



**BUILD LOCALLY**

# Let Domain Experts Build Their Own Internal Apps

Use contextual coding agents to turn plant knowledge into local Python, HTML and data applications without starting from a blank repository.

**ASSET + OPERATIONS + MAINTENANCE + KNOWLEDGE + AI**

PHOENIX turns industrial data and engineering experience into a coordinated intelligence environment that helps teams protect reliable capacity and improve decisions.

# Let Domain Experts Build Their Own Internal Apps

Many useful industrial applications are too specific for off-the-shelf software yet too small to justify a long enterprise development cycle. Engineers and analysts know what they need, but they may not have time to become full-time software developers.

Generic coding AI can write code, but without plant context it may not know the approved schema, existing APIs, security rules, naming conventions or data boundaries.

## The PHOENIX approach

LabSource gives coding agents access to governed project context, internal APIs, approved database tools, software history and domain knowledge. A local user can describe an application, have agents generate and test it, and use a second model or review agent before publishing the result internally.

## WHY THIS MATTERS

Industrial AI delivers the most value when it can combine machine condition, operating state, maintenance readiness, production demand and expert knowledge without losing governance. The differentiator is not simply access to an LLM; it is the context, tool control, verification and organizational memory surrounding the model.

## A governed context-to-decision loop

|   |                                                      |
|---|------------------------------------------------------|
| 1 | User describes the business problem                  |
| 2 | Agent identifies required data and existing services |
| 3 | CIP exposes approved tools and context               |
| 4 | Coding model creates Python/HTML/API components      |
| 5 | Tests are run in a controlled workspace              |
| 6 | Review model checks architecture/security            |
| 7 | Human approves release                               |

## What PHOENIX adds

| CAPABILITY           | VALUE                                                                             |
|----------------------|-----------------------------------------------------------------------------------|
| Contextual coding    | Code is generated against known plant schemas, APIs and workflows.                |
| Project lineage      | Requirements, branches, tasks and versions can be connected to the software tree. |
| Controlled execution | Terminals, SSH, repositories and secrets remain governed.                         |
| Reusable modules     | Successful internal apps and agents become part of the local capability library.  |

# Where this becomes useful

- Asset inspection tools
- Reliability calculators
- Downtime dashboards
- Custom data transforms
- Internal APIs
- Mobile/PWA workflows

## A practical operating model

PHOENIX is designed to support human decision-making rather than bypass it. Agents can gather evidence, explain patterns, draft code, prepare reports and propose actions. High-impact actions can remain subject to explicit human approval and the permissions enforced by CIP.

## Suggested deployment sequence

|                  |                                                                                                             |
|------------------|-------------------------------------------------------------------------------------------------------------|
| 1. Connect       | Start with approved read-only or narrowly scoped internal data sources and existing PHOENIX sensor context. |
| 2. Contextualize | Map facilities, assets, components, users, production relationships and domain vocabulary.                  |
| 3. Prove         | Choose a small number of high-value workflows with clear evidence and measurable outcomes.                  |
| 4. Capture       | Retain expert corrections, verification results and operational outcomes in the contextual knowledge layer. |
| 5. Scale         | Reuse the same governed tools, models and knowledge across reports, agents and internal applications.       |

|                         |                                                                                                                                                                          |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Business outcome</b> | The organization converts domain expertise into working internal software faster while keeping data, credentials and deployment control inside the governed environment. |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## Keep intelligence useful, bounded and auditable

|                           |                                                                                                                     |
|---------------------------|---------------------------------------------------------------------------------------------------------------------|
| Minimum necessary context | Retrieve only the data required for the task rather than exposing entire systems by default.                        |
| Separated credentials     | Models and prompts should not become the long-term store for passwords, tokens or system secrets.                   |
| Human approval            | Keep appropriate approval gates around changes that affect production systems, maintenance commitments or releases. |
| Evidence traceability     | Preserve source data, tool calls and verification details behind important recommendations.                         |
| Local knowledge ownership | Keep approved domain knowledge and contextual history under customer control.                                       |
| Model independence        | Treat models as replaceable reasoning engines while preserving tools, knowledge, workflows and evaluations.         |

### PHOENIX architecture in one sentence

Sense what is happening, securely connect the surrounding plant context through CIP, use LabSource and AgentSWARMS to select and verify the right local intelligence, then retain the evidence and expert outcome so the organization becomes smarter over time.

### About PHOENIX

PHOENIX combines industrial sensing, edge connectivity, secure contextual data access, local AI orchestration, reporting and application development to help industrial teams protect reliable capacity and improve the quality of engineering and operational decisions.

For demonstration or pilot discussions: [sales@phoenix-unifiedsystems.com](mailto:sales@phoenix-unifiedsystems.com) | 336-763-2882 | Made in the USA