

Simple Patterns for More Reliable Agents

Wil McKenney | SWE @ Capital One | Accelerate Chicago 2026

Why Agents Fail on Complex Work

Derailing the train of thought



Context Exhaustion

Complex projects span too many files and systems for one context window to hold



No Memory Between Sessions

Starting a new chat means the agent forgets everything from the previous one



No Feedback Loop

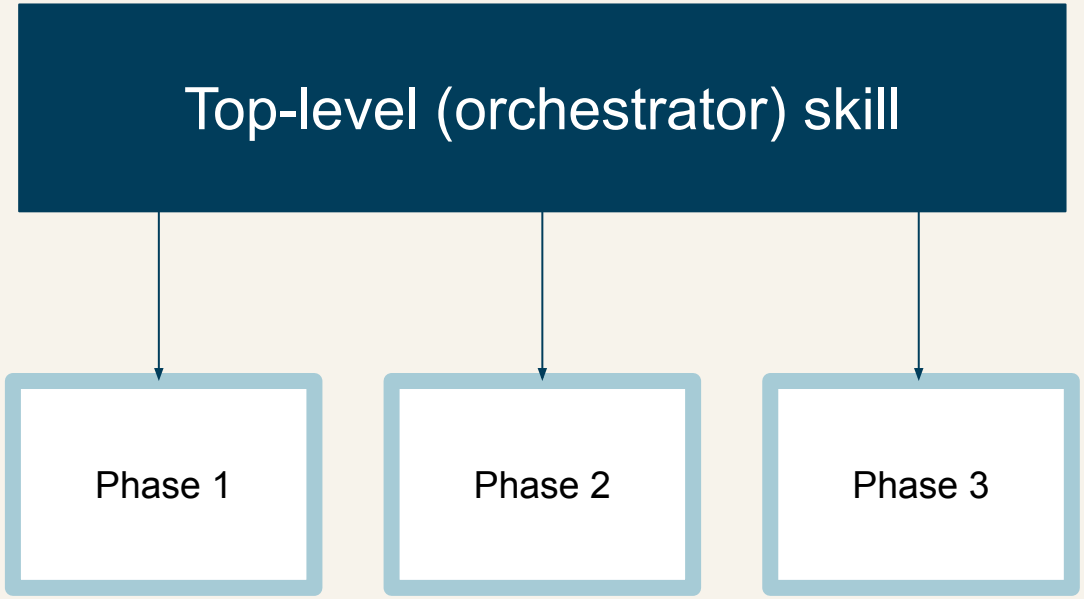
Agents can’t learn from build failures without explicit structure

Phased Approach

Skills-based orchestration for complex tasks

https://agentskills.io/home

Open standard by Anthropic




```
{some skill}/
    SKILL.md
    some_context.md
    template.md
    ...
```

Modular

Each skill can be modified independently without affecting others in the pipeline

Scalable

Add a new phase by dropping in a new skill without changing existing workflow

Self-Contained

Each skill carries its own context, templates, and docs to keep the context window clean

Persisting State for Idempotency

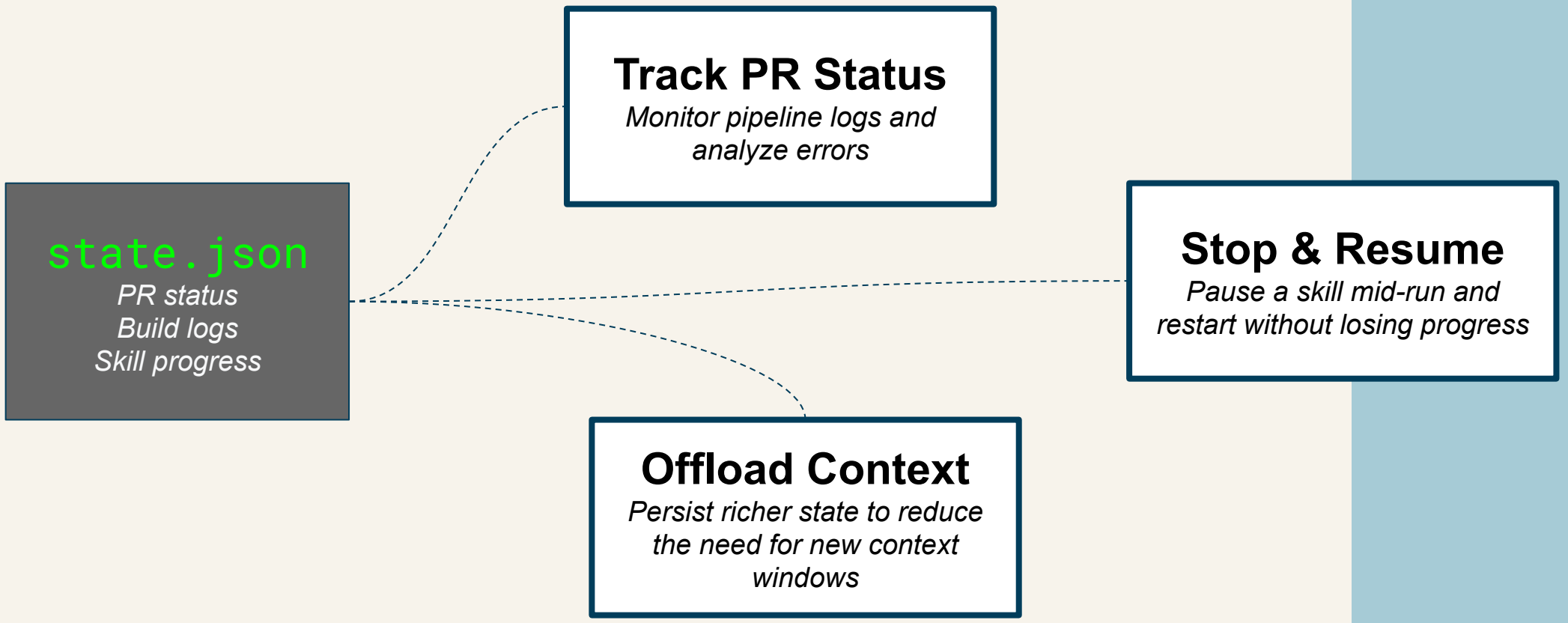
Track resource status across context windows

resource	status	action	notes
resource-a	completed	NONE	Created in Phase 1
resource-b	needs_action	CREATE	Pending - Phase 2
resource-c	unused	NONE	Not used in target
…	…	…	



Future State Driven Self Correction

Expanding state to enable resilience and richer feedback loops



Putting It Together

Three patterns that compose

1

Structure

Phased Orchestration

Break complex work into modular, self-contained skill-based phases

2

Persist

Saved State File

Track progress in a file so each phase runs idempotently

3

Correct

Feedback Loops

Use build logs and PR status to manage self-correction

Your Agent Doesn’t Need a Bigger Context Window

It Needs ~~a State File~~ ***Structure***

- Organize complex work into phased, modular skills
- Persist state to a file for idempotent, resumable execution
- Check build logs to close the feedback loop