes, hours. Not a network-wide average.
5	Why is its risk score that number?	A decomposition an engineer can dispute line by line. A score without a derivation is a number, not intelligence.
6	What changed, and when?	An attribute-level history. If it cannot say when the type changed and what changed it, it cannot be audited.

A platform at rung three answers question one, partially, and struggles from question two onward. The gap is not cosmetic. It is the difference between a tool that reports and a tool that can be relied on.

4 · PILLAR 1 - PROTOCOL SEMANTICS OVER PACKET DECODING

Decoding produces fields. Semantics produce consequence. The distinction is easiest to see in industrial control, where the same bytes mean radically different things depending on direction, target and timing.

Consider a single Modbus exchange. A decoder reports: TCP 502, function code 6, register 40001, value 1. Every vendor produces that. A semantic layer reports something else entirely: an engineering workstation issued a *write* to a safety-relevant register on a PLC that has accepted writes from exactly one host for ninety days, and this is not that host, and it happened outside the maintenance window. The first output is a row in a log. The second is a decision.

Getting from one to the other requires three things a per-packet decoder structurally cannot hold:

- **Memory that outlives the flow.** "One host has written to this register for ninety days" is not in the packet.
- **Identity that outlives the address.** The workstation may have had three addresses this month. The claim is about the device, not the address.
- **Domain modelling.** Something must encode that this register is safety-relevant and that function code 6 is a write. That is protocol knowledge, not parsing.

The same shape recurs across domains. In IoMT, a DICOM association is not interesting as a decode; it is interesting because a modality initiated an outbound transfer to a destination it has never used. In IT, a Kerberos exchange matters when a service account authenticates from a subnet where that account has never appeared. In IoT, a camera's firmware fetch matters because it reaches a host outside the vendor's published infrastructure. None of these conclusions are in a packet. All of them are in the protocol behaviour of an identified asset over time.

> **The encryption dividend.** This is why DPAI improves as encryption spreads while DPI degrades. Who talks to whom, how often, in what volumes, in what order, at what hours, with what handshake characteristics - all of it survives encryption. A platform whose value depends on reading the payload is on a shrinking base. A platform whose value depends on identified assets and their behaviour is not.

5 · PILLAR 2 - ASSET IDENTITY AS A RECONCILIATION MODEL

This is the pillar most platforms get wrong, and the failure is not subtle: customers see it as an asset count that will not hold still.

The root cause is that a single physical device presents multiple identities. It has a MAC address, which is stable but invisible across a routed boundary. It has IP addresses, which are visible everywhere and change. It may have several interfaces, a virtual address, a hostname reported by one protocol and a different one by another, and a serial number only an authenticated probe can read. Sources disagree, and they disagree in patterned ways.

Reconciliation therefore needs decisions made explicitly and written down:

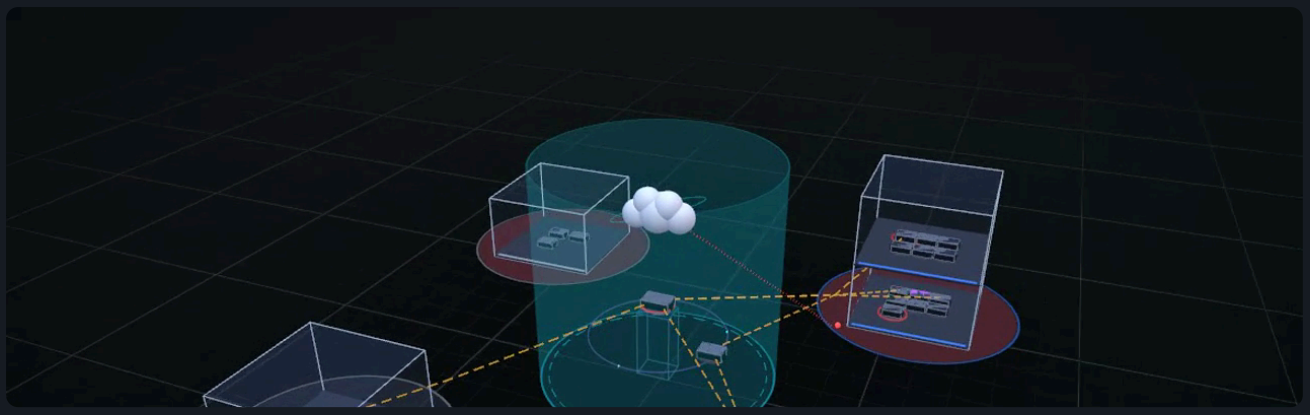
- **A primary identity key,** with a defined rule for what happens when it is unavailable. Order of observation must not change the outcome.
- **Provenance ranks.** A four-level ordering works well in practice: passive observation, then SNMP, then active probing, then operator override. Higher rank wins, and an operator override is never silently displaced by a machine.
- **Type separated from capability.** A device that bridges traffic has a bridging capability. It is not therefore a switch. Servers bridge. Hypervisors bridge. Access points bridge. Collapsing capability into type is the single most common source of wrong device types in this market.
- **State that distinguishes ignorance from absence.** "I did not get a reply" is not "offline". Reachability needs at least five states - online, degraded, unreachable, unknown and disabled - and an authentication failure must never be reported as a device being down, because it is a statement about credentials, not about the device.
- **Consolidation that cannot fork.** Re-enrolling a sensor, changing a hostname or renumbering a subnet must not create a second copy of a device that already exists.

None of this is glamorous, and none of it demonstrates well. It is also the difference between an inventory a security team builds policy on and a list they export to a spreadsheet and correct by hand.

6 · PILLAR 3 - PHYSICAL GROUNDING AND THE DIGITAL TWIN

Logical topology is a solved and widely shipped feature. Every platform in this market can draw a graph of what talks to what. That graph answers questions about traffic. It answers almost nothing about action.

The gap appears the moment an alert becomes a task. An alert names an asset. Responding to it requires knowing which building, which floor, which room, which cabinet, which switch, and which port - and in OT and IoMT it also requires knowing what is physically adjacent, because that is what determines whether isolating a device is safe. A logical graph cannot answer any of that, so the answer comes from a spreadsheet, a cable label, or an engineer's memory. All three are stale.



Physical grounding, rendered. A live environment as a navigable three-dimensional model: buildings placed on zone pads, rack-mounted switches inside them, a network boundary carrying the internet edge and VPN entry, and discovered trunk links drawn between real devices. Every object is an asset record, not an icon on a diagram.

Why "digital twin" is the right term, and where it is abused

A twin is not a three-dimensional picture. Rendering is the least interesting part, and treating visualisation as the feature is how the term got devalued. A network digital twin earns the name by holding three things at once:

1. **Recorded intent** - the placement and cabling an organisation believes it has.
2. **Observed reality** - what discovery actually finds through LLDP, CDP, SNMP bridge and interface tables, and passive observation.
3. **The delta between them** - surfaced as drift, with the provenance of each side visible, so a disagreement is a finding rather than a rendering error.

The third item is the whole point. A model that only shows recorded intent is a drawing and will be wrong within a quarter. A model that only shows observed reality cannot tell you that something moved, because it has no memory of where it was supposed to be. A twin is useful precisely because it can say *this device is not where your records place it*, and name the port it is actually on.

What physical grounding makes newly possible

- **Blast radius as a physical question.** Not just which hosts a compromised device can reach logically, but which devices share its switch, its rack and its power domain.
- **Path tracing that reflects wiring.** The actual layer-two and layer-three path between two assets, hop by hop, which is what tells you whether a segmentation control is genuinely in the path or merely configured.
- **Segmentation you can verify by looking.** Zone boundaries drawn over physical placement make a violation visible instead of inferable from a table.
- **Response that starts at the right cabinet.** The mean time from alert to hands-on falls when the platform names the rack.

> **An honest boundary.** Physical placement cannot be fully derived from the network. Discovery can establish that a device is on a specific port of a specific switch, and switch-level grouping follows from that. Which room the switch is in is a fact about the building, and it enters the model through operator assignment. A vendor claiming fully automatic physical placement is either inferring from naming conventions or overstating. The correct design accepts a small amount of human input once, ranks it above machine inference, and then keeps everything downstream of it current automatically.

7 · PILLAR 4 - EVIDENCE DISCIPLINE

The fourth pillar is the one no vendor markets, because it consists of refusing to do things. It is also the one that decides whether anybody trusts the other three.

Intelligence platforms consume evidence from many sources: passive observation, polling, active probes, external threat feeds, operator input. Each source can be wrong in characteristic ways, and an intelligence layer that accepts everything produces confident nonsense. Discipline means rejecting bad evidence at the boundary, with a recorded reason, before it reaches a conclusion.

A worked example: indicator matching

Threat indicator feeds routinely contain entries that cannot support a detection. Common cases:

- Link-local and loopback addresses, which every network contains by design.
- Multicast and broadcast destinations, which match constantly and mean nothing.
- Public DNS resolvers and major CDN endpoints, which most of the organisation legitimately contacts.
- Ranges so broad they cannot discriminate. A very wide network prefix in an indicator feed is not an indicator; it is a guarantee of noise.
- Addresses reserved by standards bodies for documentation and testing.

Left unfiltered, these produce alert volumes that look like activity and function as denial of service against the analyst. The instinctive fix - let the operator suppress each one - is the wrong shape, because it makes every customer rediscover the same facts and leaves the underlying data poisoned.

The disciplined design differs in four ways:

1. **Validation is centralised.** One place decides whether an indicator is matchable, and every consumer, in every language, uses that same decision. Two matchers with two opinions is a defect waiting for a customer to find.
2. **Verdicts are explicit and enumerated.** Accept, suppress or reject, each with a reason from a fixed vocabulary rather than free text, so suppression can be audited and reported on.
3. **Authoritative registries replace hard-coded lists.** Special-purpose address assignments are published by standards bodies. Encoding the registry is maintainable; accumulating a list of exceptions from support tickets is not.
4. **Overrides exist but are narrow and explicit.** A customer with a genuine reason to match inside a normally excluded range must be able to say so, per feed, deliberately, and that exception must be visible in the record.

> **The measurable effect.** In one controlled comparison on a live network, an unvalidated matcher generated over 1,100 detections in a single working period, at more than 90 per minute, of which exactly one described real activity. The rest were link-local addresses, multicast destinations, a broadcast address and a public DNS resolver. With centralised validation in place, the same network over the same period produced the one real detection and nothing else. The detection logic did not change. The evidence discipline did.

The same principle, applied everywhere

Evidence discipline is not an indicator-feed feature. It is a posture, and it shows up wherever a platform draws a conclusion:

- **Vendor identification** is only as good as the hardware-address registry behind it, and locally administered addresses must be recognised as carrying no vendor information at all rather than being mapped to a nearby entry.
- **Support claims must be graded.** There is a real difference between "conforms to the standard", "validated against a captured fixture", "tested on the physical hardware" and "verified in production". Collapsing those into "supported" is how a customer discovers a limitation during an incident.
- **Absence of evidence must be representable.** Unknown is a legitimate state and must be distinguishable from a negative finding. A platform that cannot say "I do not know" will guess.
- **A skipped test is not a passing test.** This applies to the vendor's own engineering as much as to the product's output.

8 · FOUR DOMAINS, ONE MODEL

DPAI applies across four domains, and the reason it can is that all four share the same underlying problem: consequential devices that cannot host an agent and cannot be interrogated aggressively. What differs is the consequence of getting it wrong.

DOMAIN	WHY AGENTS DO NOT WORK	WHAT DPAI CONTRIBUTES	COST OF A WRONG ANSWER
OT	Controllers run vendor-locked firmware, cannot be patched on demand, and often cannot tolerate unexpected traffic.	Semantic control-action visibility, per-device peer baselines, physical placement for safe isolation decisions.	Production stoppage or a safety event. Isolating the wrong device is itself the incident.
IT	Agents cover managed endpoints. Printers, cameras, building systems and unmanaged hosts are exactly the gap.	Reconciled inventory including unmanaged devices, exposure context, segmentation verification.	An unknown asset becomes the entry point, and nobody owns it.
IoT	Constrained devices, no agent capability, frequently shipped with unchangeable defaults.	Identity from protocol behaviour, firmware and vendor-infrastructure deviation, physical grouping by switch and rack.	Large populations of identical devices compromised together, at a scale no manual process absorbs.
IoMT	Regulatory constraints on modification, long certified lifecycles, clinical availability requirements.	Modality and transfer-pattern awareness, careful reachability semantics, location to room for clinical coordination.	Patient impact, and a device that cannot simply be taken offline to investigate.

The temptation is to build four products. That is a mistake, and the reason is architectural: identity reconciliation, provenance, physical grounding and evidence discipline are the same engineering in all four domains. What is domain-specific is protocol semantics and consequence modelling, which belong in configuration and rules, not in separate platforms. A vendor with four codebases will have four asset models, and a hospital with a plant, or a plant with a clinic, will be shown two different answers about the same network.

9 · WHAT DPAI DOES NOT SOLVE

A category defined only by its strengths is advertising. These are real limits, and any honest DPAI vendor has them.

Encrypted payloads stay encrypted

DPAI reasons about behaviour, peers, timing and volume, and that is genuinely resilient. It is not equivalent to reading content. Where content matters - data loss detection, some malware families - network-derived intelligence narrows the question rather than answering it, and saying otherwise invites a decrypt-everything argument nobody wins.

Passive observation has a horizon

A device that never transmits within the capture scope does not exist to a passive sensor. Closing that gap requires polling or active probing, which means transmitting, which must then be operator-gated, scoped and off by default. There is no configuration that is simultaneously fully passive and fully complete, and pretending otherwise is the most common overclaim in this market.

Physical placement needs a human once

As section 6 states plainly: the network reveals ports and switches, not rooms. The design goal is to need that input once per structural change, not once per device, and to keep everything derived from it current automatically.

Behavioural baselines need time and stability

A per-device baseline is meaningless on day one and unreliable across a genuine change in operations. A platform that alerts confidently from a two-hour baseline is generating noise with a statistical accent. Baseline maturity should be visible to the operator, not implied.

DPAI is not the response layer

It produces the picture decisions are made from. It does not manage cases, orchestrate playbooks or control endpoints, and a platform that claims all of it is describing a roadmap. The honest promise is that asset intelligence is the input every one of those layers is currently missing.

10 · HOW TO EVALUATE A DPAI CLAIM

Feature lists cannot separate rung three from rung seven, because both list the same protocols and both use the word "visibility". These twelve questions can, in about an hour, on a live or lab network. They are written to be asked of any vendor, including this one.

#	ASK	WHAT THE ANSWER REVEALS
1	Show me one device and every attribute's source.	If provenance is not per attribute, reconciliation is implicit and the record cannot be defended.
2	I disagree with this device's type. Change it. Now re-run discovery.	If the correction is overwritten, operators will stop correcting anything, and the inventory decays.
3	Which rack and port is this device on?	Separates physical grounding from a logical graph. Follow up: what if my records disagree?
4	Something answers no polls but passes traffic. What do you call it?	"Offline" is wrong. Look for a reachability model that separates unknown from absent.
5	A poll fails authentication. What do you report?	If a credential problem shows as a device outage, the state model is missing.
6	Load an indicator feed containing a link-local address and a very broad prefix.	Does it alert, silently drop, or reject with a stated reason? Only the third is discipline.
7	Is this device type or a capability? Show me a server that bridges.	Tests the type-versus-capability separation directly, where most platforms fail.
8	Explain this risk score to me line by line.	An undecomposable score cannot be argued with, so it will not be acted on.
9	Which of your supported vendors are hardware-tested, and which are inferred?	A vendor who cannot grade their own coverage has not measured it.
10	Does anything in this deployment transmit onto my network? Show me the setting.	Both "nothing ever" and "of course" are wrong answers. The right one is specific and points at a default-off control.
11	Re-enroll a sensor, then show me the asset count.	Duplicate records under identity churn is the classic reconciliation failure and it is easy to test.
12	What did this asset look like thirty days ago?	Attribute-level history is what turns an inventory into evidence.

CLOSING ARGUMENT

The industry has spent two decades getting very good at reading packets, and it worked. The reason network security still feels unsolved is not that the reading got worse. It is that the reading was never the answer to the question operators actually have, which is: what is on my network, what is it doing, where is it, and what should I do about it.

That question is answered one rung up, by a layer that has been treated as plumbing behind the sensor rather than as the product. Naming it changes what gets engineered. It makes asset reconciliation a design problem instead of a database side effect. It makes provenance a requirement instead of a debugging convenience. It makes physical placement an attribute instead of a spreadsheet. It makes rejecting bad evidence a feature instead of a support workaround.

Deep Protocol & Asset Intelligence is that name. It sits above Deep Packet Inspection and depends on it, which is why this paper does not argue that DPI was wrong. It argues that DPI was rung three, that the industry has been standing there for a long time, and that everything a security team is actually asking for lives at rungs five, six and seven.

The test of the category is simple, and section 3 states it as six questions about one device. Any vendor should be able to answer all six, in front of you, on your network. Ask them. Ask us.