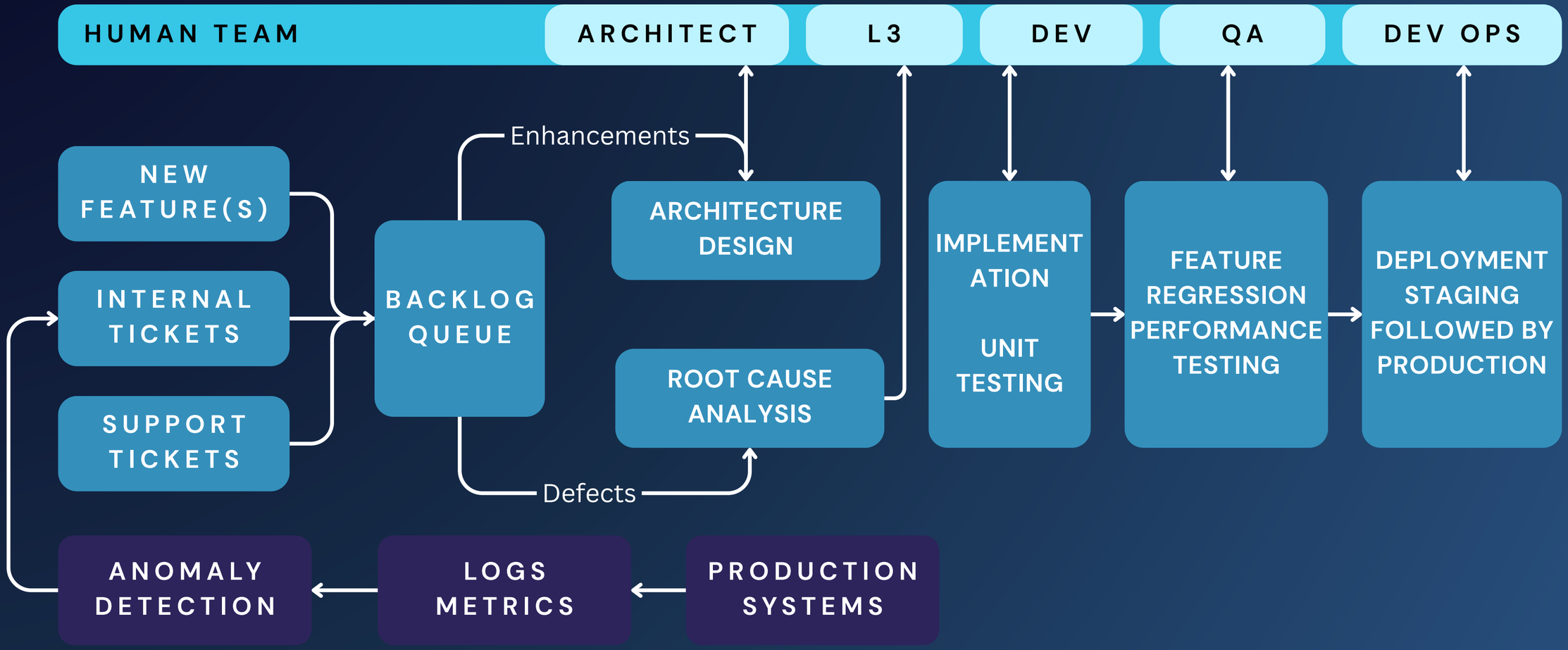
CoWeave.ai

Weave automation.

Orchestrate intelligence in your SDLC.

Traditional SDLC Process



The Current Engineering Reality

The Coordination Tax

- Distributed teams, endless meetings, and approval chains bottleneck the entire pipeline
- Ticket triaging, architecture reviews, code reviews, deployments—all require synchronous coordination

Knowledge Silos

- Critical context trapped in engineers' heads, Slack threads, and outdated wikis
- Constant context switching and steep learning curves for new team members

The LLM Context Tax

- Teams use AI assistants but manually gather context for every prompt
- Copy-pasting code, re-explaining architecture, no memory across tasks
- Engineers spend more time as "context gatherers" than problem solvers

What is **AI-Assisted** S/W Development?

A Fundamental Paradigm Shift

The New Model:

- LLMs perform the **first line of thinking** and work across the SDLC
- Human engineers focus on **review, refinement, and strategic** decisions
- **Human supervision** at every step ensures quality and accountability

LLMs Take the First Pass:

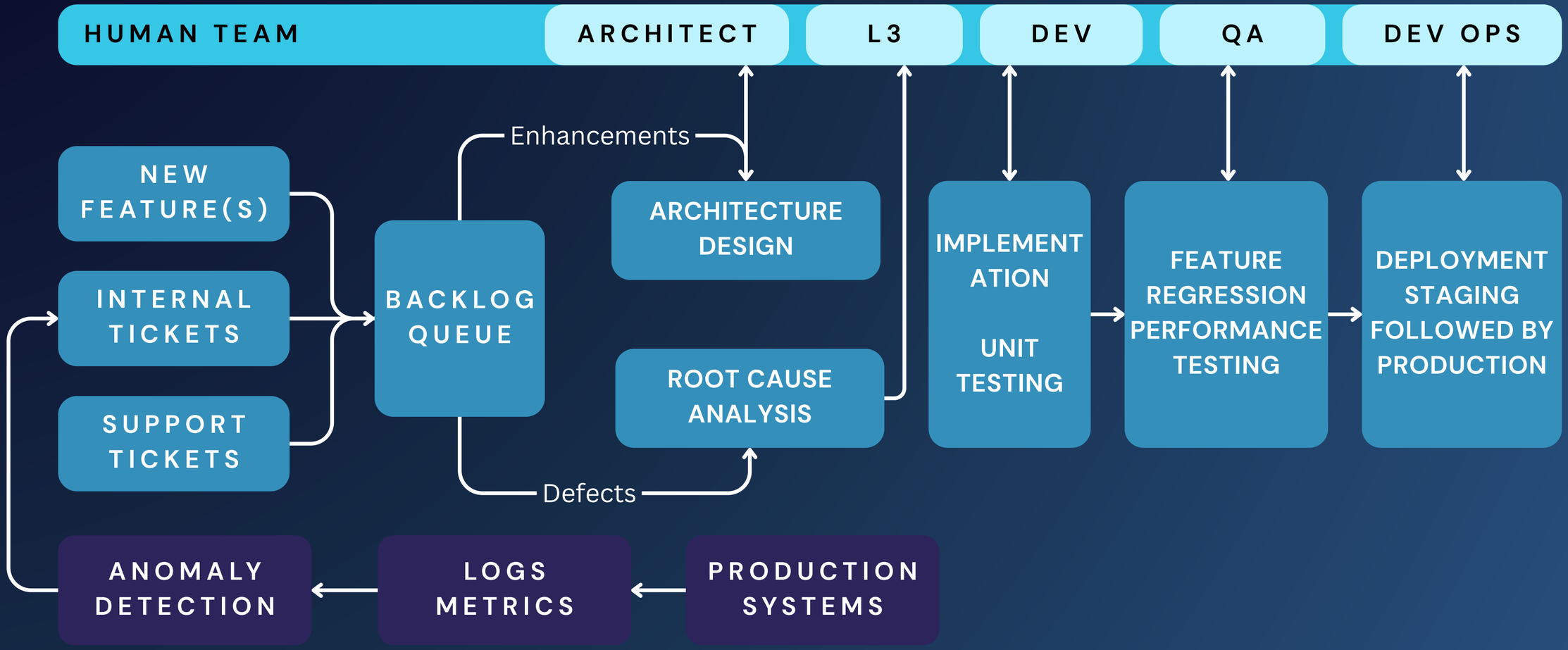
- Architecture and design
- Root cause analysis
- Implementation and testing
- Deployment automation

The Result:

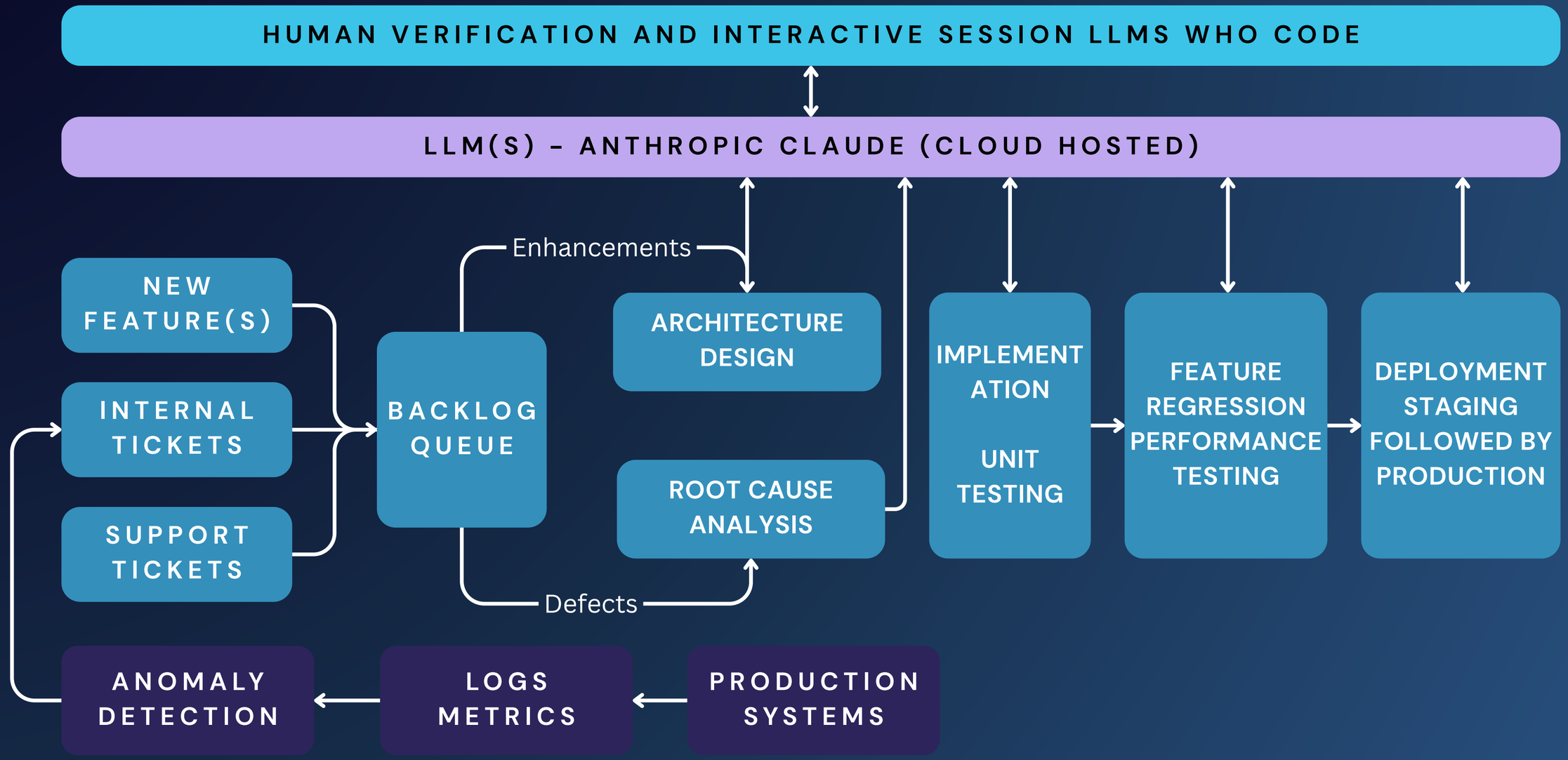
- Eliminate the co**ordination tax**—AI works 24/7, no meetings required
- **Turbo charge** operational efficiency and software reliability
- **Elevate engineers** from executors to expert reviewers

Introducing
CoWeave.ai

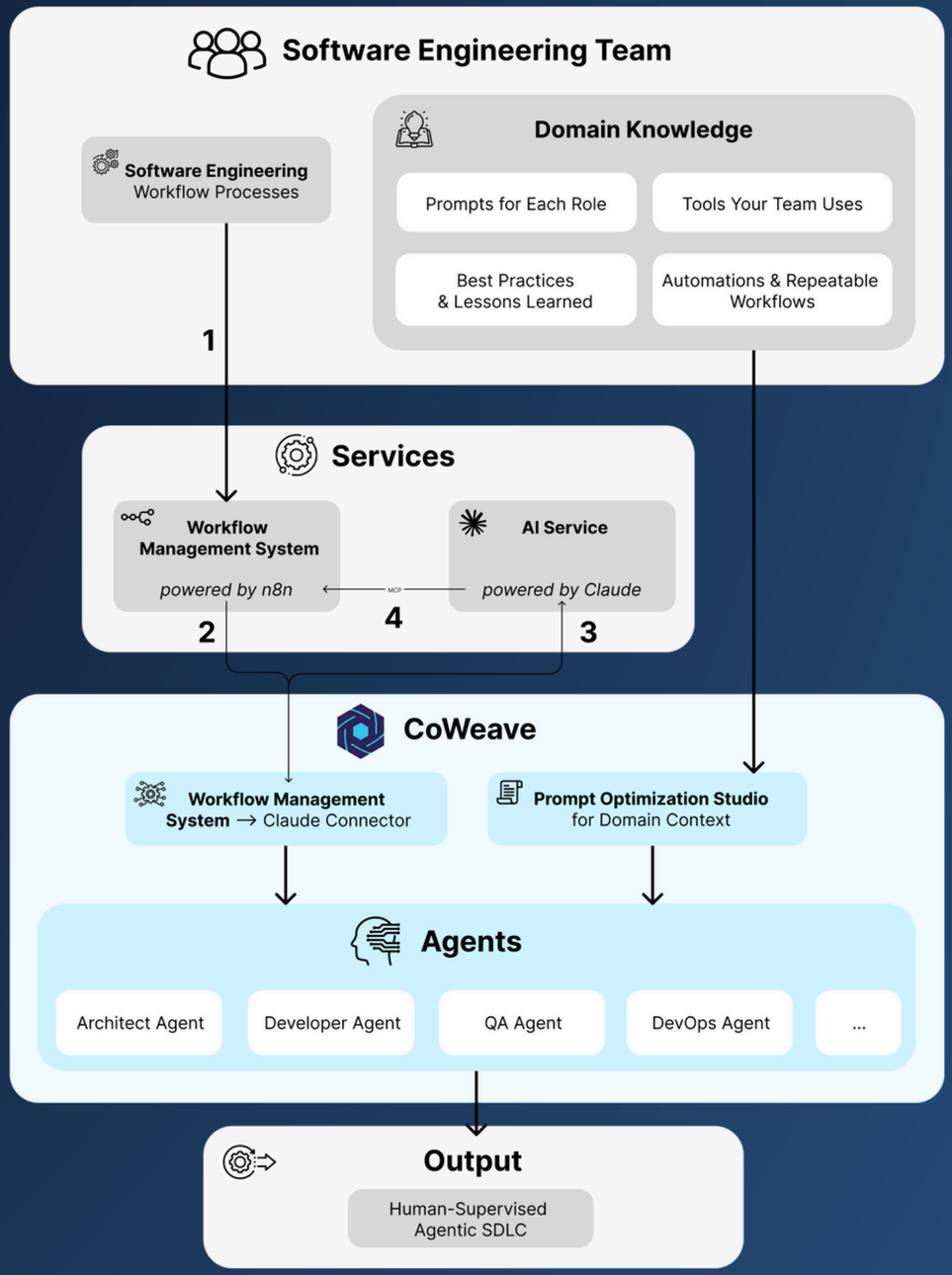
Traditional SDLC Process



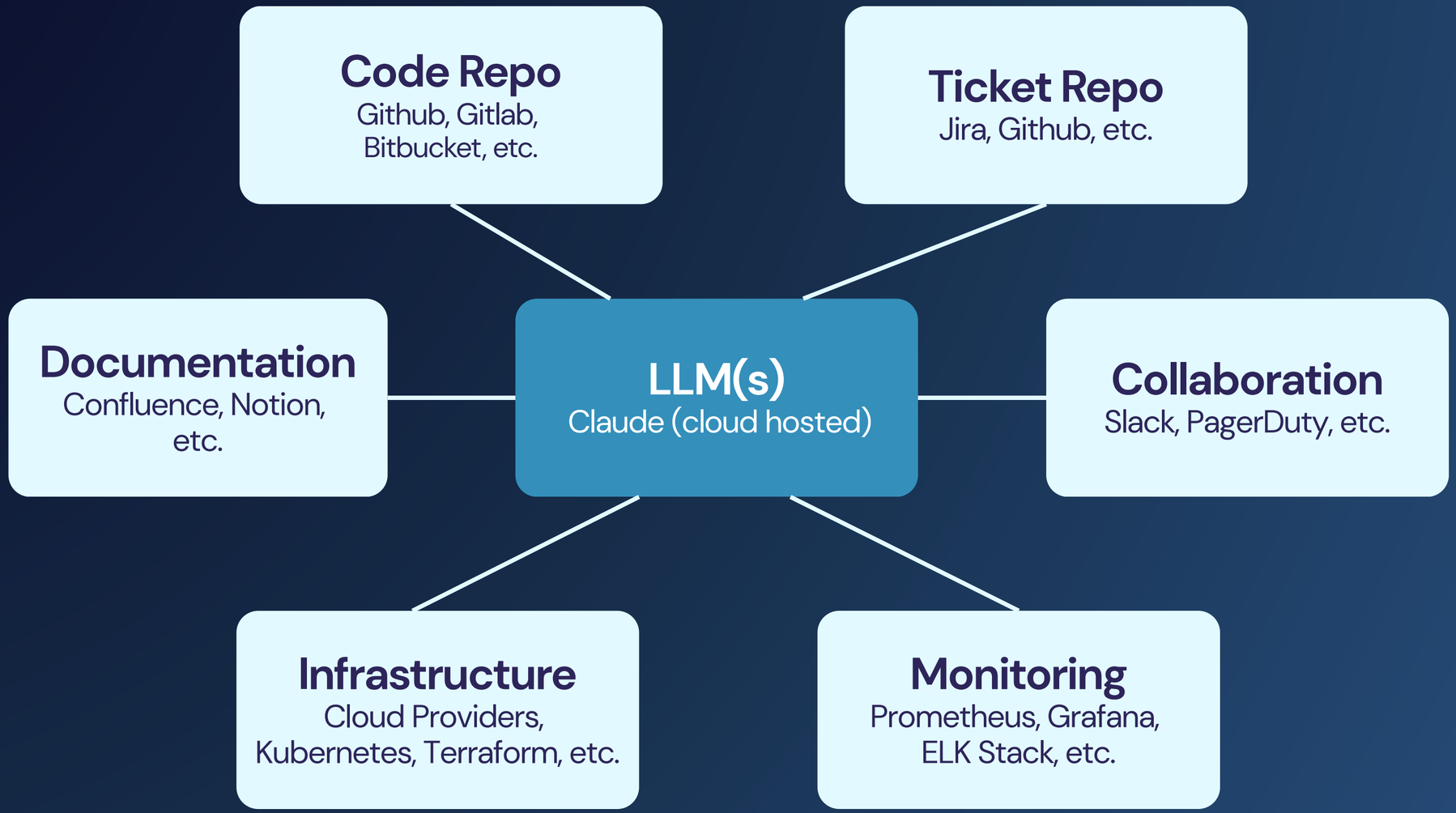
AI-Assisted SDLC Process



CoWeave in a nutshell



Integrate with Your Systems



Why can’t I use **LLM-powered IDEs** directly?

You can and

You Will

with this platform

however...

Generic LLMs have **Key Limitations**

No Institutional Knowledge

Generic LLMs are *novices* to your company. They *don't understand* your unique codebase, internal APIs, or the system-wide impact of a local change.

Process Blindness

They lack an understanding of your team's specific workflows, like your Git process or testing methodologies. You must give them step-by-step instructions for every action.

Context Awareness

Most LLMs cannot remember sufficient conversational history or project context across sessions, forcing engineers to manually copy and paste details for every task.

Constant Manual Work

Engineers must act as "prompt engineers," manually providing rules and context, which makes the process inefficient and prone to error.

Risk of Unreliable Output

Because they lack your domain knowledge, they can produce technically sound but impractical solutions that waste time and introduce new problems.

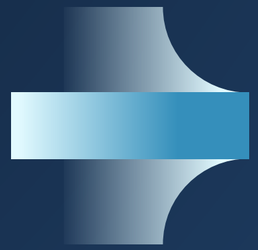
That's Where We Come In

AI as an Enabler

- Generic LLMs become a powerful and customized workhorse
- Human Engineers get a **significant head start**

Adaptive Intelligence

- Automate tasks, reduce errors, and develop with quality
- Accelerate Software Development Lifecycle



Augmentation, Not Overhaul

- Adapt to your specific engineering practices
- No re-engineering required

Value CoWeave Provides

- Create **agentic workflows with domain-specific** knowledge fine-tuned for your specific processes, tech stack, and systems
- Capture and apply **team best practices automatically:** ensure consistency without every engineer manually guiding the AI
- **Purpose-built** integrations with engineering tools and platforms
- **Optimized** for technical decision-making and troubleshooting

Team-Level

SDLC Automation

1. **Product manager** submits PRD
2. **Architect Agent** creates design
3. **Human** interactively reviews design

1. **Development Agent** implements code
2. **Development Agent** creates Git Pull Request (PR)
3. **Human** interactively reviews Git PR
4. **Human** merges PR

1. **Testing Agent** updates feature regression test suites
2. **Human** approves tests
3. **Testing Agent** runs test cases and compares results

1. **Deployment** Agent updates deployment framework
2. **Human** approves deployment framework
3. **Human** deploys changes

1. **Agent** polls ticket repository for bugs and defects
2. **Agent** performs RCA
3. **Agent** updates tickets for human approval

1. **Product manager** submits PRD
2. **Architect Agent** creates design
3. **Human** interactively reviews design

1. **Development Agent** implements code
2. **Development Agent** creates Git Pull Request (PR)
3. **Human** interactively reviews Git PR
4. **Human** merges PR

1. **Testing Agent** updates feature regression test suites
2. **Human** approves tests
3. **Testing Agent** runs test cases and compares results

1. **Deployment** Agent updates deployment framework
2. **Human** approves deployment framework
3. **Human** deploys changes

1. **Agent** polls ticket repository for bugs and defects
2. **Agent** performs RCA
3. **Agent** updates tickets for human approval

1. **Agent** polls ticket repository for bugs and defects
2. **Agent** performs RCA
3. **Agent** updates tickets for human approval

1. **Product manager** submits PRD
2. **Architect Agent** creates design
3. **Human** interactively reviews design

1. **Development Agent** implements code
2. **Development Agent** creates Git Pull Request (PR)
3. **Human** interactively reviews Git PR
4. **Human** merges PR

1. **Deployment** Agent updates deployment framework
2. **Human** approves deployment framework
3. **Human** deploys changes

1. **Testing Agent** updates feature regression test suites
2. **Human** approves tests
3. **Testing Agent** runs test cases and compares results

1. **Agent** polls ticket repository for bugs and defects
2. **Agent** performs RCA
3. **Agent** updates tickets for human approval

1. **Product manager** submits PRD
2. **Architect Agent** creates design
3. **Human** interactively reviews design

1. **Development Agent** implements code
2. **Development Agent** creates Git Pull Request (PR)
3. **Human** interactively reviews Git PR
4. **Human** merges PR

1. **Deployment** Agent updates deployment framework
2. **Human** approves deployment framework
3. **Human** deploys changes

1. **Testing Agent** updates feature regression test suites
2. **Human** approves tests
3. **Testing Agent** runs test cases and compares results

1. **Product manager** submits PRD
2. **Architect Agent** creates design
3. **Human** interactively reviews design

1. **Development Agent** implements code
2. **Development Agent** creates Git Pull Request (PR)
3. **Human** interactively reviews Git PR
4. **Human** merges PR

1. **Testing Agent** updates feature regression test suites
2. **Human** approves tests
3. **Testing Agent** runs test cases and compares results

1. **Deployment** Agent updates deployment framework
2. **Human** approves deployment framework
3. **Human** deploys changes

1. **Agent** polls ticket repository for bugs and defects
2. **Agent** performs RCA
3. **Agent** updates tickets for human approval

1. **Product manager** submits PRD
2. **Architect Agent** creates design
3. **Human** interactively reviews design

1. **Development Agent** implements code
2. **Development Agent** creates Git Pull Request (PR)
3. **Human** interactively reviews Git PR
4. **Human** merges PR

1. **Testing Agent** updates feature regression test suites
2. **Human** approves tests
3. **Testing Agent** runs test cases and compares results

1. **Deployment** Agent updates deployment framework
2. **Human** approves deployment framework
3. **Human** deploys changes

1. **Agent** polls ticket repository for bugs and defects
2. **Agent** performs RCA
3. **Agent** updates tickets for human approval

Demo

Prompt Optimization Studio built with CoWeave

CoWeave's Impact
(Our Team's Experience)

- **Operational Excellence**: 40–60% reduction in MTTR, 30–50% fewer deployment failures
- **Engineering Velocity**: 50–60% less time on routine tasks, 40–60% faster onboarding
- **Business Outcomes**: 99.9%+ availability, 60–80% fewer customer incidents

The Future of Engineering Operations

- **Context-aware LLM(s)** handle the heavy lifting lifting using your team's practices and knowledge with human supervision
- **Engineers elevated** to high-value creative and strategic work
- **75-85% reduction** in MTTR: faster resolution, less firefighting
- **Enhanced system reliability** with reduced human effort
- **Accelerated innovation** and **improved job satisfaction** for your Dev, DevOps and SRE teams

The Future of Software Development

Humans + AI, CoWeave Together

CoWeave.ai