Robotic System Standardization via Structural Field Composition

by Nick Clark | Published March 27, 2026 | PDF

A warehouse deploying robots from three manufacturers faces a standardization problem that ROS topics and DDS middleware cannot solve. The robots can exchange messages, but they cannot verify each other's governance constraints, negotiate capability boundaries, or participate in shared trust relationships. Canonical agent schema addresses this by defining what a robotic agent is through typed structural fields, enabling standardization at the identity and governance level rather than the communication level.

The standardization gap in multi-vendor robotics

Robotic system integration today is a custom engineering effort for every deployment. A logistics company deploying picking robots from one vendor, transport robots from another, and packing robots from a third must build custom integration layers for each combination. ROS provides a common communication framework, but communication is not coordination. The robots can publish to shared topics, but they have no structural way to verify each other's capabilities, negotiate task boundaries, or establish trust.

The result is that multi-vendor robotic deployments are brittle. Every new vendor requires a new integration effort. Every software update from one vendor risks breaking the integration with others. The operational overhead of maintaining multi-vendor robotic systems consumes engineering resources that should be spent on operational improvement.

Why communication standards do not solve robot standardization

ROS 2, DDS, and OPC-UA provide standardized communication between robotic systems. They define message formats, transport protocols, and discovery mechanisms. But they do not define what a robot is as a governed, capable, trustworthy agent.

Two robots can communicate via DDS while having completely incompatible safety models, incompatible capability representations, and no shared governance framework. A picking robot that declares itself capable of handling fragile items through a vendor-specific capability format cannot be verified by a transport robot from a different vendor. The communication works. The structural interoperability does not.

How canonical agent schema addresses this

The canonical agent schema defines six typed fields that apply to any robotic agent regardless of manufacturer. The governance field carries the robot's operational policy: safety constraints, authorized zones, maximum speeds, payload limits. The capability field declares what the robot can physically do: reach envelope, payload capacity, sensor modalities, locomotion type. The trust field tracks the robot's relationships with other agents in its operational environment.

These fields are structural, not vendor-specific. A picking robot from any manufacturer expresses its capabilities in the same typed format. A transport robot from any manufacturer can evaluate those capabilities against its own governance constraints before accepting a handoff. The evaluation happens through field comparison, not through vendor-specific API calls.

Role typing enables robots to declare their operational role as a structural property. A picking robot, a transport robot, and a packing robot carry different role types with different capability requirements and governance constraints. Role compatibility is evaluated structurally: a picking robot can delegate to a transport robot if the transport robot's capability field satisfies the delegation requirements in the picking robot's governance policy.

What implementation looks like

A warehouse deploying canonical agent schema wraps each vendor's robot in a schema adapter that maps the vendor's proprietary state representation to the six canonical fields. The adapter runs on the robot's compute module or on a lightweight edge proxy. Once mapped, robots from different vendors participate in shared governance through their canonical fields.

For manufacturers, adopting canonical agent schema means their robots can interoperate with any other schema-compliant robot without bilateral integration agreements. A robot manufacturer that ships products with canonical schema support opens its addressable market to every deployment that uses the standard.

For system integrators, canonical agent schema reduces the integration effort from quadratic to linear. Instead of building custom adapters between every pair of vendors, the integrator maps each vendor to the canonical schema once. Interoperability between vendors follows from the schema definition.

Agent Schema All 21 steps →

Define what an autonomous agent is — structurally.

Patent
US 19/452,651 · filed
Primary Technical Disclosure

AQ

deterministic
autonomy

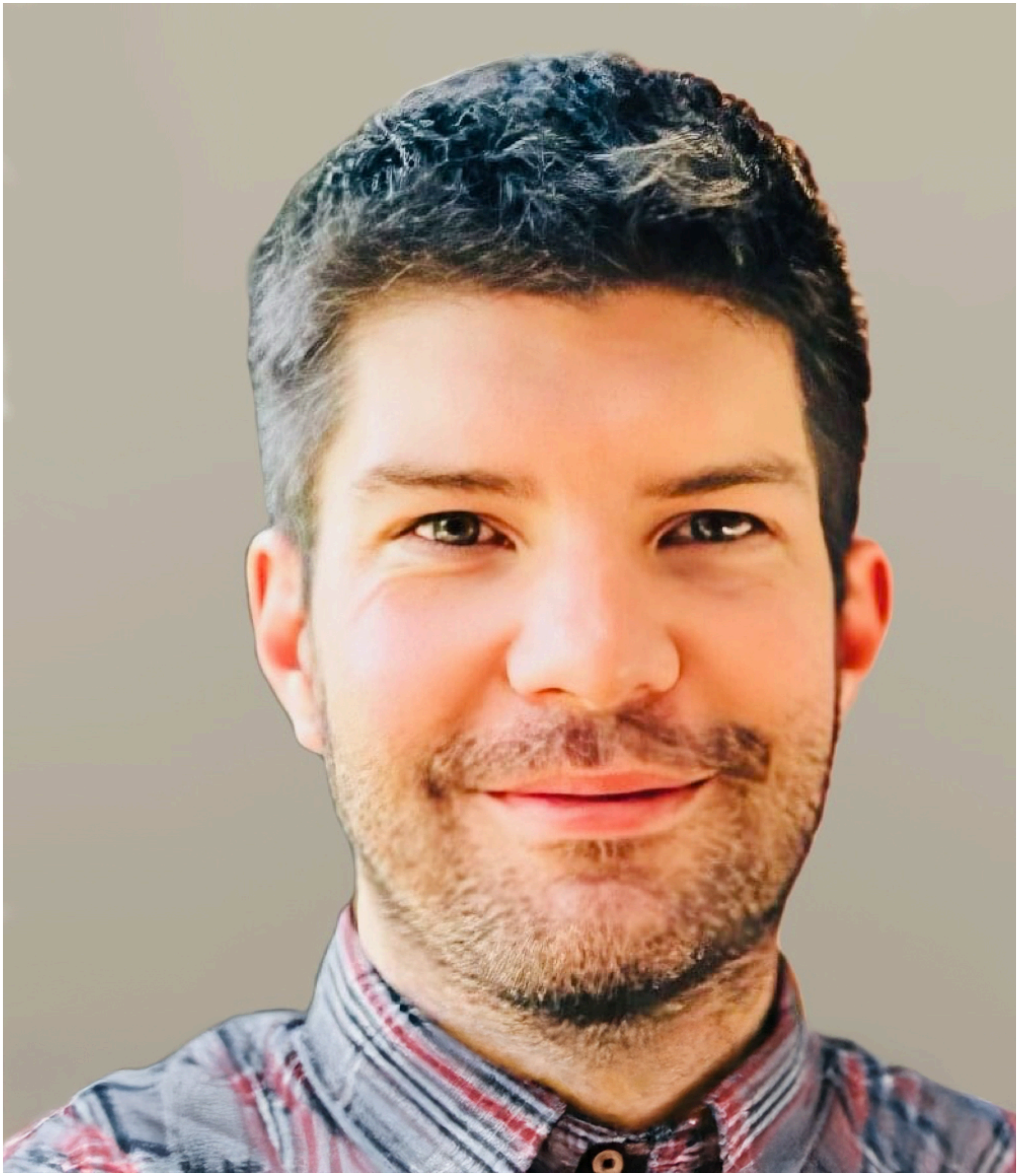
Legal

Last updated: 2026-03-03

-
- Inventive Steps
- Licensing
- Patents
- Articles
- Legal

- [Opportunities]
- [Sitemap]

-
- nick@qu3ry.net
- 72 28 14 36 01



Invented by Nick Clark | Founding Investors: Devin Wilkie