Claude Code as a General Agent

MGMT 675: Generative AI for Finance

Kerry Back

Recall: What is an AI Agent?

- Chatbot with tools for actions beyond conversation
- Code execution (Python, etc.)
- File generation (Excel, Word)
- Database access
- Web search and API calls

Agent Components

1. **LLM**: The "brain" (GPT-4, Claude)
2. **System Prompt**: Instructions
3. **Agent Logic**: Control flow
4. **Tools**: Code, APIs, databases

Stand-Alone Agent Architecture

Custom-Built Agent

- Hard-coded system prompt
- Hard-coded agent logic
- Specific tools integrated
- Fixed to one use case
- Requires developer to modify

Example: Rice Data Portal

Stand-Alone Agent Flow

1. User prompt → Agent
2. Agent adds system prompt
3. Combined → LLM
4. LLM → SQL or question
5. Agent executes or asks user
6. Loop until done

Claude Code: A General-Purpose Agent

- Claude Code is an agent application
- LLM: Claude Opus or Sonnet
- Built-in tools: file I/O, bash, code execution, web search
- Agent logic: built into Claude Code
- **Skills** customize it for specific tasks

Claude Code Architecture

Component	Provider
LLM	Claude Opus/Sonnet
Agent Logic	Claude Code app
Base Tools	Claude Code app
System Prompt	**Skill**
Custom Tools	**Skill scripts**

What is a Skill?

A **skill** is a folder containing instructions and optional code that transforms Claude Code into a specialized agent for a specific domain or task.

A Skill Provides

- System prompt (domain knowledge)
- Workflow instructions
- Best practices and constraints
- Python/JavaScript helper scripts
- Reference documentation

Claude Code Provides

- The LLM (Opus or Sonnet)
- Agent control loop
- File read/write tools
- Code execution sandbox
- Web search capability

Anatomy of a Skill

Skill Folder Structure

```
skills/
  xlsx/
    SKILL.md      <- Main file
    scripts/
      recalc.py
    references/
      schema.md
```

SKILL.md Structure

```
---
name: xlsx
description: "Excel file..."
---

# Requirements for Outputs
- Zero formula errors
- Use formulas, not hardcodes

# Workflows
1. Choose pandas or openpyxl
2. Create/modify file
3. Recalculate formulas
```

SKILL.md: The System Prompt

- **Frontmatter**: Name and description (YAML header)
- **Requirements**: Quality standards and constraints
- **Workflows**: Step-by-step procedures
- **Code Examples**: Patterns for the LLM to follow
- **Error Handling**: How to diagnose and fix problems

The SKILL.md file is automatically injected into Claude's context when working in a project that contains the skill.

Scripts: The Custom Tools

- Python or JavaScript files in `scripts/` folder
- Claude Code can execute them
- Extend Claude's capabilities
- Handle tasks LLM can't do directly

Example Scripts

- `recalc.py` – Recalculate Excel formulas via LibreOffice
- `validate.py` – Check file structure
- `unpack.py` – Extract XML from Office files
- `html2pptx.js` – Convert HTML to PowerPoint

Example: Rice Database Skill

Stand-Alone App

- Custom web application
- Hard-coded SQL generation prompt
- Fixed agent logic in Python
- Single-purpose: query database
- Requires developer to update

Claude Code + Skill

- SKILL.md with database schema
- Connection code examples
- Table descriptions
- **Plus**: Can also make charts, Excel files, Word reports
- User can extend easily

Same database access, but infinitely more flexible

Skill vs Stand-Alone Agent

Feature	Stand-Alone	Skill
Agent logic	Custom code	Claude Code
System prompt	Hard-coded	SKILL.md file
LLM	Your choice	Claude Opus/Sonnet
Tools	Custom built	Scripts + built-in
Maintenance	Developer	Edit markdown
Combine tasks	No	Yes

Skills let you create **specialized agents** without writing agent logic. Claude Code handles the control flow; you just provide the domain knowledge.

Where Skills Live

- **Project skills**: `.claude/skills/` in your project folder
- **User skills**: `~/.claude/skills/` (shared across projects)
- Claude Code automatically loads skills from both locations
- Skills can reference each other (e.g., xlsx skill uses ooxml validation)

Community Skills Repository: github.com/VoltAgent/awesome-agent-skills
200+ skills from official teams and the community

Creating Your Own Skill

1. Create folder: `.claude/skills/my-skill/`
2. Create `SKILL.md` with:
   - YAML frontmatter (name, description)
   - Domain knowledge and instructions
   - Code examples and workflows
3. Optionally add `scripts/` folder with helper code
4. Start using Claude Code – skill is automatically loaded

Summary

Claude Code

- General-purpose agent
- LLM + Agent Logic + Tools
- Works out of the box

Skills

- Specialize Claude Code
- System prompt + scripts
- Easy to create and modify

Skills replace stand-alone agents

Same capabilities, less code, more flexibility