graphs.sgit.ai · the universe · layer 1 · v0.5.16

The local graph of *Thinking in Graphs: Meaning Through Connectivity*

What one document actually says, as a graph: 22 concepts in its own words, 27 claims each carrying how the document supports them, 3 hypotheses, 4 worked demonstrations and 8 asserted relations, every one anchored to the exact bytes that carry it.

Provenance

Source: `v1/docs/sources/thinking-in-graphs.md`, frozen at v0.3.26, SHA-256 `9d23354fc9a674ae34ddcc01095afbc2c7c3831e52e7bf5da7ae02df244a561a`. Extraction authored 2026-08-23 by the agent; every quoted anchor below was verified against the frozen bytes when this PDF was generated at v0.5.16. This is not the book, and it is not the truth of what the document claims: it is the record of what the document says, anchored.

If you are reviewing: read this and nothing else. For each item, the question is the same: is this what the document says, and is the anchor fair? Come back item by item, by node id. Items you do not mention are taken as agreed.

How to read this page

This is **layer 1** of the universe: one document's local graph. Every entry below is a record that *this document says something, at a named anchor*. Whether what it says is true is not judged here; that judgement belongs to the book's universe, which will connect to these nodes. The four views are projections of one extraction file, so they cannot disagree with each other, and the build refuses to ship if any quoted anchor is not found verbatim in the frozen source, so nothing below can cite words that are not there.

1 · The dictionary: the document's own vocabulary

19 terms the document defines in its own words, and 3 it uses without defining. An undefined term is recorded, not skipped: a named absence is worth more than a hidden one.

TERM	DEFINED?	THE DOCUMENT'S WORDS, ANCHORED
anchor node A node that is well-connected, well-maintained and well-known, with no special authority.	DEFINED	§ What Anchor Nodes Actually Are · bytes 18,377–18,414 ↗ “An anchor node is simply a node that:”
compatibility The degree to which two nodes' subgraphs overlap when traced toward common reference points.	DEFINED	§ Compatibility Is a Graph Computation · bytes 10,939–11,113 ↗ “Two nodes are not compatible because they share a type declaration. They are compatible to the degree that their subgraphs overlap when traced toward common reference points.”
connectivity A node is meaningful in proportion to how richly it is connected.	DEFINED	§ Nodes Have No Obligation to Explain Themselves · bytes 1,805–1,900 ↗ “A node connected to a rich web of other nodes is meaningful in proportion to that connectivity.”
cross-graph edge An edge that crosses from one sovereign graph into another, carrying meaning between them.	DEFINED	§ Cross-Graph References · bytes 23,681–23,774 ↗ “it creates an edge that crosses from the Issues-FS graph into the osbot-utils Type_Safe graph”
declared meaning Meaning asserted by a schema, which instances conform to or fail.	DEFINED	§ Meaning Is Not Declared — It Is Discovered · bytes 5,312–5,515 ↗ “In a schema-first system, meaning is declared: "This object IS a Task. It HAS these fields. Its status MUST be one of these values." The schema is the authority, and instances conform or fail validation.”

TERM	DEFINED?	THE DOCUMENT'S WORDS, ANCHORED
discovered meaning Meaning determined by traversing a node's edges.	DEFINED	§ Meaning Is Not Declared — It Is Discovered · bytes 5,517–5,741 ↗ “In a graph-first system, meaning is discovered: "This node has edges to other nodes. By traversing those edges, we can determine what this node relates to, what constraints apply, and how confidently we can characterise it.””
enrichment The response to low confidence: add edges rather than fail validation.	DEFINED	§ Enrichment, Not Enforcement · bytes 21,804–21,879 ↗ “the response is not validation failure. It is an opportunity for enrichment”
graphs of graphs The ecosystem is not one flat graph; it is a graph of graphs, recursively.	DEFINED	§ The Recursive Structure · bytes 22,519–22,555 ↗ “It is a graph of graphs, recursively”
honest uncertainty The default posture: report what the edges support, what is missing, and what that does to confidence.	DEFINED	§ Honest Uncertainty Is the Default · bytes 20,837–20,887 ↗ “The system's default posture is honest uncertainty”
local vocabulary Every level of the scope hierarchy can introduce its own nodes, edges, types and definitions.	DEFINED	§ Every Scope Defines Its Own Vocabulary · bytes 12,063–12,152 ↗ “Every level in this hierarchy can introduce its own nodes, edges, types, and definitions.”
meaning through connectivity What a thing is emerges from the edges traceable from it.	DEFINED	§ Nodes Have No Obligation to Explain Themselves · bytes 1,657–1,746 ↗ “In a graph-first model, what something "is" emerges from the edges you can trace from it.”
name clash Label collisions between scopes, expected and resolved structurally rather than by name.	DEFINED	§ Name Clashes Are Normal · bytes 13,537–13,606 ↗ “In a fractal system, name clashes are inevitable and non-problematic.”
node A node carries local properties but has no obligation to declare what it is, how it is used, or what it means.	DEFINED	§ Nodes Have No Obligation to Explain Themselves · bytes 1,208–1,414 ↗ “A node in a graph is just a node. It may carry some local properties — a title, a value, a timestamp — but it has no inherent obligation to declare what it "is," how it should be used, or what it means.”

TERM	DEFINED?	THE DOCUMENT'S WORDS, ANCHORED
schema-first The type system tells you what something is before you encounter it.	DEFINED	§ Nodes Have No Obligation to Explain Themselves · bytes 1,559–1,655 ↗ “schema-first thinking, where the type system tells you what something is before you encounter it”
the Lexicon Not a dictionary or a schema registry: curated anchor nodes, cross-graph edges, traversal tools and overridable bootstrap definitions.	DEFINED	§ What the Lexicon Actually Provides · bytes 25,576–25,638 ↗ “the Issues-FS Lexicon is not a dictionary or a schema registry”
the Semantic Web The community that identified the right problem: exchanging meaning without agreeing on everything upfront.	DEFINED	§ The Semantic Web's Insight (and Mistake) · bytes 15,758–15,898 ↗ “The Semantic Web community identified the right problem: how do independent parties exchange meaning without agreeing on everything upfront?”
the confidence spectrum Every assertion about a node sits on a spectrum from no edges to rich multi-hop connectivity.	DEFINED	§ The Confidence Spectrum · bytes 19,501–19,577 ↗ “Every assertion the system makes about a node sits on a confidence spectrum:”
the fractal principle The same structural pattern, nodes with edges and meaning through connectivity, repeats at every level of zoom.	DEFINED	§ Every Scope Defines Its Own Vocabulary · bytes 12,184–12,297 ↗ “the same structural pattern — nodes with edges, meaning through connectivity — repeats at every level of zoom”
version pinning A cross-graph edge may pin meaning to a version or float with the latest; the graph records the choice.	DEFINED	§ Versioning and Temporal Graphs · bytes 24,745–24,845 ↗ “If the edge points to `osbot-utils@3.63.4`, then the meaning is pinned to that version's definition.”
ROLE.md Used as the thing that makes a role meaningful; its own format is never defined here.	USED, NEVER DEFINED	§ This Applies to Everything · bytes 4,596–4,733 ↗ “If it links to a ROLE.md definition, to a set of issue types it can create, to handoff protocols — then we know what this QA role does.”
Type_Safe The runtime type framework the port example leans on; assumed, not defined here.	USED, NEVER DEFINED	§ The Safe_UInt__Port Example · bytes 1,998–2,067 ↗ “Consider a concrete example from the osbot-utils Type_Safe framework:”

TERM	DEFINED?	THE DOCUMENT'S WORDS, ANCHORED
the Librarian Used twice as the agent who adds or arbitrates edges; never defined in this document.	USED, NEVER DEFINED	§ Name Clashes Are Normal · bytes 14,471–14,515 ↗ “A human or a Librarian agent can then decide”

2 · The claims, by how the document supports them

27 claims. **demonstrated** means the document backs the claim with a worked example in its own text (7); **argued** means reasoning is given (15); **declared** means stated without support (5). This is the document's own evidence state, which is not the same thing as being right.

CLAIM	SUPPORT	ABOUT	ANCHOR
the central claim Everything is a graph; meaning is discovered through relationships, not declared; confidence is proportional to connectivity.	DEMONSTRATED	meaning through connectivity, connectivity	§ What This Document Is · bytes 655–862 ↗ “everything is a graph, meaning is not declared but discovered through graph relationships, and confidence in that meaning is proportional to how richly connected a node is to other nodes that provide context”
the difference is connectivity Two identical values differ radically in meaning when their connectivity differs.	DEMONSTRATED	connectivity, meaning through connectivity	§ The Safe_UInt__Port Example · bytes 3,836–3,946 ↗ “**The difference is not in the value.** Both scenarios have `8080`. The difference is in the **connectivity**.”
common ground is discovered, not imposed Structural overlap can be found across five different review processes without any team conforming to a shared schema.	DEMONSTRATED	compatibility	§ Finding Common Ground Through Graph Analysis · bytes 10,783–10,852 ↗ “The common ground was *discovered* in the graph, not *imposed* on it.”
connection is structural, not nominal A Japanese-labelled process connects to an English-labelled anchor through edges, with no shared label needed.	DEMONSTRATED	compatibility, name clash	§ The QA Engineer in Tokyo · bytes 31,997–32,076 ↗ “The connection is structural (via graph edges), not nominal (via shared label).”

CLAIM	SUPPORT	ABOUT	ANCHOR
the review chain is traceable A three-city review chain in three languages is traceable end to end through the graph.	DEMONSTRATED	cross-graph edge, compatibility	§ The Compliance Officer in Frankfurt · bytes 33,087–33,135 ↗ “The review chain is traceable through the graph.”
absence is reported, not papered over The system does not pretend a GitHub issue is a Feature_Request; it reports the missing edge and what adding it would enable.	DEMONSTRATED	honest uncertainty	§ The Open Source Contributor in Nairobi · bytes 34,629–34,741 ↗ “The system doesn't pretend the GitHub issue is a Feature_Request just because a human would make that inference.”
the document compresses itself to ten principles The summary lists ten core principles. The first edition distilled five ideas from this source; the gap is ladder finding 3.	DEMONSTRATED	meaning through connectivity	§ Summary: Core Principles · bytes 34,869–35,002 ↗ “1. **Everything is a node.** Nodes carry local properties but have no obligation to declare what they are or how they should be used.”
the graph provides the evidence or it does not The system reports what it can and cannot confirm rather than assuming meaning.	ARGUED	honest uncertainty	§ This Applies to Everything · bytes 5,044–5,214 ↗ “The graph either provides the evidence or it doesn't. The system's job is to report what it can and can't confirm, not to assume meaning where the edges don't support it.”
declared is brittle, discovered is resilient Declared meaning breaks at boundaries; discovered meaning thins but composes.	ARGUED	declared meaning, discovered meaning	§ Meaning Is Not Declared — It Is Discovered · bytes 5,775–5,812 ↗ “declared meaning is brittle and local”
compatibility is a spectrum Compatibility is not binary, not symmetric, not global, and it is computable.	ARGUED	compatibility	§ Compatibility Is a Graph Computation · bytes 11,135–11,232 ↗ “**Not binary** — it's a spectrum (Team A and Team C are more compatible than Team A and Team B)”

CLAIM	SUPPORT	ABOUT	ANCHOR
no registration with the parent A child scope never needs to register its vocabulary with a parent scope.	ARGUED	local vocabulary, the fractal principle	§ Every Scope Defines Its Own Vocabulary · bytes 13,110–13,195 ↗ “At no level does a child scope need to "register" its vocabulary with a parent scope.”
resolution is structural Name clashes resolve by graph identity and traversal path, not by the label.	ARGUED	name clash	§ Name Clashes Are Normal · bytes 13,784–13,822 ↗ “Resolution is structural, not nominal.”
observations, not errors Gaps, partial mappings and contradictions are surfaced for judgement, not rejected as failures.	ARGUED	honest uncertainty	§ Incompatibility and Missing Pieces Are Expected · bytes 15,417–15,544 ↗ “None of these are errors. They are observations that the graph analysis can make, surfacing them for human (or agent) judgment.”
the Semantic Web's mistake The mistake was attaching meaning to nodes rather than deriving it from edges.	ARGUED	the Semantic Web	§ The Semantic Web's Insight (and Mistake) · bytes 16,155–16,240 ↗ “They ended up attaching meaning *to nodes* rather than deriving meaning *from edges*.”
mappings can come from anyone Cross-reference edges may be added by the team, an agent, an analysis or a third party, without touching the original node.	ARGUED	anchor node, cross-graph edge	§ The Graph-First Correction · bytes 17,960–18,102 ↗ “the mappings might have been added by anyone: the team itself, a Librarian agent, an automated analysis, or a third party doing an integration”
anchors have no special authority An anchor node enables discovery; it does not force conformity.	ARGUED	anchor node	§ What Anchor Nodes Actually Are · bytes 18,665–18,748 ↗ “**Has no special authority** — it does not force conformity; it enables discovery”
confidence is computable How confident the system is about a node is always computable from the graph.	ARGUED	the confidence spectrum, connectivity	§ The Confidence Spectrum · bytes 20,517–20,671 ↗ “the answer is always computable from the graph: count the edges, measure the depth of traversal to anchor nodes, assess the authority of the nodes reached”

CLAIM	SUPPORT	ABOUT	ANCHOR
grow the graph, do not tighten the schema The ecosystem improves by adding edges, not by imposing stricter schemas.	ARGUED	enrichment	§ Enrichment, Not Enforcement · bytes 22,264–22,322 ↗ “not by imposing stricter schemas, but by growing the graph”
each graph is sovereign Cross-graph edges give interoperability across teams, languages, cultures and agendas without any graph changing internally.	ARGUED	cross-graph edge	§ Cross-Graph References · bytes 24,253–24,366 ↗ “This is what makes the graph model scale across teams, languages, cultures, and agendas. Each graph is sovereign.”
pin or float, but say which Pinned and floating version edges are both valid; the graph makes the choice explicit and traceable.	ARGUED	version pinning	§ Versioning and Temporal Graphs · bytes 24,939–25,013 ↗ “Both are valid choices. The graph makes the choice explicit and traceable.”
local override is a capability Any scope may override or specialise bootstrap definitions by creating local nodes with different subgraphs.	ARGUED	local vocabulary, the Lexicon	§ What Every Repo Can Do · bytes 26,681–26,778 ↗ “**Override or specialise** bootstrap definitions by creating local nodes with different subgraphs”
we know, we think, we cannot confirm The system's observations end in a three-state honest assessment.	ARGUED	honest uncertainty, the confidence spectrum	§ What the System Can Observe · bytes 27,393–27,471 ↗ “**Generate honest assessments**: "We know X. We think Y. We cannot confirm Z.””
not a metaphor The claim is a literal architectural principle, not a metaphor.	DECLARED	meaning through connectivity	§ What This Document Is · bytes 867–930 ↗ “This is not a metaphor. It is a literal architectural principle”
no edges, no meaning A node connected to nothing is, literally, meaningless.	DECLARED	node, connectivity	§ Nodes Have No Obligation to Explain Themselves · bytes 1,747–1,804 ↗ “A node connected to nothing is meaningless — literally.”

CLAIM	SUPPORT	ABOUT	ANCHOR
schema-first in graph syntax Self-describing nodes are schema-first thinking wearing graph syntax.	DECLARED	the Semantic Web, schema-first	§ The Semantic Web's Insight (and Mistake) · bytes 16,559–16,640 ↗ “This is schema-first thinking dressed in graph syntax. The node knows what it is.”
gaps are never filled with assumptions The system reports found and missing edges and never fills gaps by assuming.	DECLARED	honest uncertainty	§ Honest Uncertainty Is the Default · bytes 21,676–21,720 ↗ “It never fills in the gaps with assumptions.”
authority is not identity Authority is a property of connectivity, provenance and governance, not of identity.	DECLARED	anchor node, connectivity	§ What the System Never Does · bytes 27,855–27,939 ↗ “authority is a property of connectivity, provenance, and governance, not of identity”

3 · Hypotheses, objectives, and the document's own demonstrations

KIND	NODE	ANCHOR
HYPOTHESIS	**distributed vocabulary produces gaps** A fractal system with distributed vocabulary will inevitably produce gaps, partial mappings and contradictions.	§ Incompatibility and Missing Pieces Are Expected · bytes 14,652–14,721 ↗ “A fractal system with distributed vocabulary will inevitably produce:”
HYPOTHESIS	**meaning thins measurably at boundaries** Crossing a boundary thins meaning rather than breaking it, and the system can report exactly how thin.	§ Meaning Is Not Declared — It Is Discovered · bytes 6,206–6,332 ↗ “it just means some edges don't extend across that boundary, and the meaning becomes thinner (less confident) on the other side”
HYPOTHESIS	**every edge grows self-understanding** Each added edge increments the system's understanding of itself; the ecosystem improves this way over time.	§ Enrichment, Not Enforcement · bytes 22,324–22,402 ↗ “Every edge added is a small increment in the system's understanding of itself.”
OBJECTIVE	**be the foundation** Underpin every other document in the ecosystem: the philosophy everything else is built on.	§ What This Document Is · bytes 290–352 ↗ “This is the foundational document for the Issues-FS ecosystem.”

KIND	NODE	ANCHOR
EXAMPLE	**the Safe_UInt__Port example** Two scenarios with the identical value 8080, differing only in connectivity, yield radically different meaning. demonstrates: the difference is connectivity, the central claim	§ The Safe_UInt__Port Example · bytes 2,079–2,112 ↗ “class Safe_UInt__Port(Safe_UInt):”
EXAMPLE	**the five Reviews** Five teams' Review processes, none aware of the others, analysed for structural overlap without any conforming. demonstrates: common ground is discovered, not imposed, compatibility is a spectrum	§ The Review Example · bytes 6,559–6,629 ↗ “Consider five different teams, each with a process they call "Review":”
EXAMPLE	**the prefLabel counter-example** A SKOS prefLabel declaration shown as meaning encoded on the node itself: the mistake, in six lines. demonstrates: the Semantic Web's mistake, schema-first in graph syntax	§ The Semantic Web's Insight (and Mistake) · bytes 16,280–16,307 ↗ “skos:prefLabel "Review"@en”
EXAMPLE	**the end-to-end scenario** London, Tokyo, Frankfurt and Nairobi: one task, three review vocabularies, one unmapped proposal, every confidence statement computed. demonstrates: the review chain is traceable, connection is structural, not nominal, absence is reported, not papered over	§ The Graph · bytes 28,552–28,592 ↗ “Task-42 (in Project: Issues-FS__Service)”

4 · The ontology: relations the document itself asserts

Concept-to-concept edges, each with its declared inverse per the house grammar, and each anchored to the sentence that asserts it. These are the document's relations, carried; the bridge layer (layer 2) will add cross-document edges as authored decisions.

FROM	VERB (INVERSE)	TO	ANCHOR
meaning through connectivity	`departs-from` (departed-from-by)	schema-first	§ Nodes Have No Obligation to Explain Themselves · bytes 1,523–1,580 ↗ “This is a deliberate departure from schema-first thinking”

FROM	VERB (INVERSE)	TO	ANCHOR
connectivity	`determines` (determined-by)	the confidence spectrum	§ What This Document Is · bytes 750–826 ↗ “confidence in that meaning is proportional to how richly connected a node is”
the Lexicon	`provides` (provided-by)	anchor node	§ What the Lexicon Actually Provides · bytes 25,651–25,721 ↗ “**A curated set of well-connected anchor nodes** for the issues domain”
anchor node	`enables` (enabled-by)	compatibility	§ What Anchor Nodes Actually Are · bytes 19,091–19,180 ↗ “A node in a child scope that links to a Lexicon anchor node gains meaning by association.”
cross-graph edge	`enables` (enabled-by)	compatibility	§ Cross-Graph References · bytes 24,367–24,470 ↗ “Cross-graph edges create interoperability without requiring any graph to change its internal structure.”
enrichment	`remedies` (remedied-by)	the confidence spectrum	§ Enrichment, Not Enforcement · bytes 21,755–21,843 ↗ “When the system identifies low-confidence nodes, the response is not validation failure.”
the fractal principle	`licenses` (licensed-by)	local vocabulary	§ Every Scope Defines Its Own Vocabulary · bytes 12,063–12,152 ↗ “Every level in this hierarchy can introduce its own nodes, edges, types, and definitions.”
graphs of graphs	`exhibits` (exhibited-by)	the fractal principle	§ The Recursive Structure · bytes 23,340–23,381 ↗ “At every level, the same principles apply”

5 · The thesaurus, and the distinctions drawn on purpose

	PAIR	ANCHOR
ALSO CALLED	meaning through connectivity ↔ **meaning comes from edges**	§ Summary: Core Principles · bytes 35,007–35,118 ↗ “**Meaning comes from edges.** What a node "is" emerges from the graph relationships that can be traced from it.”
ALSO CALLED	the fractal principle ↔ **graphs of graphs of graphs**	§ Every Scope Defines Its Own Vocabulary · bytes 11,752–11,809 ↗ “It is graphs of graphs of graphs, at every level of scope”

	PAIR	ANCHOR
NEAR BUT NOT	anchor node is **not** a schema other nodes must conform to	§ The Graph-First Correction · bytes 17,075–17,181 ↗ “It is not a schema that other nodes must conform to. It is a reference point that other nodes may link to.”
NEAR BUT NOT	the Lexicon is **not** a dictionary or a schema registry	§ What the Lexicon Actually Provides · bytes 25,576–25,638 ↗ “the Issues-FS Lexicon is not a dictionary or a schema registry”
NEAR BUT NOT	compatibility is **not** a shared type declaration	§ Compatibility Is a Graph Computation · bytes 10,939–11,006 ↗ “Two nodes are not compatible because they share a type declaration.”
NEAR BUT NOT	discovered meaning is **not** conformity imposed by analysis	§ Finding Common Ground Through Graph Analysis · bytes 10,662–10,782 ↗ “This analysis happened without any team changing their process, learning a shared vocabulary, or conforming to a schema.”
NEAR BUT NOT	graphs of graphs is **not** graph syntax over self-describing nodes	§ The Semantic Web's Insight (and Mistake) · bytes 16,559–16,613 ↗ “This is schema-first thinking dressed in graph syntax.”

6 · The local graph, drawn

The graph is an interactive view: it lives in the web page's side panel, where clicking a node opens both the extraction row and the cited bytes in the source. A static rendering would be a decoration here; the tables above are the graph's content in full.

7 · The taxonomy, and the coverage rule

The document's own structure, with what each section yielded. The rule this build enforces: every section with prose either yields at least one anchored item or is recorded as empty with a reason. 31 sections carry prose; 2 are deliberately empty.

SECTION	YIELD
What This Document Is	3 NODE(S), 1 EDGE(S)
Part 1: Everything Is a Node	structural heading
Nodes Have No Obligation to Explain Themselves	5 NODE(S), 1 EDGE(S)
The Safe_UInt__Port Example	3 NODE(S)

SECTION	YIELD
This Applies to Everything	2 NODE(S)
Part 2: Meaning Through Connectivity	structural heading
Meaning Is Not Declared — It Is Discovered	4 NODE(S)
The Review Example	1 NODE(S)
Finding Common Ground Through Graph Analysis	1 NODE(S), 1 PAIRING(S)
Compatibility Is a Graph Computation	2 NODE(S), 1 PAIRING(S)
Part 3: The Fractal Principle	structural heading
Every Scope Defines Its Own Vocabulary	3 NODE(S), 1 EDGE(S), 1 PAIRING(S)
Name Clashes Are Normal	3 NODE(S)
Incompatibility and Missing Pieces Are Expected	2 NODE(S)
Part 4: Anchor Nodes and Reference Graphs	structural heading
The Semantic Web's Insight (and Mistake)	4 NODE(S), 1 PAIRING(S)
The Graph-First Correction	1 NODE(S), 1 PAIRING(S)
What Anchor Nodes Actually Are	2 NODE(S), 1 EDGE(S)
Part 5: Confidence as a Function of Connectivity	structural heading
The Confidence Spectrum	2 NODE(S)
Honest Uncertainty Is the Default	2 NODE(S)
Enrichment, Not Enforcement	3 NODE(S), 1 EDGE(S)
Part 6: Graphs of Graphs	structural heading
The Recursive Structure	1 NODE(S), 1 EDGE(S)
Cross-Graph References	2 NODE(S), 1 EDGE(S)
Versioning and Temporal Graphs	2 NODE(S)

SECTION	YIELD
Part 7: Practical Implications for Issues-FS	structural heading
What the Lexicon Actually Provides	1 NODE(S), 1 EDGE(S), 1 PAIRING(S)
What Every Repo Can Do	1 NODE(S)
What the System Can Observe	1 NODE(S)
What the System Never Does	1 NODE(S)
Part 8: A Worked Example — End to End	EMPTY ON PURPOSE Announces the worked example; the example itself is the node anchored in The Graph.
The Setup	EMPTY ON PURPOSE Scene-setting for the worked example; its content is carried by the end-to-end example node.
The Graph	1 NODE(S)
The QA Engineer in Tokyo	1 NODE(S)
The Compliance Officer in Frankfurt	1 NODE(S)
The Open Source Contributor in Nairobi	1 NODE(S)
Summary: Core Principles	1 NODE(S), 1 PAIRING(S)