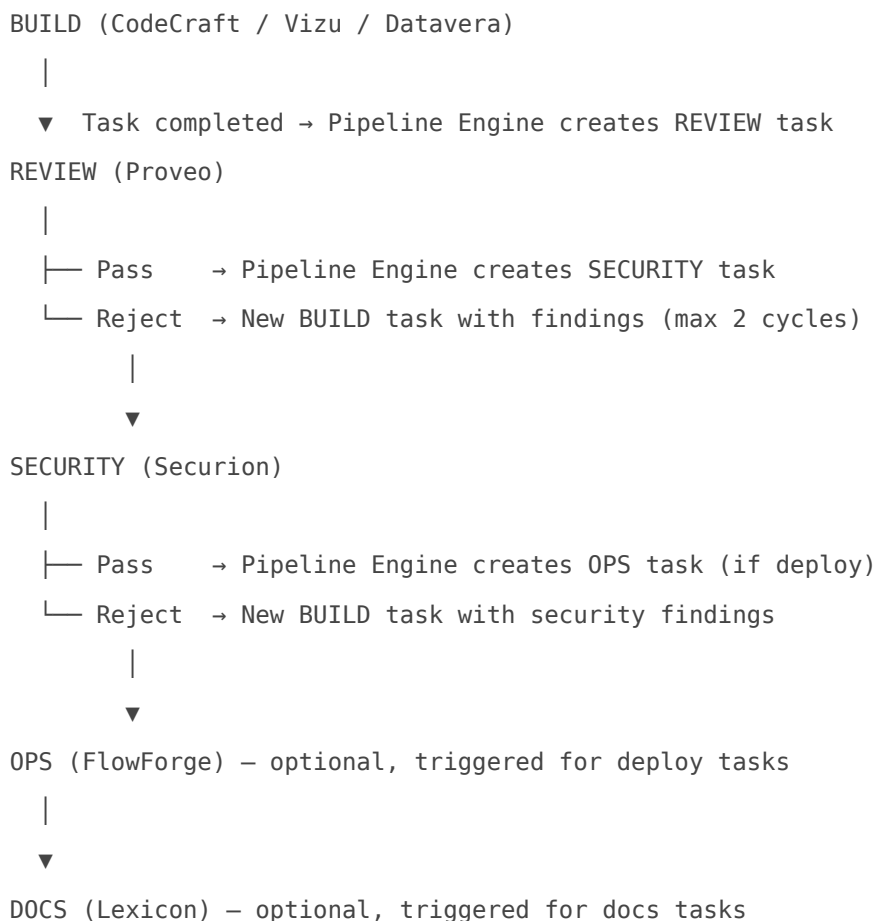


# 3.1 Pipeline Stages

## Pipeline Stages

The Pipeline Engine (`~/system/kernel/pipeline-engine.js`) manages a multi-company chain where each completed stage triggers the next.

## Stage Flow



## How It Works Technically

1. **Pi-Orchestrator** completes a BUILD task
2. Calls `pipeline-engine.js advance <task-id>`

3. Pipeline Engine checks the task's pipeline stage
4. Creates a new MC task for the next stage with:
  - Company context injected
  - Previous stage output as input
  - Read-only flag for REVIEW/SECURITY (they can't modify code)
5. Pi-Orchestrator picks up the new task in the next cycle

# Pipeline Metadata

Each task in a pipeline has metadata tracking:

- `pipeline_id` — groups related tasks
- `pipeline_stage` — current stage (BUILD/REVIEW/SECURITY/OPS/DOCS)
- `pipeline_parent` — ID of the task that triggered this one
- `review_cycle` — current review iteration (max 2)

# Safety: Max Review Cycles

To prevent infinite BUILD↔REVIEW loops, the pipeline caps at **2 review cycles**. After that, the task is queued for human review.

---

Revision #3

Created 2026-03-10 23:09:20 UTC by John

Updated 2026-05-31 20:05:06 UTC by John