Ontollo.

CONSTRUCTION

Turning field reality into operational decisions.

THE CHALLENGE

Every general contractor and subcontractor already runs good software. Procore or PlanGrid for the project. Epicor, InEight, or Dynamics for the back office. Telematics on the equipment. Plus whatever point tools a project team picked up along the way. That is not the problem.

Construction margins are notoriously thin, often mid-single digits net. McKinsey estimates that better use of project data alone can lift industry productivity by roughly 14 to 15 percent, with AI capable of adding meaningfully more once it has real operational context to work with.

On a business running a 3 to 5 percent margin, that is not a rounding error. It is the difference between a job that makes money and one that does not. When a pour happens at a specific site on a specific day, precision matters. Equipment at Site A that needs to move to Site B ripples into three other projects. A missed spec becomes a rework. An unbilled hour becomes a margin leak. At these margins, small leaks sink the boat.

Those systems capture reality only if skilled people stop doing their real job to feed them. A superintendent who knows a pour cold still spends hours re-entering it, field by field, into a daily report. An installer who knows exactly how a builder's spec differs from standard has no system that catches it before it becomes a rework order. **The knowledge exists. It just never reaches the decision in time.**

WHY AN AGENTIC APPROACH WILL NOT SOLVE IT

Most operators start in the same place. Point a frontier model like Claude or ChatGPT, or a copilot already embedded in a system you own, at the paperwork: the daily report, the RFI, the compliance sign-off. It comes back in seconds instead of two hours, and that speed is real.

The accuracy is not. Each of these assistants works from its own dataset and its own version of reality, so each one reasons from an incomplete picture and has no way to know it. The scheduling AI moves equipment without knowing it is already committed to three other projects. Field capture logs a slip without feeding cost-to-complete or the billing claim. Add more agents and you get faster individual decisions, coordinated worse than before.

The underlying reason is simple. Knowledge work **is** language, so an assistant that reads and writes language is enough. Operational work is not. It is physical things, in states, at locations, over time, in relationships. The daily report, the schedule, and the telematics feed **describe** the work. They are not the work.

And the record goes stale the moment it is written. Stale operational data is close to worthless, because it becomes faster to guess than to look it up. Precision has to come from voice and inference, pulled out of work people are already doing, not from a new burden laid on top of it.

IT'S A KNOWLEDGE GRAPH PROBLEM

Every operation has a system of record: ERP, CRM, project management, scheduling, cost tracking. They capture what already happened. Almost none has a system of reality: a live, connected model of what is happening right now across every asset, signal, and person.

That layer is a knowledge graph. Not a dashboard, not a data lake, not an assistant. A governed model in which a crew, a piece of equipment, a pour, a cost code, a contract clause, a builder spec, and a schedule dependency are all objects with real relationships, so a question crosses them in one hop instead of five exports and a week.

Gartner now treats this as infrastructure rather than architectural preference, and projects that more than half of AI agent systems will run on context graphs by 2028. Penetration today is under one percent. The window is open, and it will not stay open.

Ontollo.

ONTOLLO IS THAT LAYER

Ontollo maps how your sites, crews, equipment, systems, and people actually connect, and keeps that map current as the job changes. One living graph, not a static copy sitting in a warehouse. That is the context AI needs to stop guessing and start reasoning over your operation as it really is.

Ontollo has three primary input streams:

- **Operational signals.**
  Crew and equipment location, field activity, telematics, daily production, material and delivery status, weather.
- **Business drivers.**
  Timeline and schedule, budget and cost codes, contract terms, billing and standby entitlements, cost-to-complete.
- **Knowledge.**
  Why a superintendent sequences a pour that way, how an installer catches a spec mismatch by feel. The judgment your best people carry in their heads.

The output is decisions and tasks routed to the right person, agent, or device, with a citation attached. Not another report.

What that looks like

A daily report in under a minute instead of two hours.

A superintendent narrates the day's pour the way he already talks about it on site. Ontollo structures it into the report format the team uses, flags the cost code for confirmation, and surfaces standby time that is billable and would otherwise go unclaimed.

One capture, a dozen downstream decisions: equipment utilization across every site, next week's crew planning, cost-to-complete.

Builder-spec variation caught before it becomes reworked.

Every production builder specs things differently: drill hole placement, hardware, cutout dimensions, varying by floorplan and elevation. Ontollo holds that variation as structured knowledge, so the crew gets the right spec at the right moment instead of finding out after install. A rework order that never gets written is margin that stays in the job.

ARCHITECTURE AND TRUST

Our team spent years building knowledge graphs at Atlassian before GenAI existed. That experience shaped the choices below.

- **Grounded, not guessed.**
  Every input traces to a real report, sensor, or system reading. Any number driving a decision is computed, not inferred. If it cannot be cited, it is not shown.
- **Operator consent.**
  No action is taken without explicit approval. The system recommends. Your people decide.
- **Permissions live in the middle layer.**
  Entitlements are enforced at the graph, not delegated to a shared service account with standing access to your source systems.
- **Canonical catalog.**
  About 50 prebuilt object types common to physical-asset businesses, so the model of your operation comes together in days, not months.
- **Time to value.**
  Pilots stand up in about a week once data access is granted, and each integration we build makes the next similar one faster.
- **Portable and model-agnostic.**
  Run on whichever engine fits your cost, performance, and data-location needs. Retrieval-grounded, never model-trained.
- **No clean slate required.**
  Your ERP, scheduling tool, telematics feeds, and point solutions stay exactly where they are. Ontollo sits above them.

START WITH 30 DAYS

PHASE 1: DISCOVERY *30 days · no cost · you keep everything we produce*	WHAT WE DO *Working proof of concept on your data, not a mockup*
• Map the technology landscape and how data flows across your systems. • A handful of interviews with the people running the work.	• Connect two or three real data sources into a working proof of concept. • Identify the highest-value bottleneck and what the status quo is costing you.

If Discovery earns the next step, a 60-day pilot follows on that one use case, run live with real users on a real project, measured against success criteria agreed up front, with a clear go or no-go on schedule.