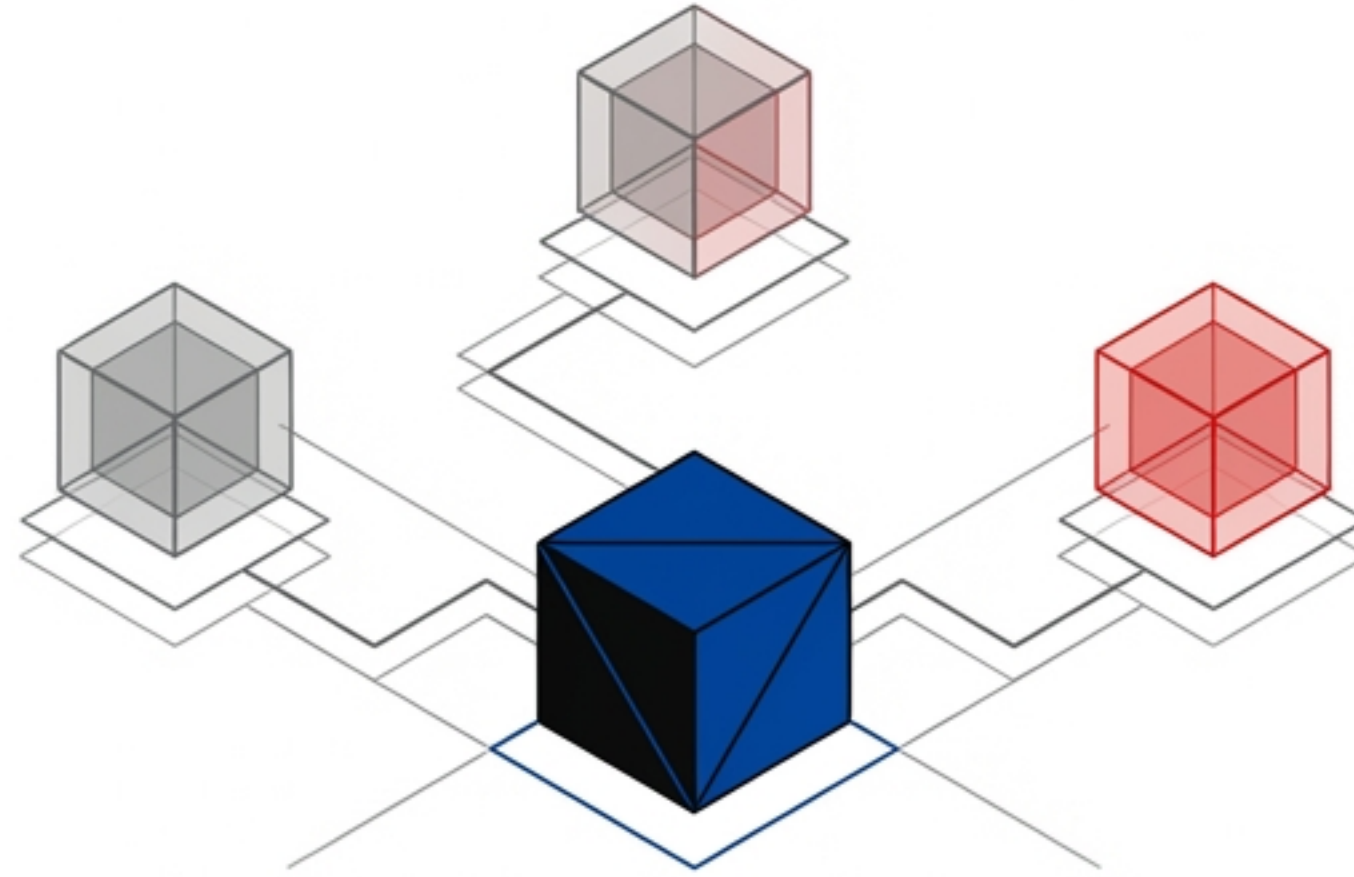


# CC-ADVANCECYBER



Enterprise Orchestration for Claude Code

VERSION 1.0.0 | BUSINESS REQUIREMENTS OVERVIEW  
MIT LICENSE | STATUS: ACTIVE | DATE: 2026-01-31

# Executive Summary: The Enterprise Shift

CC-ADVANCECYBER transforms Claude Code from a single-instance CLI tool into a multi-variant enterprise platform. It provides a complete software development orchestration system without modifying the core installation.

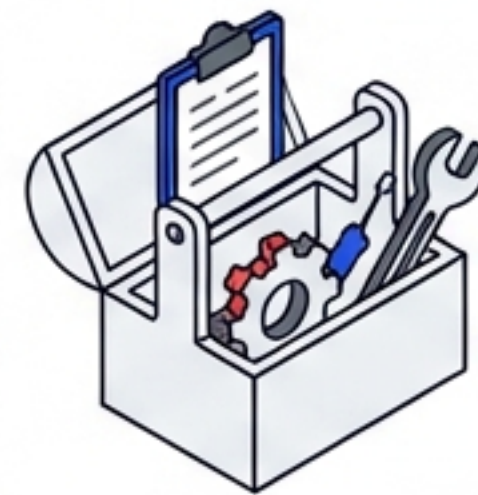
**Isolated. Configurable. Intelligent.**

## KEY CAPABILITIES



### The Workforce

55 Specialized Agents ranging from Architecture to Security.



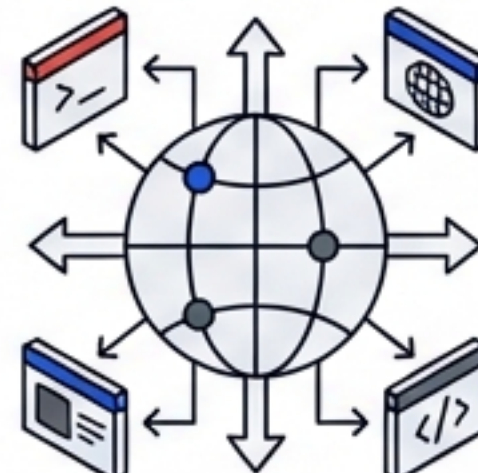
### The Toolkit

38 Production Skills including Document Gen & n8n Automation.



### The Brain

Persistent Memory via Qdrant & PostgreSQL.



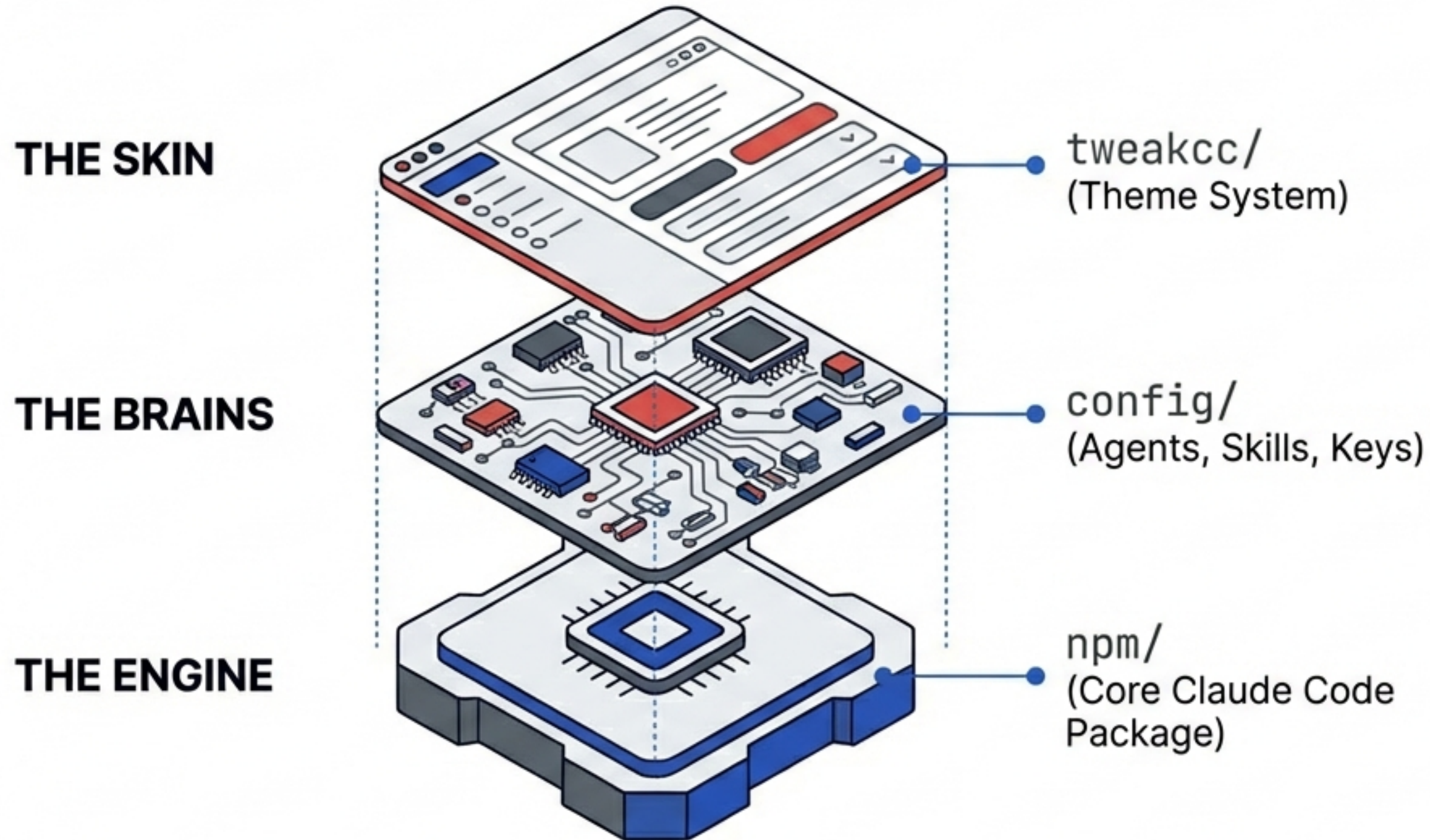
### Connectivity

Multi-Provider support (Enterprise, OpenRouter, Local).



# High-Level System Architecture

## The Overlay Philosophy



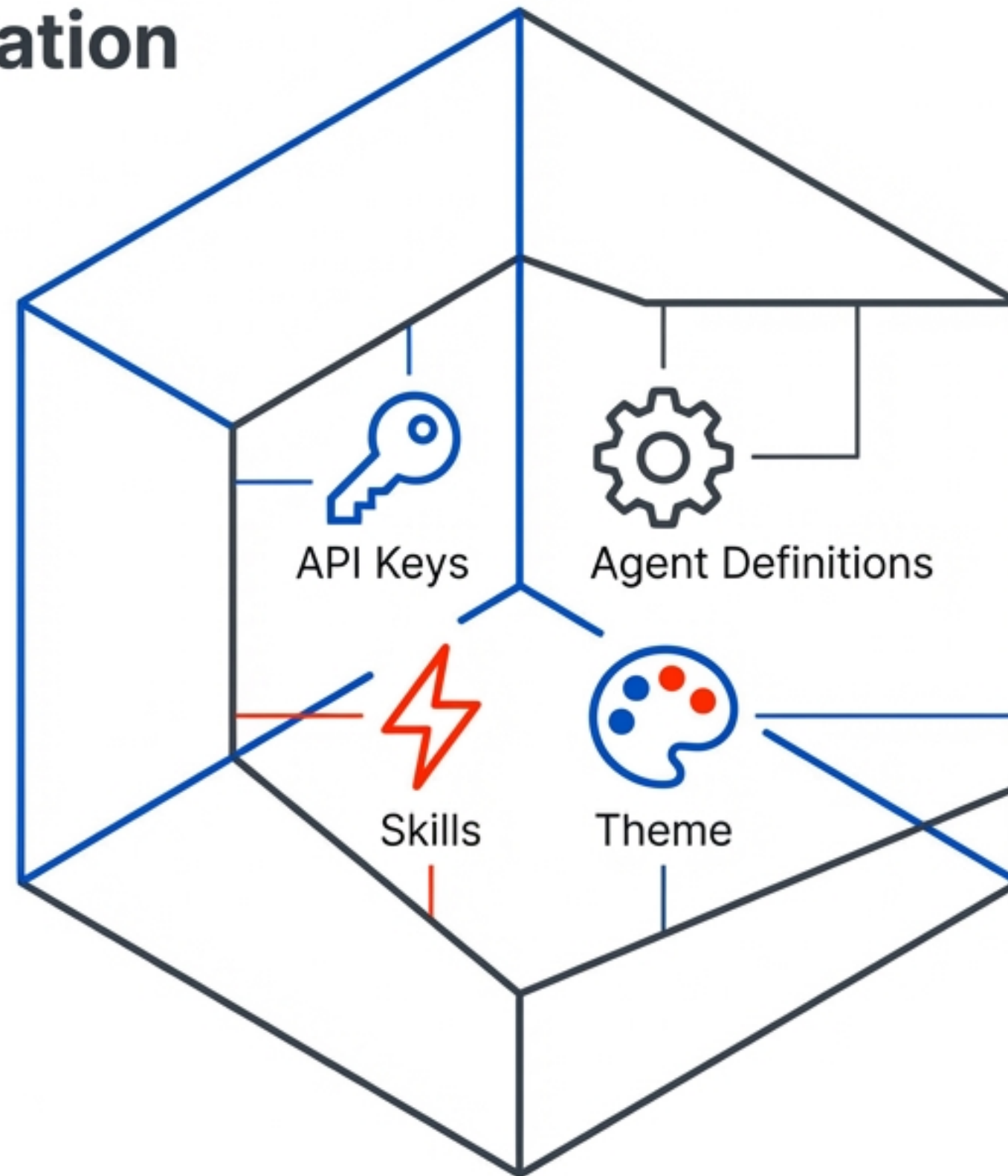
The system is designed as an overlay. It manages the environment, memory, and personality, allowing the core engine to focus purely on code generation.



# The Variant System

## Isolation & Configuration

Technical Swiss  
Laboratory



### Definition:

A Variant is a complete, isolated Claude Code installation environment. This allows users to run a strict “SecOps” environment alongside a loose “Creative” environment with zero cross-contamination.

```
// distinct configs  
stored in variant.json
```



# Provider Flexibility

	Provider	Description	Auth	Gateway
1	Mirror (Enterprise)	Enterprise Claude environment	OAuth (None)	Default
2	OpenRouter	Access to 100+ models	Token	<a href="https://openrouter.ai/api">openrouter.ai/api</a>
3	CCRouter	Local LLM Integration	Token	<a href="http://127.0.0.1:3456">127.0.0.1:3456</a>
4	Custom	Bring Your Own Endpoint	Configurable	User Defined

## Model Mapping

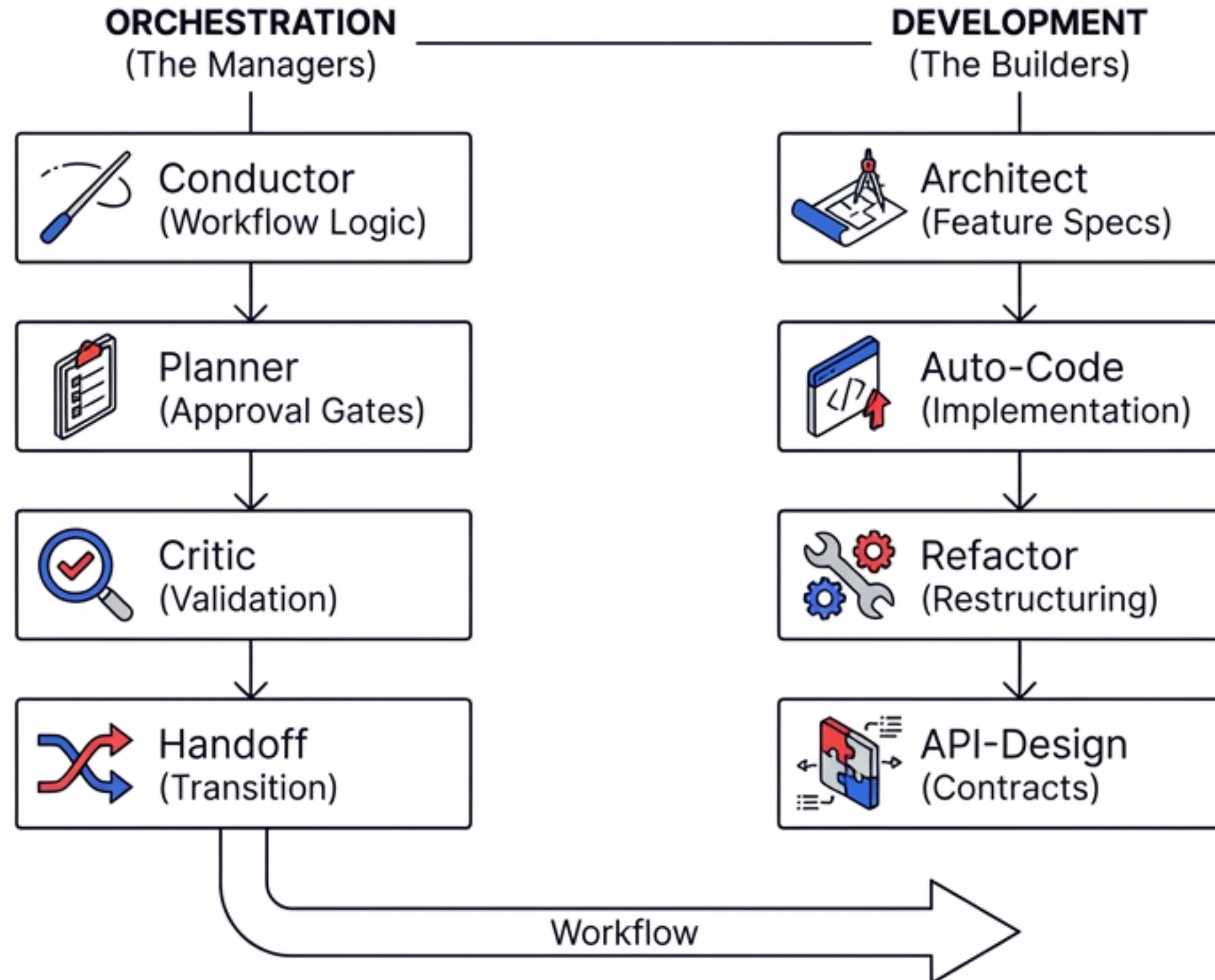
Generic requests are mapped to providers via Environment Variables.

```
ANTHROPIC_DEFAULT_SONNET_MODEL = 'provider-specific-id'
ANTHROPIC_DEFAULT_OPUS_MODEL = 'provider-specific-id'
```

# The Workforce: Part 1

Technical Swiss Laboratory

## Orchestration & Development





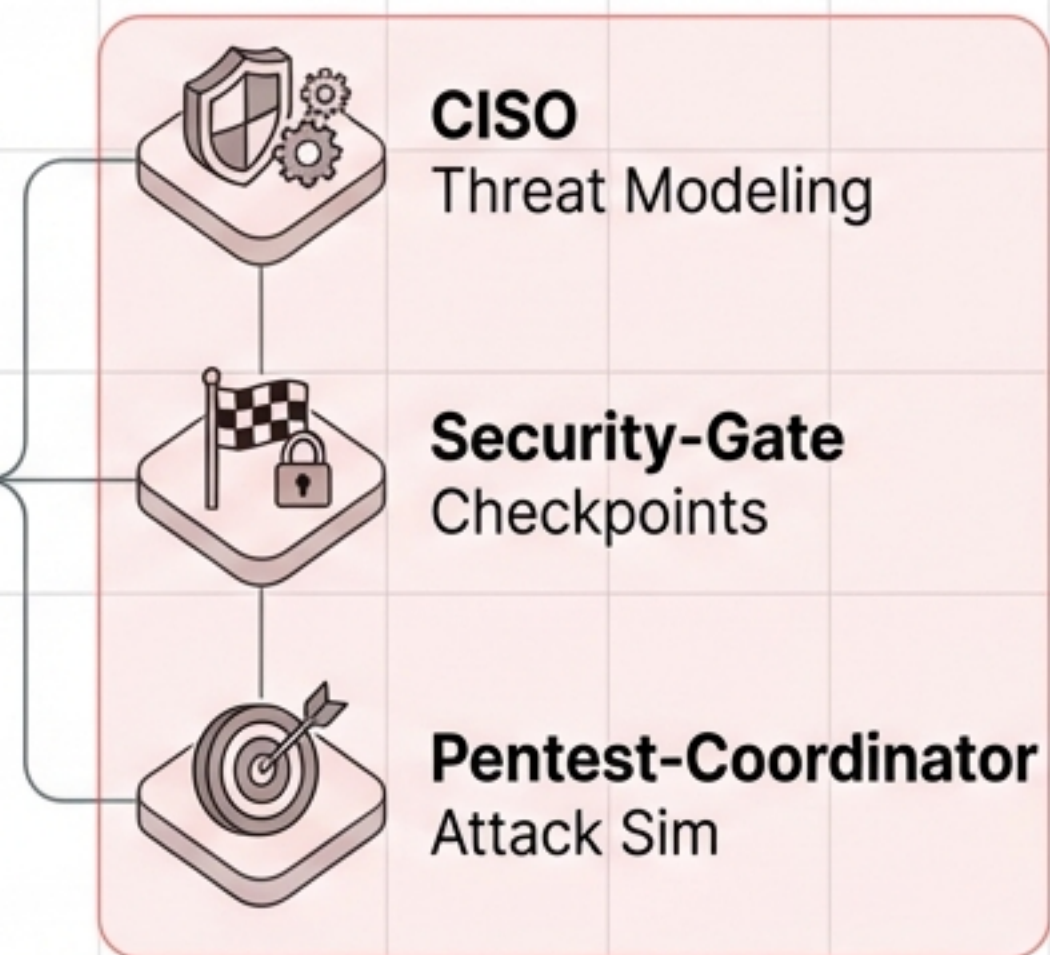
# The Workforce: Part 2

Technical Swiss Laboratory

## Quality, Security & Ops

Inter, Technical Graphite

### SECURITY (The Guardrails) JetBrains Mono



### QUALITY & OPS (The Delivery)

Deep Obsidian, JetBrains Mono



**QA**  
Gap Analysis



**Compliance**  
Regulatory



**Supply-Chain-Security**  
SLSA



**Observability**  
Alerting

**Key Insight:** Security is not an afterthought configuration; it is an active agentic role within the loop.

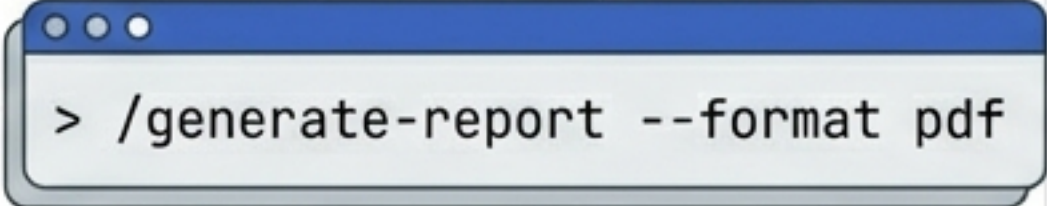


# The Toolkit: 38 Production Skills

Technical Swiss Laboratory

Invokable via /skill-name

Inter, Technical Graphite



```
> /generate-report --format pdf
```

## Document Generation

- PDF Creation
- PPTX Slides
- XLSX Data
- DOCX Reports

## Frontend

- UI Design
- Artifact Generation
- React Components

## Automation

- n8n Workflow Integration
- Webhooks
- Triggers

## Research & Utility

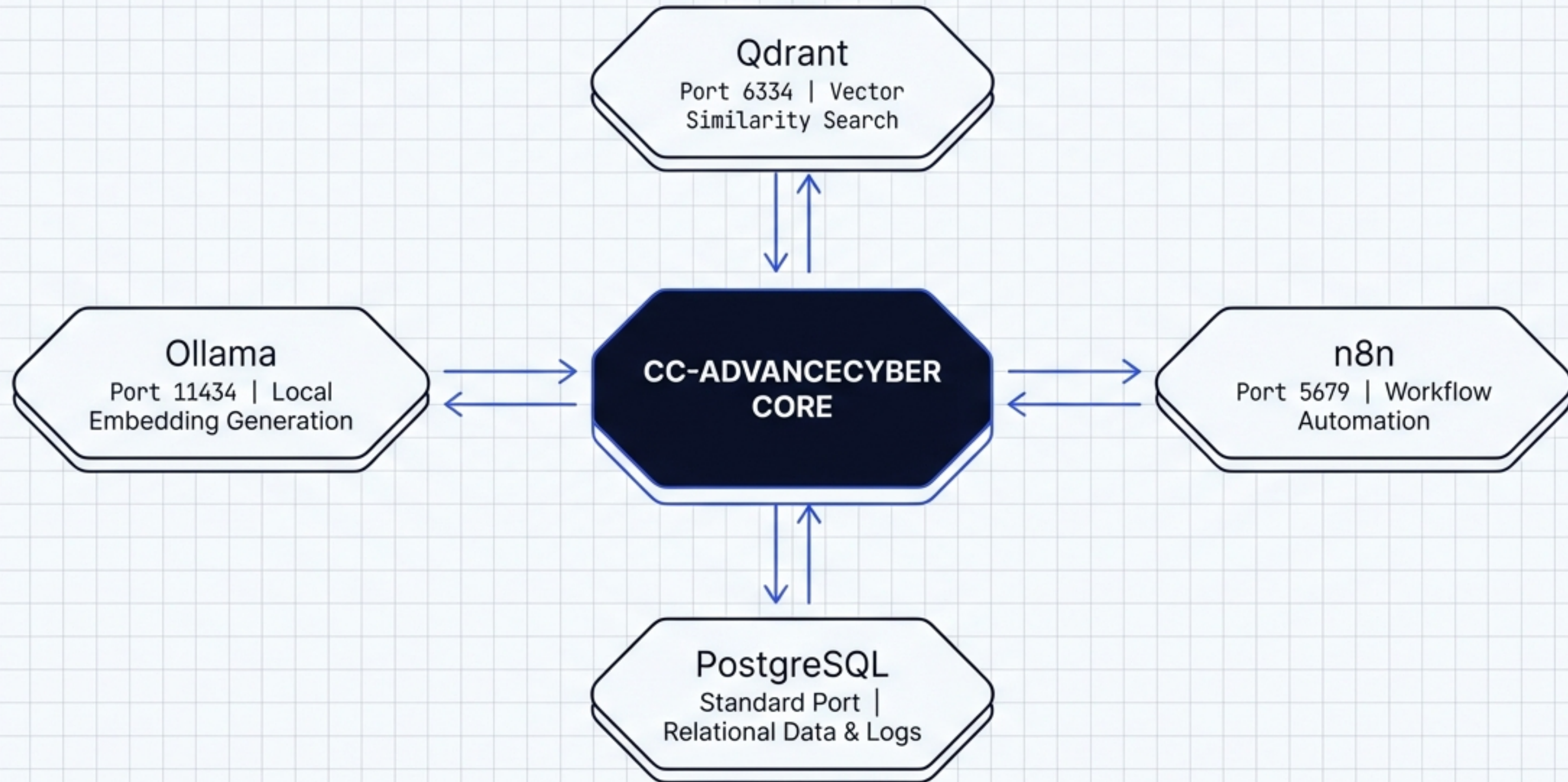
- Data Synthesis
- Content Creation
- Brand Management



# The Brain: Persistent Memory Architecture

Technical Swiss Laboratory

Long-term Storage + Vector Search





# Memory Tools & Capabilities

## in Deep Obsidian

Technical Swiss Laboratory

### INPUT & STORAGE

- `memory_store`  
Embeddings
- `episode`  
Task Recording
- `learning`  
Procedure Patterns

### OPTIMIZATION

- `memory_promote`  
Working to Long-term
- `memory_summarize`  
Consolidate
- `memory_organize`  
Cluster & Prune

### RETRIEVAL

- `memory_recall`  
Semantic Search
- `rag_search`  
External Obsidian Vaults

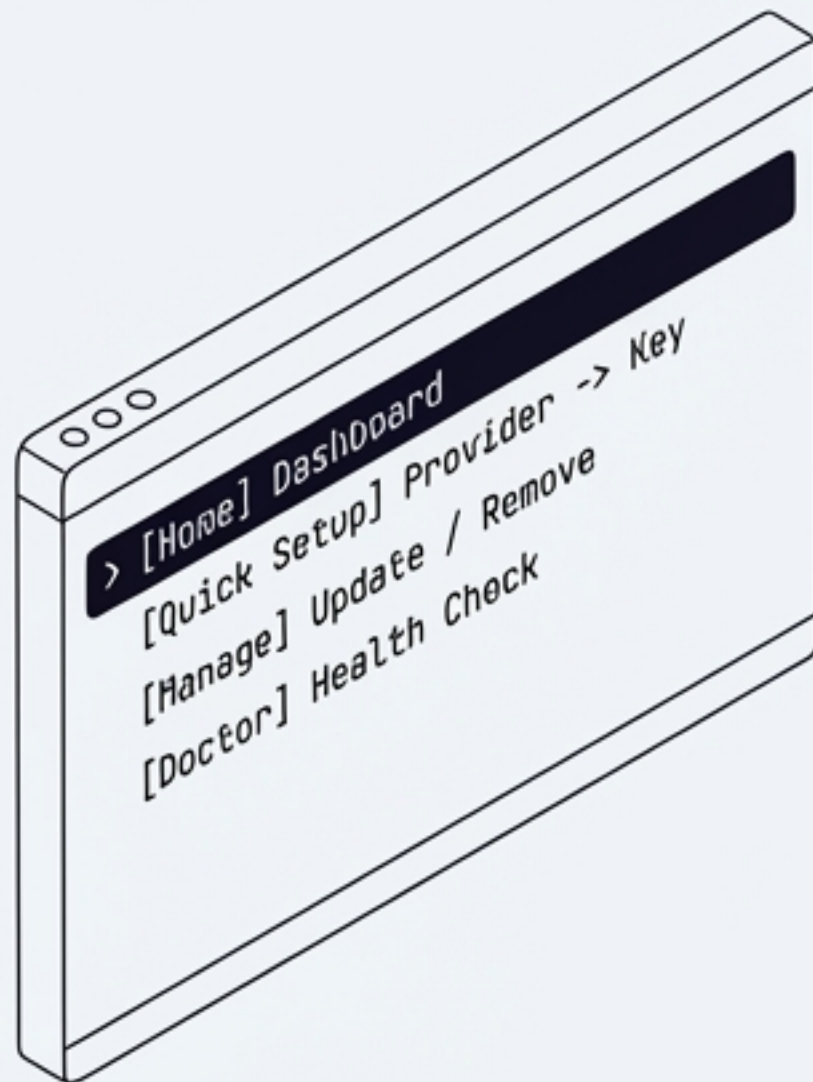




# User Experience: TUI & CLI

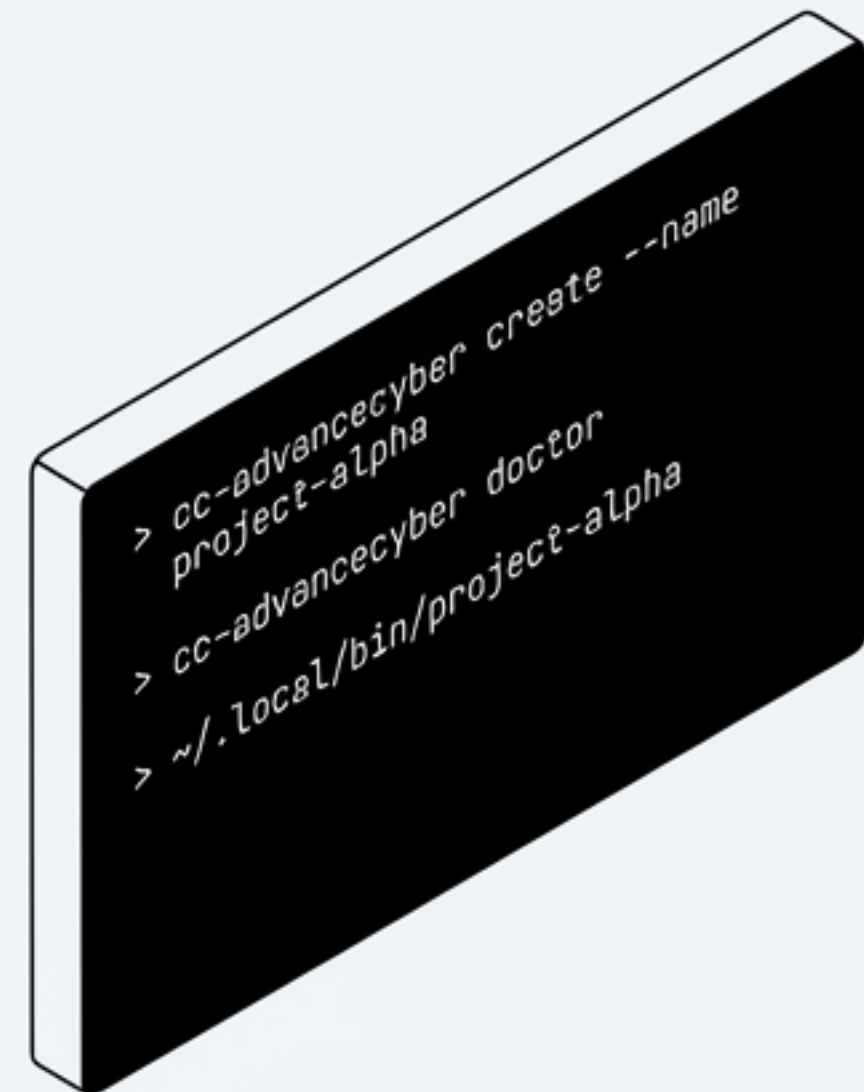
## Terminal User Interface (Ink/React)

Inter, Technical Graphite



## Native Shell Integration

Inter, Technical Graphite

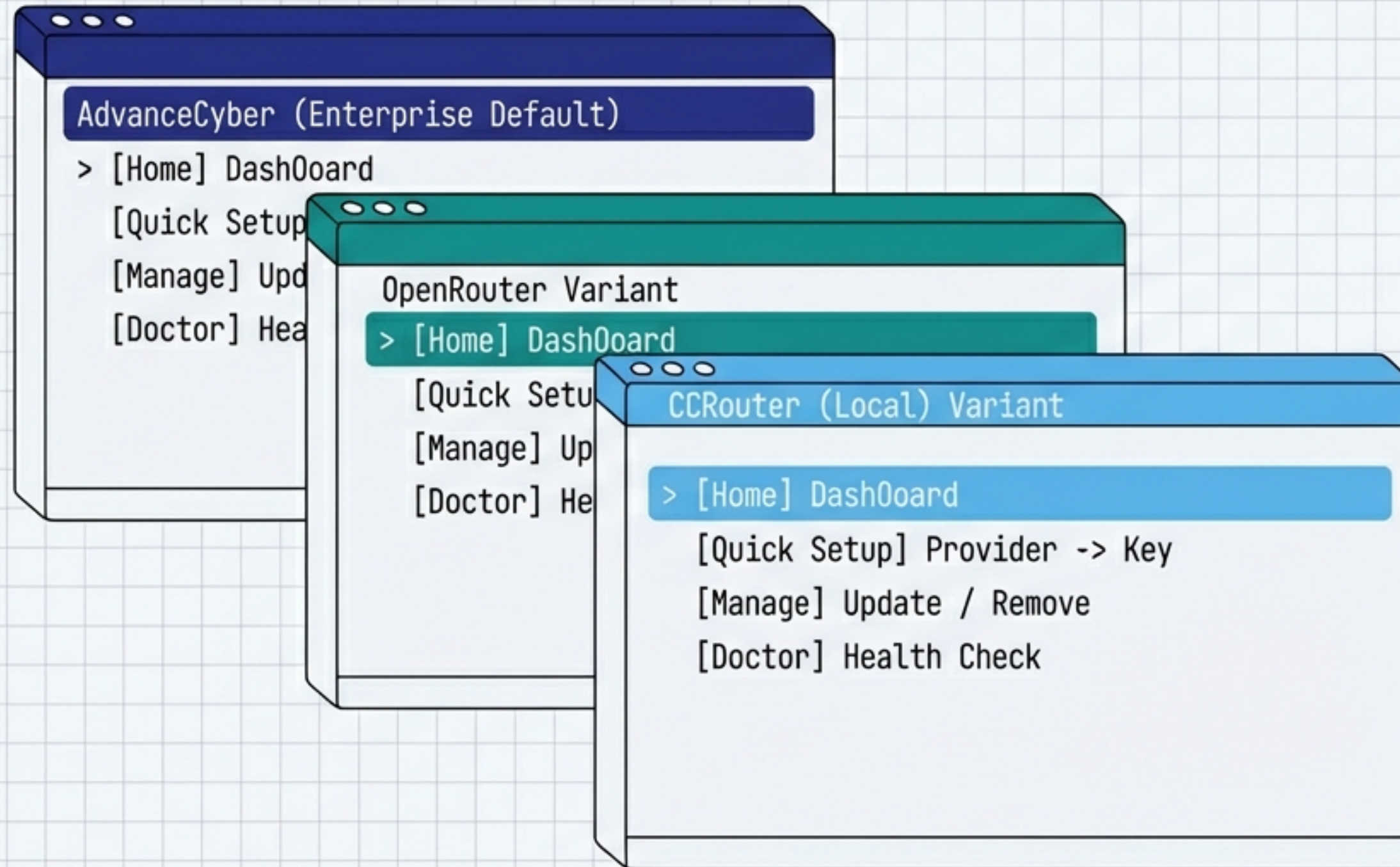




# Theming & Brand System (tweakcc)

Technical Swiss Laboratory

Visual Differentiation for Context Safety



## Customization Capabilities:

- RGB/HSL Color Pickers
- Thinking Verbs & Spinners
- Input Box Styling
- Startup Banners



# Security & Requirements

---

Technical Swiss Laboratory

## Security Architecture



**Isolation:** No modification to global `~/.claude/` directory.



**Auth:** API keys stored strictly per-variant in `settings.json`.



**Permissions:** Granular control over MCP servers and `--dangerously-skip-permissions` flags.

## Prerequisites Checklist



Node.js 18+ (Runtime)



Docker (Memory Backend)



Ollama (Local Embeddings)



# Summary & Roadmap

```
PACKAGE: cc-advancecyber  
VERSION: 1.0.0  
LICENSE: MIT  
BINARY: cc-advancecyber
```

A comprehensive, secure, and memorable AI  
AI development environment.