3.4 Traceability to Part 2 Distributed Node-Based Logical Architecture / Platform-Independent Model (PIM)

Go To Traceability to Parent and Source Architectures

Part 2 defines the reusable, distributed, node-based Logical Architecture / Platform-Independent Model (PIM). This part preserves that logical architecture by mapping selected Part 2 logical concepts, as specialized by Part 3, to Phase 0 implementation artifacts.

The Phase 0 Implementation Profile realizes selected logical Nodes as implementation participants, selected logical Communication Endpoints as DDS Topics or equivalent communication artifacts, selected logical information structures as IDL structures and generated types, selected Runtime Plane distinctions as implementation conventions, and selected logical interaction patterns as implementation-level message flows or command flows.

Table 3-4: Part 2 logical elements realized by the Phase 0 Implementation Profile.

Part 2: Logical Element	Part 4: Phase 0 implementation treatment
Logical Node	Realized by selected implementation Nodes, modules, processes, or runtime participants.
Logical Node Role	Realized by implementation responsibilities, modules, functions, or service behaviors.
Logical Communication Endpoint	Realized by DDS Topics or other selected communication artifacts.
Logical Data Structure Definition	Realized by IDL structures, generated types, and implementation data definitions.
Logical Data Structure Instance	Realized by DDS samples, runtime messages, commands, acknowledgements, records, and generated type instances.
Logical Runtime Plane	Realized through implementation conventions that distinguish Control, Data, Health and Observability, Policy and Release, and Audit and Provenance concerns.
Logical Interaction Pattern	Realized by implementation-level communication flows, command flows, processing steps, scripts, or orchestration conventions.
Logical Governance Model	Realized through implementation versioning, compatibility rules, generation rules, repository conventions, and review controls.
Logical Traceability Model	Realized through mapping tables, file headers, generated artifact metadata, repository metadata, and traceability records.