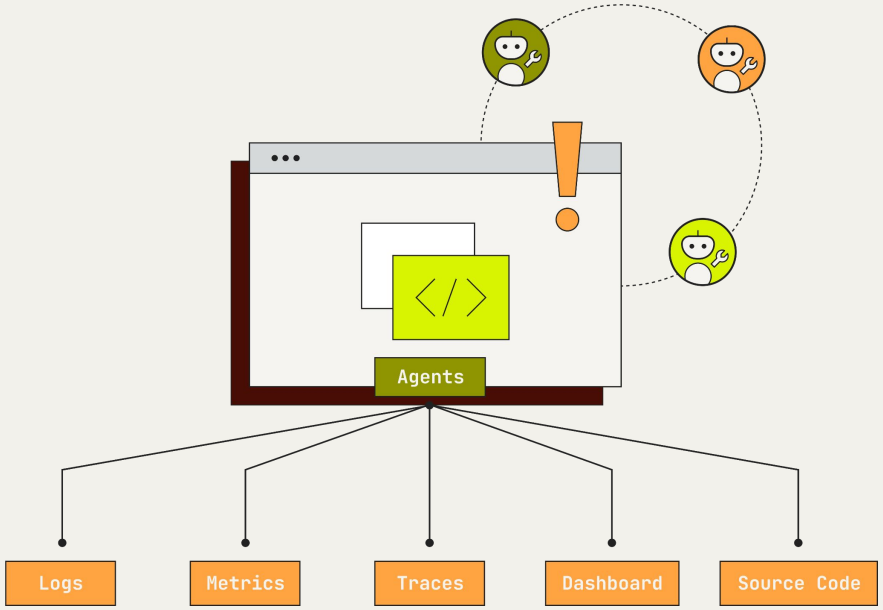
Resolve.ai

Agentic AI for software in production



CS146S Guest Lecture

Mayank Agarwal
Milind Ganjoo

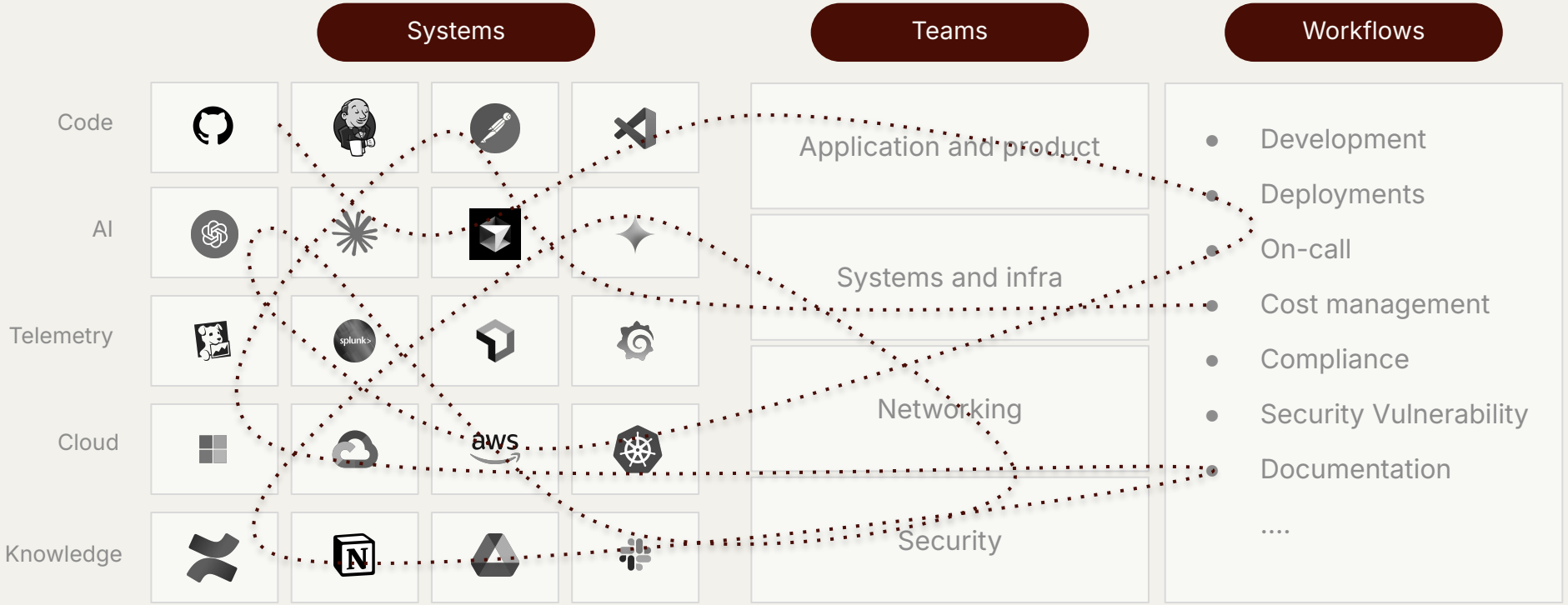
Nov 2025

Mayank Agarwal
Founder & CTO
Resolve AI

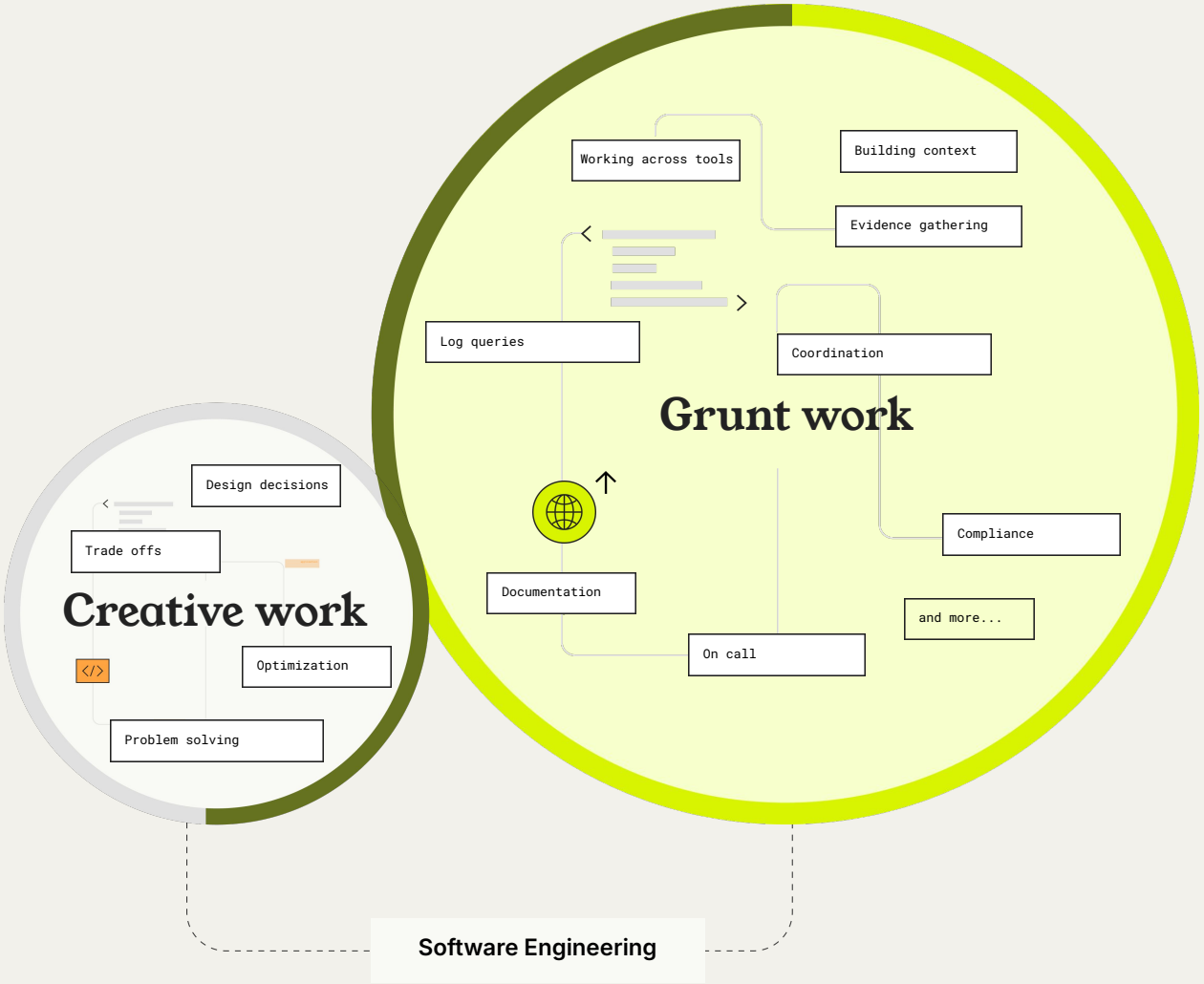
Milind Ganjoo
Member of Technical Staff
Resolve AI

How we imagine software engineering

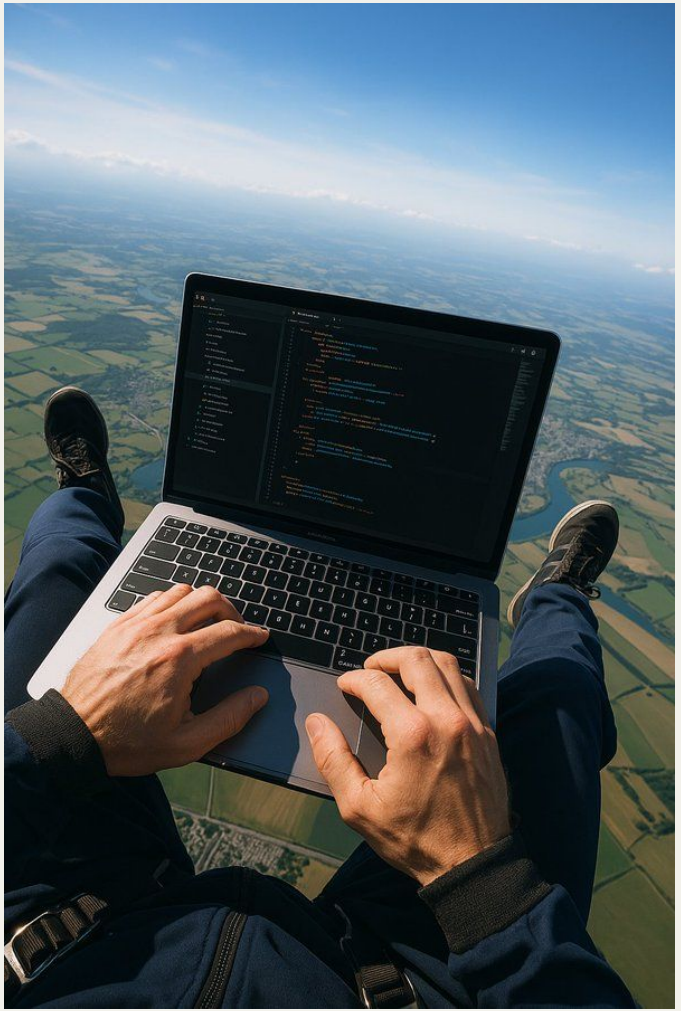
In reality, software engineering is complex and messy



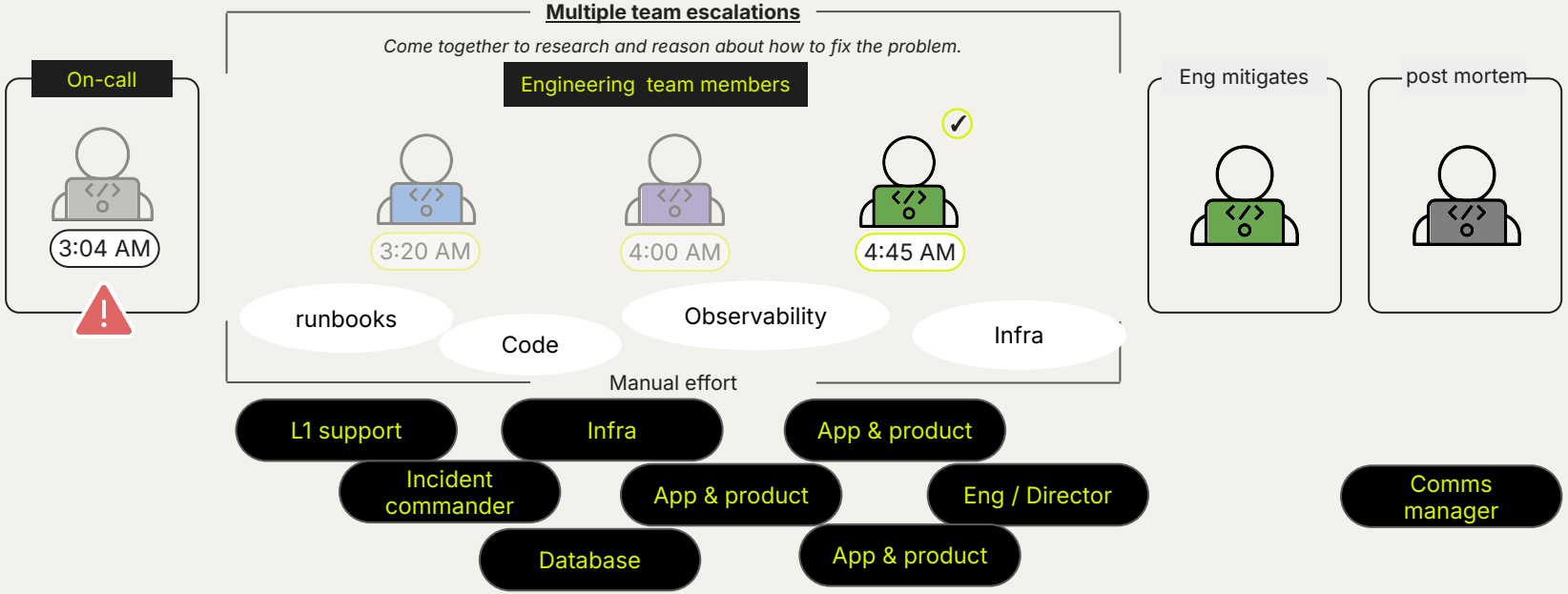
Software Engineers spend 70+% of their time on Grunt work



Life of a software engineer who is on call

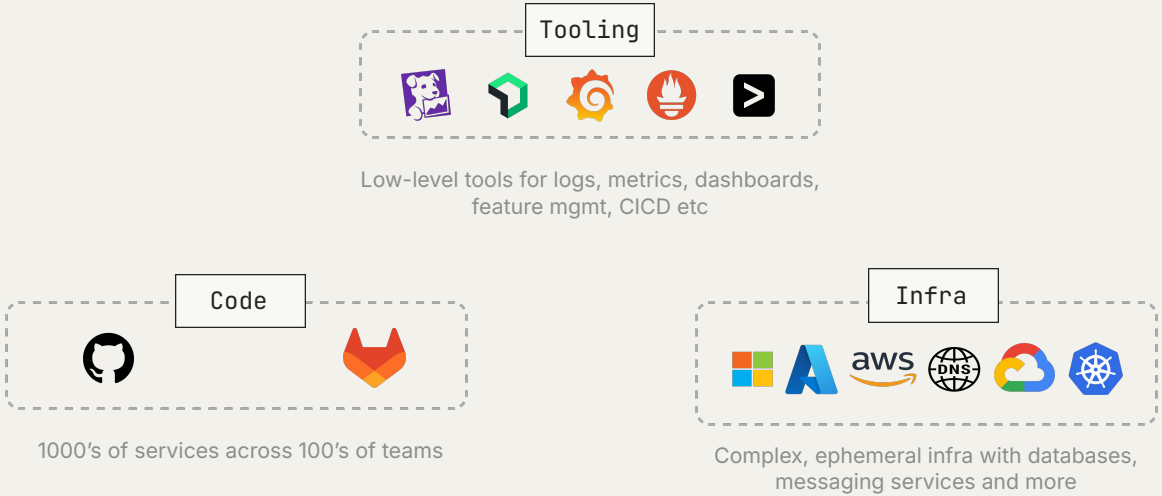


What happens when someone is paged at 3:04 AM?

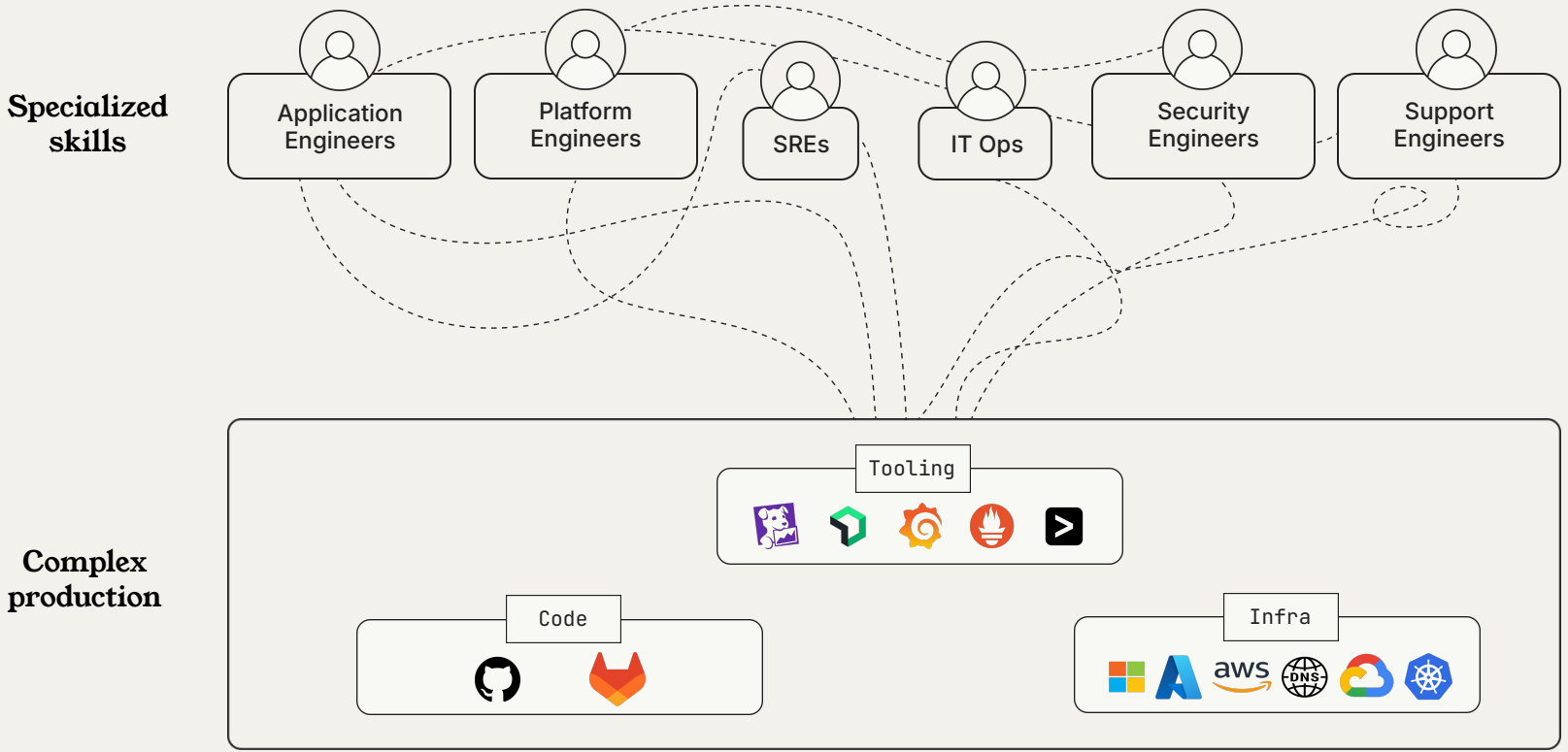


What makes production hard for humans (and models)?

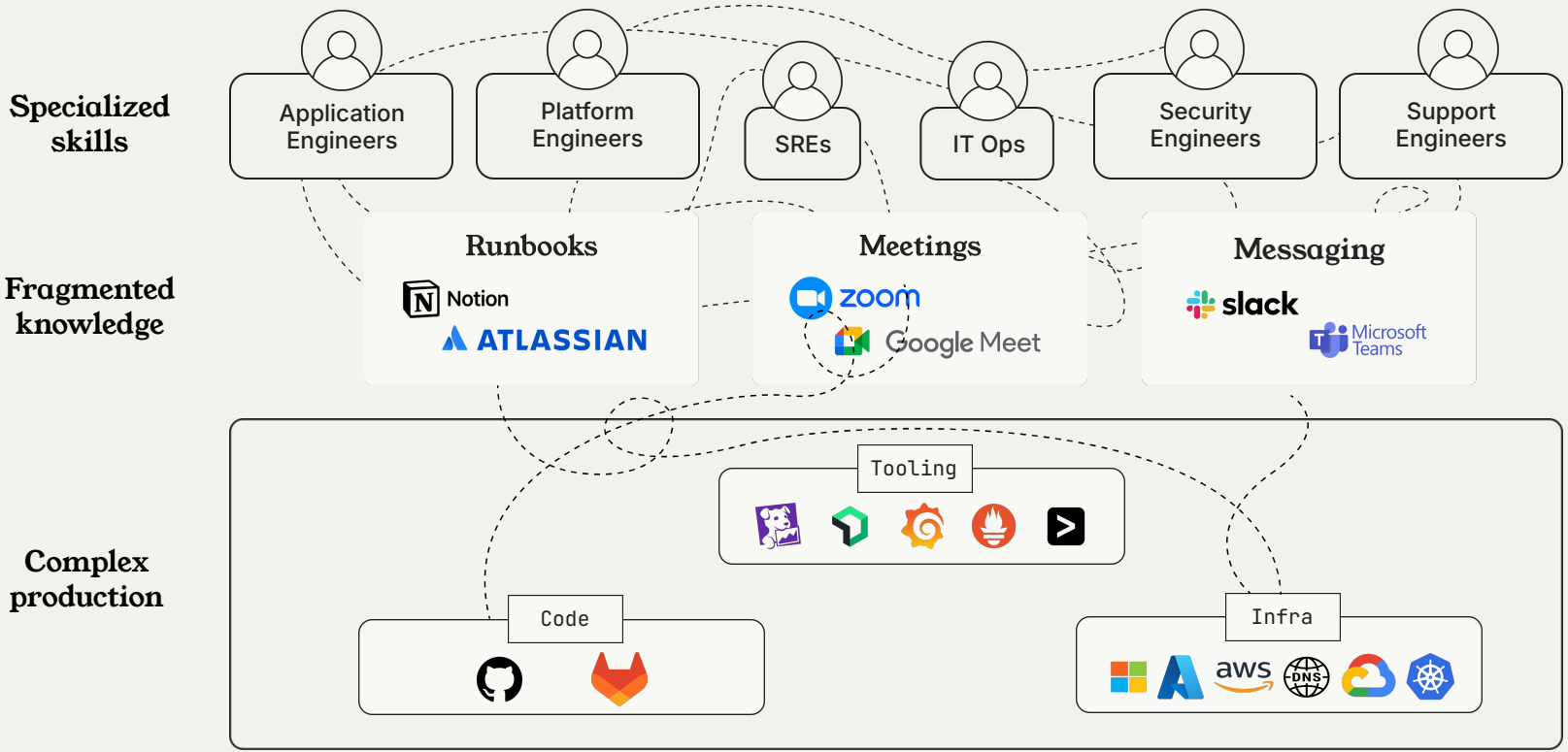
Navigating siloed data across many systems and tools



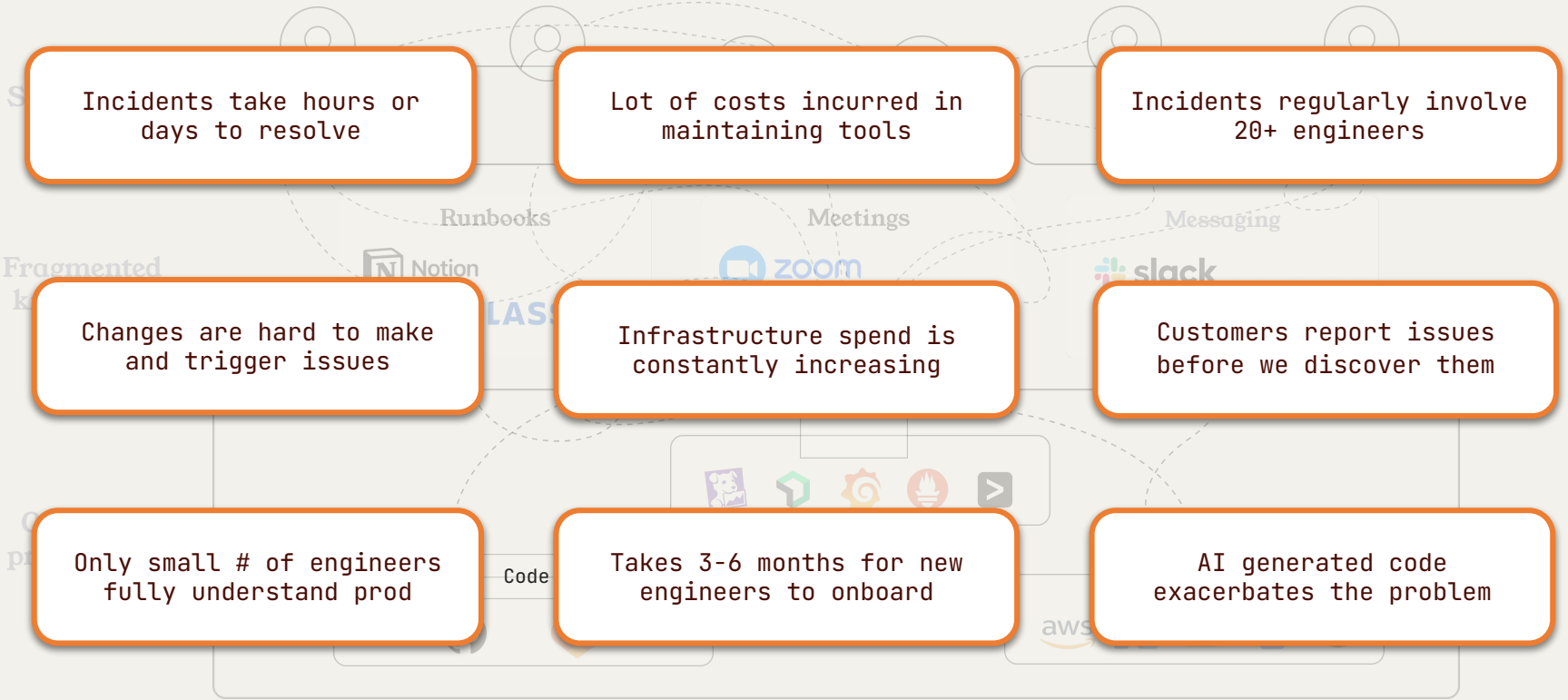
Coordinating across multiple teams with varied expertise



Fragmented and often undocumented context

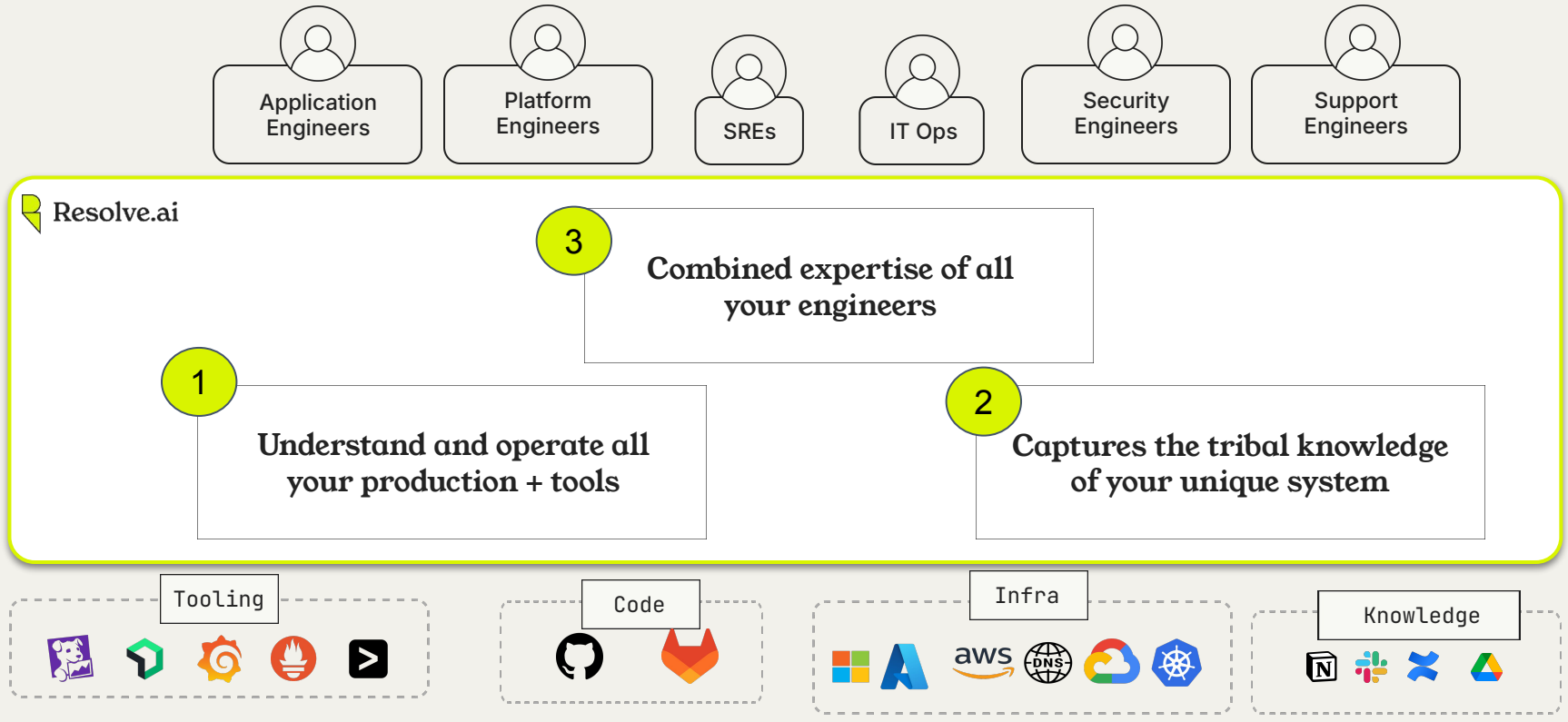


... directly impacts revenue and costs every day



What is needed for AI to help engineers with production systems?

Agent-first approach to work on production systems

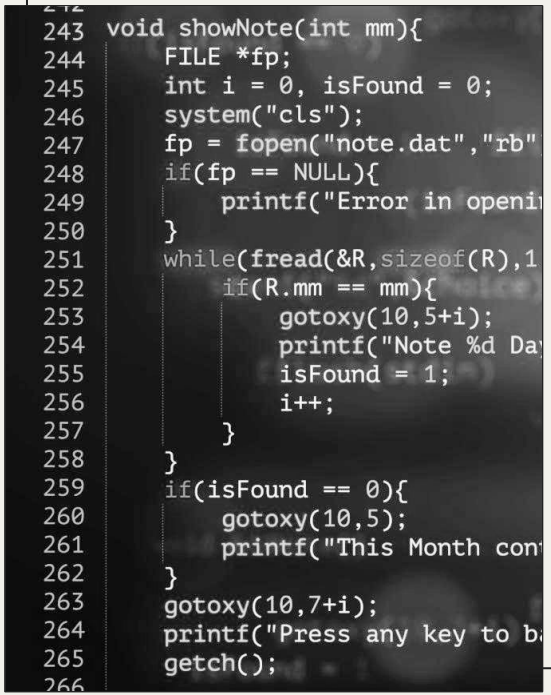


Let us see it in action

How AI for production systems works

01	Production systems are complex and always changing	Understands and operates all your production tools
02	Knowledge is fragmented or undocumented	Captures the tribal knowledge and gets smarter over time
03	Investigations need expertise from multiple teams	Combines expertise of all your engineers

Production systems are complex and always changing

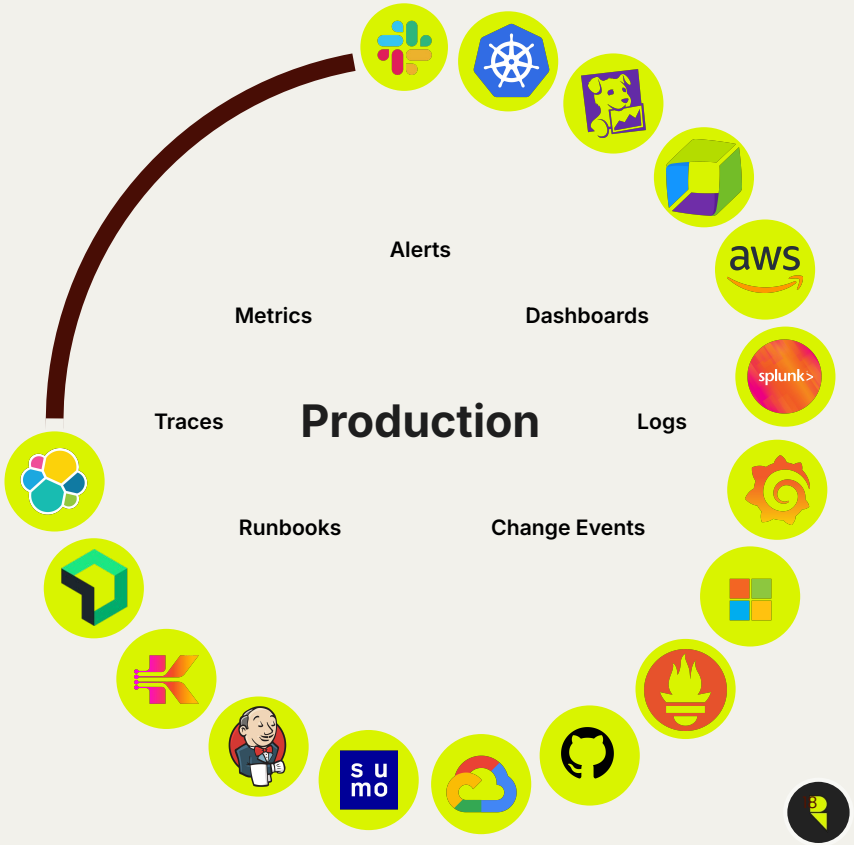


```
void showNote(int mm){
    FILE *fp;
    int i = 0, isFound = 0;
    system("cls");
    fp = fopen("note.dat","rb"
    if(fp == NULL){
        printf("Error in openi
    }
    while(fread(&R,sizeof(R),1
        if(R.mm == mm){
            gotoxy(10,5+i);
            printf("Note %d Da
            isFound = 1;
            i++;
        }
    }
    if(isFound == 0){
        gotoxy(10,5);
        printf("This Month con
    }
    gotoxy(10,7+i);
    printf("Press any key to b
    getch();
```

- Hundreds of tools: different query languages, access mechanisms, and operational behaviors for each
- Should map across code <> infra <> telemetry and identify complex dependencies
- *Massive* scale (millions of logs lines, thousands of metrics, dozens of platforms, etc.)

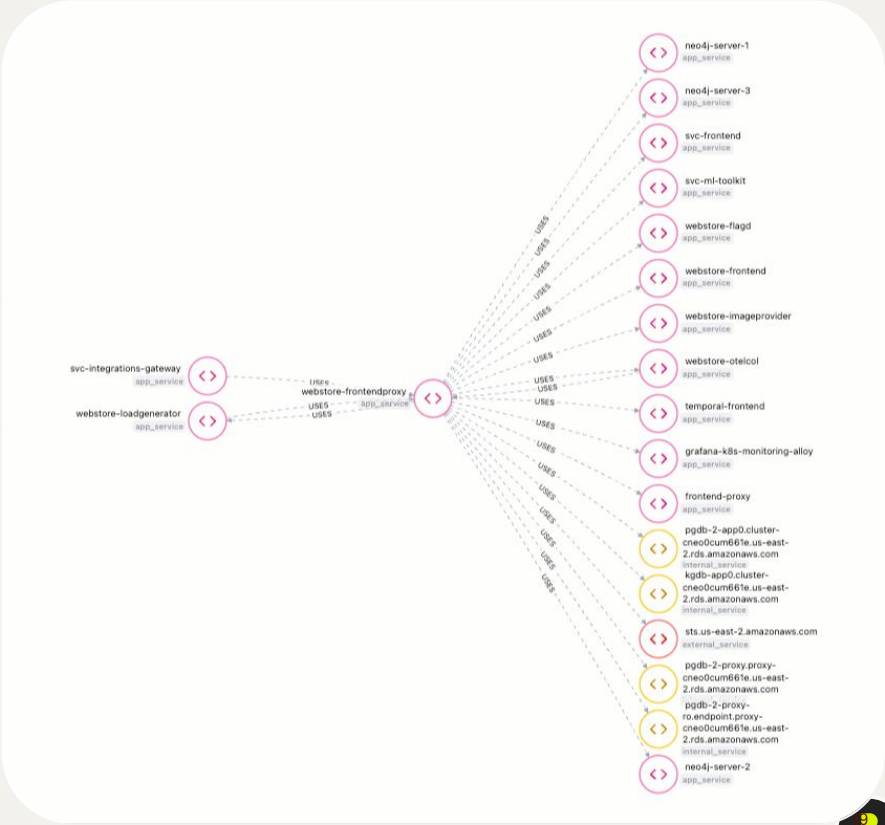
AI systems should deeply understand your production

1. Connects to code, infra, tools, and knowledge



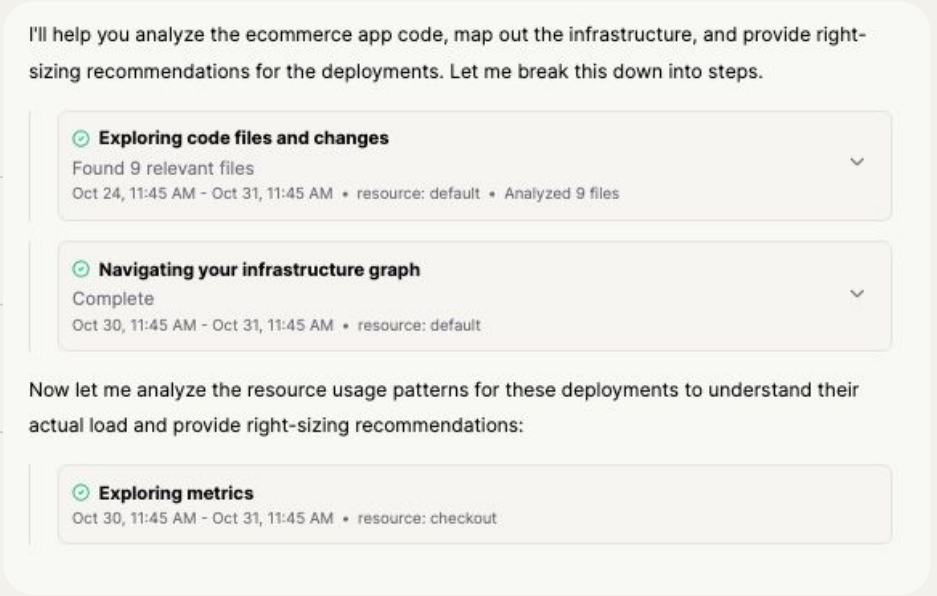
AI systems should deeply understand your production

1. Connects to code, infra, tools, and knowledge
2. Models how your systems works



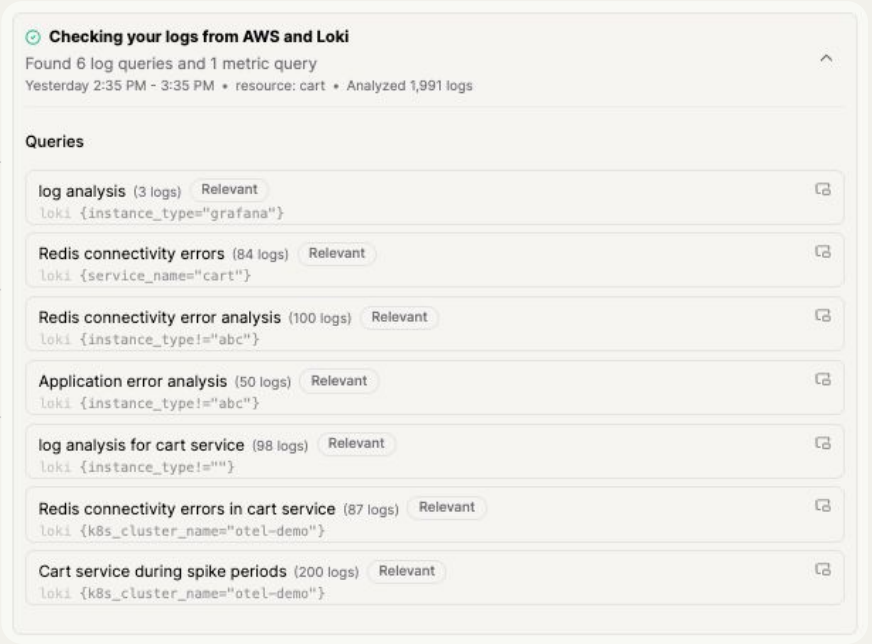
AI systems should deeply understand your production

1. Connects to code, infra, tools, and knowledge
2. Models how your systems works
3. Navigates to the right nodes in the graph to gather evidence

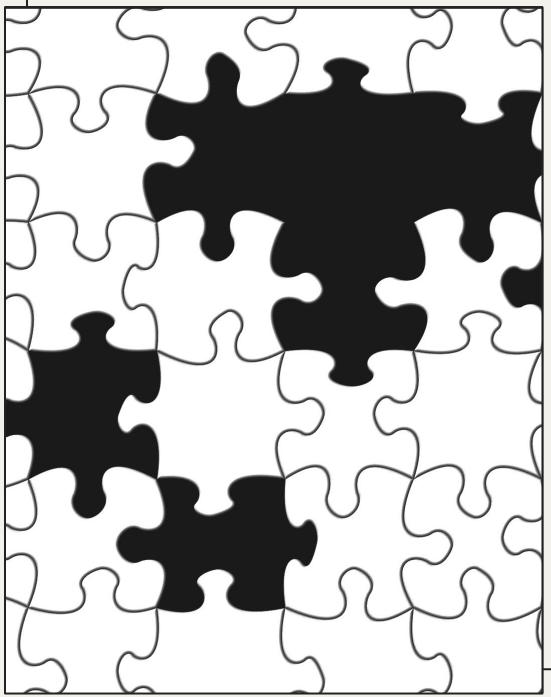


AI systems should deeply understand your production

1. Connects to code, infra, tools, and knowledge
2. Models how your systems works
3. Navigates to the right nodes in the graph to gather evidence
4. Operates every tool or system like experts



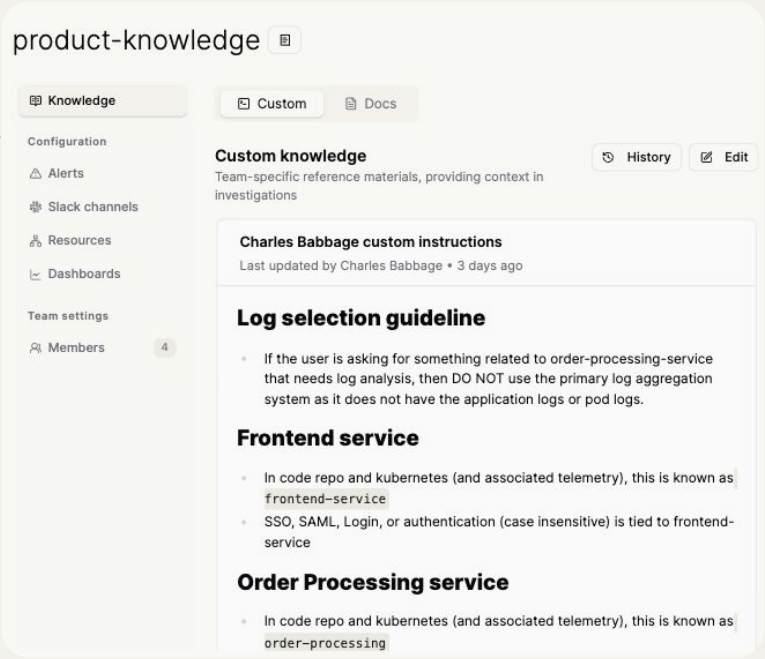
Knowledge is fragmented or undocumented



- Knowledge is scattered across different runbooks, documentation, chats (if you're lucky! Might be completely undocumented otherwise)
- In-the-loop feedback or learnings are often undocumented or lost
- New investigations need to adapt continuously and require learnings from previous iterations

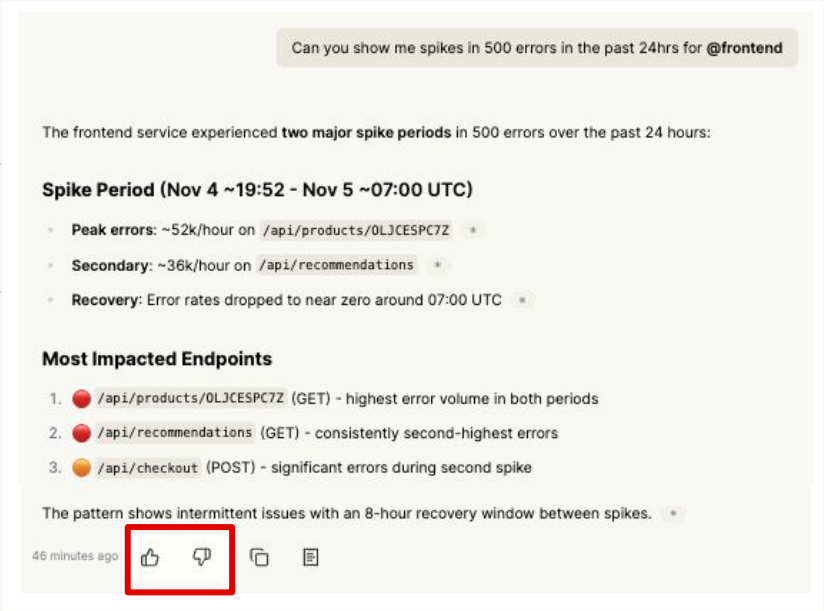
AI systems should capture tribal knowledge and get smarter with every interaction

1. Captures company and team-wide knowledge



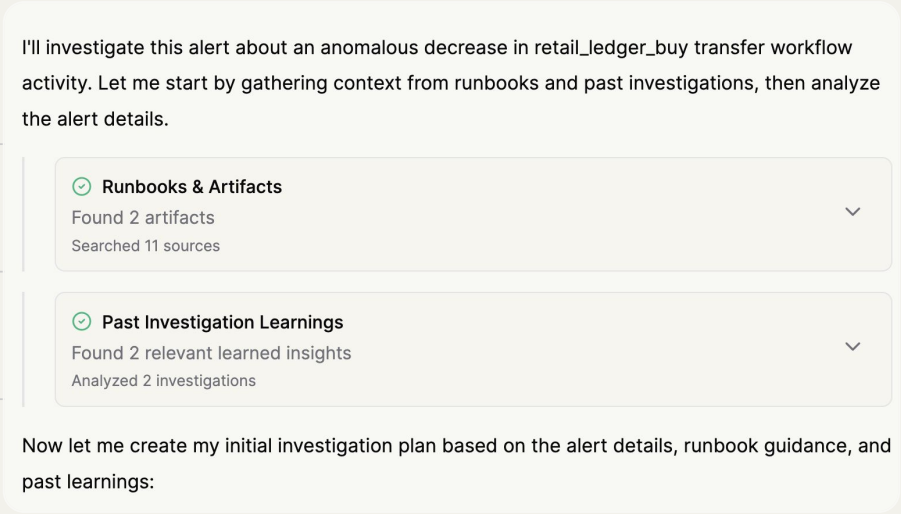
AI systems should capture tribal knowledge and get smarter with every interaction

1. Captures company and team-wide knowledge
2. Remembers In-the-loop feedback/teachings

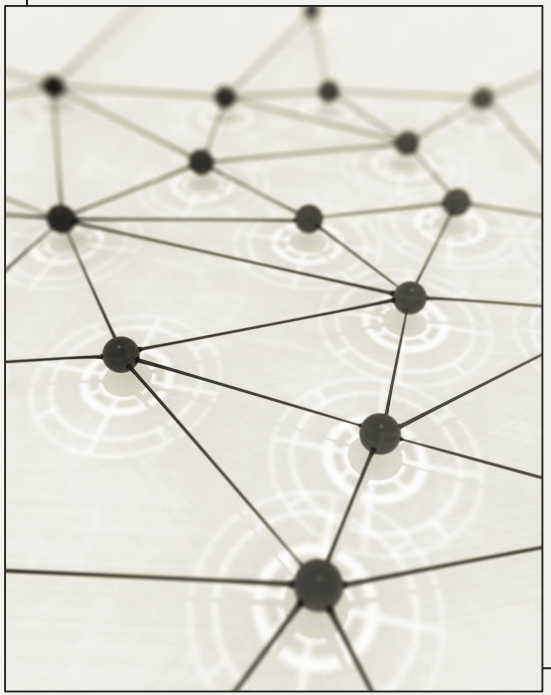


AI systems should capture tribal knowledge and get smarter with every interaction

1. Captures company and team-wide knowledge
2. Remembers In-the-loop feedback/teachings
3. Retrieves context specific information



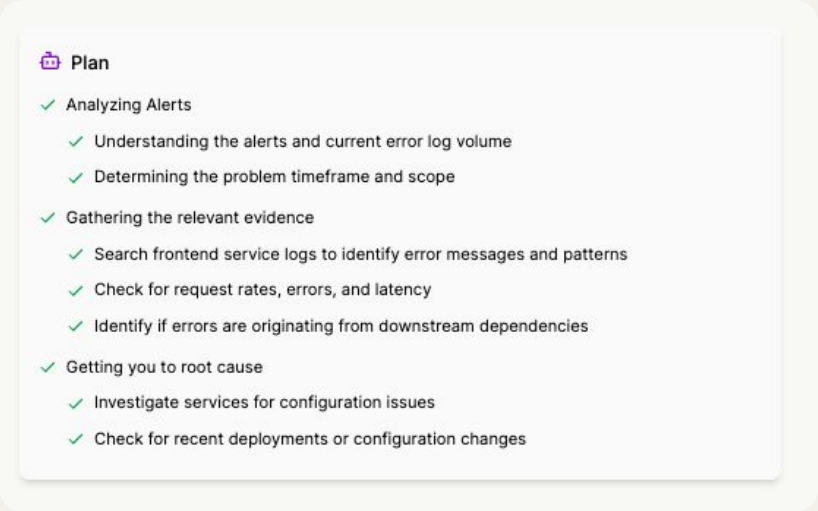
Investigating production needs expertise from multiple teams



- Triaging an incident is hard. Especially for novel incidents or new engineers
- Sequential investigations take a lot of time if not on the right path
- Coordination across teams adds time or loss of information in hand-offs
- Organizational and expertise boundaries make it hard to gather context

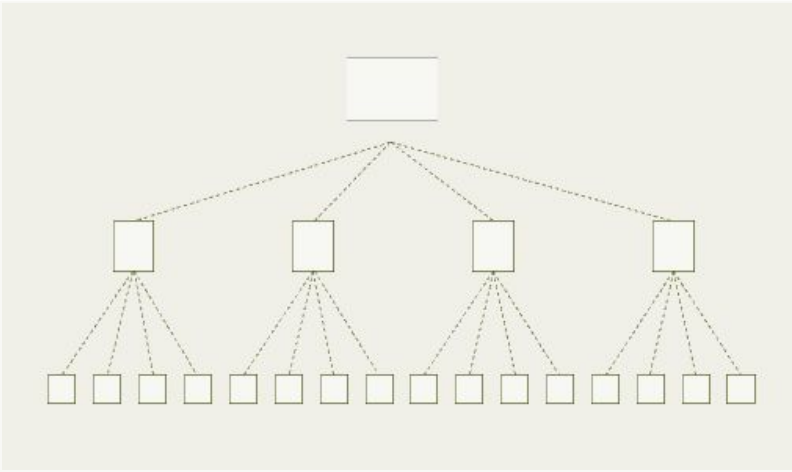
AI systems should combine expertise of all engineers across teams

1. Creates an investigation plan



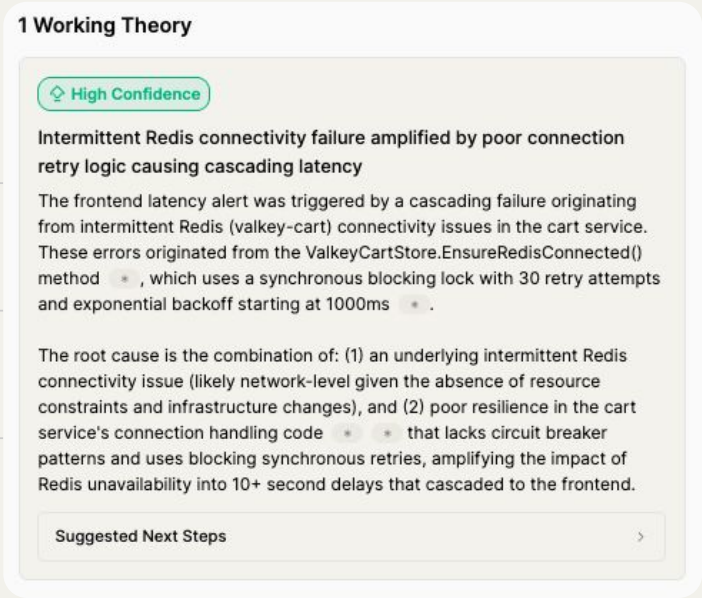
AI systems should combine expertise of all engineers across teams

1. Creates an investigation plan
2. Pursues multiple hypotheses in parallel



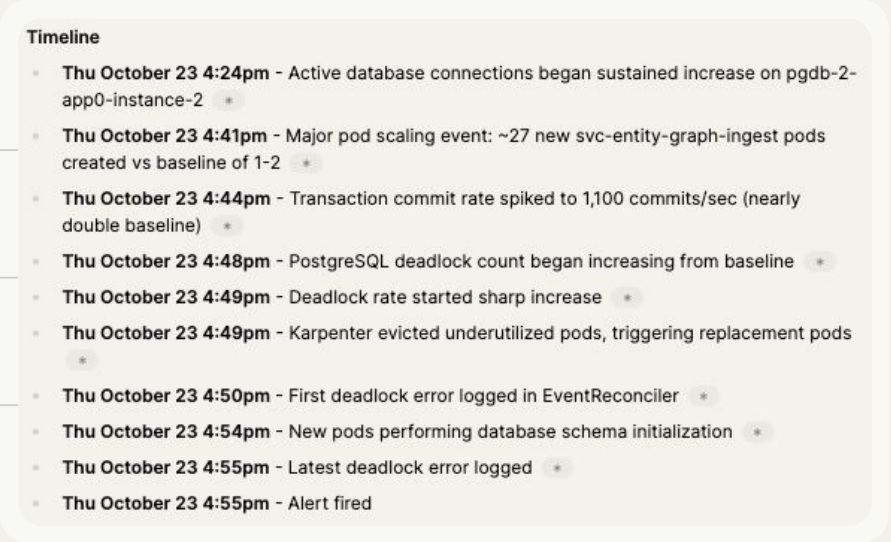
AI systems should combine expertise of all engineers across teams

1. Creates an investigation plan
2. Pursues multiple hypotheses in parallel
3. Refines plan continuously until you get to root cause



AI systems should combine expertise of all engineers across teams

1. Creates an investigation plan
2. Pursues multiple hypotheses in parallel
3. Refines plan continuously until you get to the root cause
4. Enables multi user collaboration across org boundaries to get to the right answer every time



How AI for production systems works

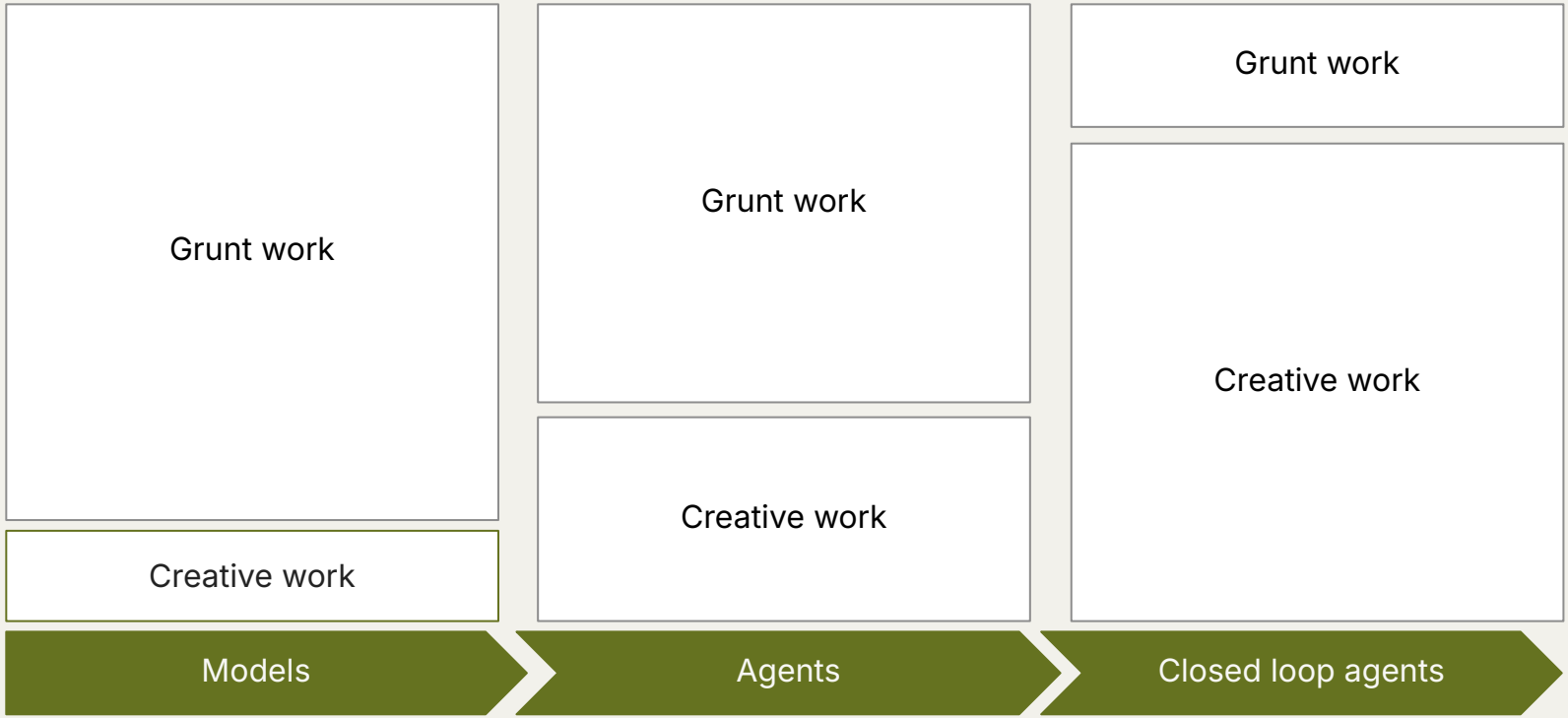
01	Production systems are complex and always changing	Understands and operates all your production tools
02	Knowledge is fragmented or undocumented	Captures the tribal knowledge and gets smarter over time
03	Investigations need expertise from multiple teams	Combines expertise of all your engineers

Lessons learned building AI for prod

- **This isn't just a model problem**
  - Navigating production requires a lot of domain expertise which is coded into the architecture. You can’t just prompt-engineer models to build production AI
- **Context windows are limited and production context is infinite**
  - Can't fit 10M logs in any context window. Intelligence is knowing WHAT to query, WHEN, and HOW to filter based on production understanding.
- **Working with tools is a non-trivial problem**
  - Raw APIs are unusable: large responses, outputs are messy, and meant for humans. Must build AI systems that can filter noise, return structured summaries, handle errors gracefully, work in parallel
- **Evals are as hard as the product itself**
  - Building evals requires replicating production complexity - (services, dependencies, etc.,). Without evals, you can't trust the outputs.

AI is changing software engineering

By next year software engineering will look fundamentally different



Resolve.ai

Thank You

We're hiring!

linkedin.com/in/resolveai

@resolveai

Q&A

LET'S DISCUSS