Agentic AI Workflows

From Direct Generation to Iterative Intelligence

Vanilson Burégio

What is an Agentic Workflow?

An Agentic AI workflow is a process where an LLM-based app executes multiple steps to complete a task.

Example: Planning a 5-day trip to Tokyo

Single-shot Prompting Approach:

- Generate complete itinerary in one go
- No research or verification
- Generic recommendations
- May miss key details or conflicts

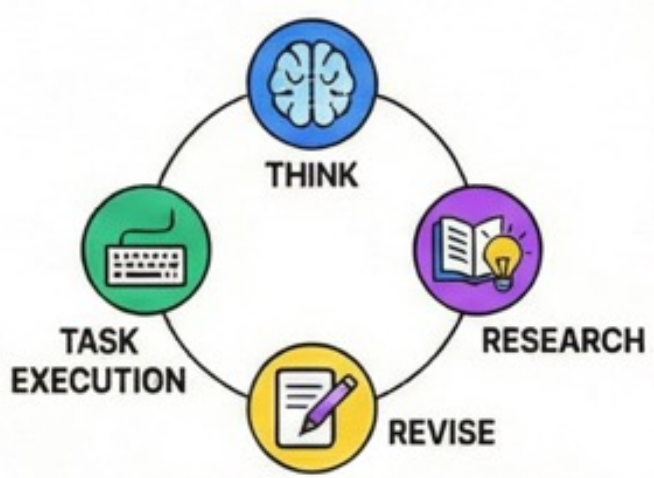
Agentic Approach:

- Research destinations and activities
- Check availability and hours
- Verify logistics and timing
- Iterate and optimize the plan

Agentic Trip Planning Process

Example: Planning a 5-day trip to Tokyo

1. Research top attractions & neighborhoods
2. Check opening hours & seasonal events
3. Create initial day-by-day itinerary *(draft)*
4. Review travel times & logistics *(review draft)*
5. Add restaurant recommendations
6. Optimize for time & proximity



Each step involves multiple API calls, data validation, and decision-making

Degree of Autonomy

*"Agentic" acknowledges that **systems can be autonomous to different degrees** - not a binary choice of agent vs. non-agent.*

Less Autonomous

Fixed sequence:
Search → Select → Schedule
Tokyo temples → lunch spots → evening activities

Semi-Autonomous

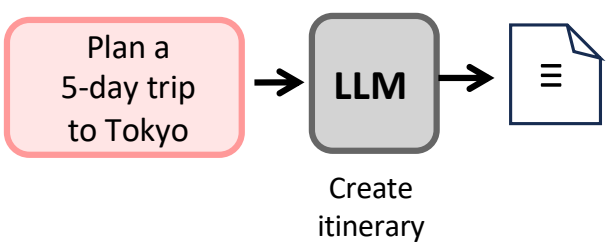
LLM chooses tools:
Web search OR review sites
Decide: TripAdvisor vs Google Maps vs blogs

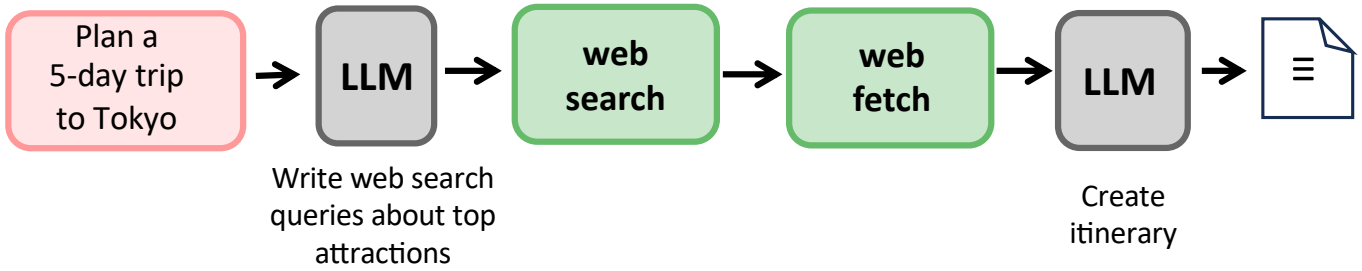
Highly Autonomous

Plans entire workflow:
Decides steps dynamically
Adapts to weather, finds alternatives, rebooks

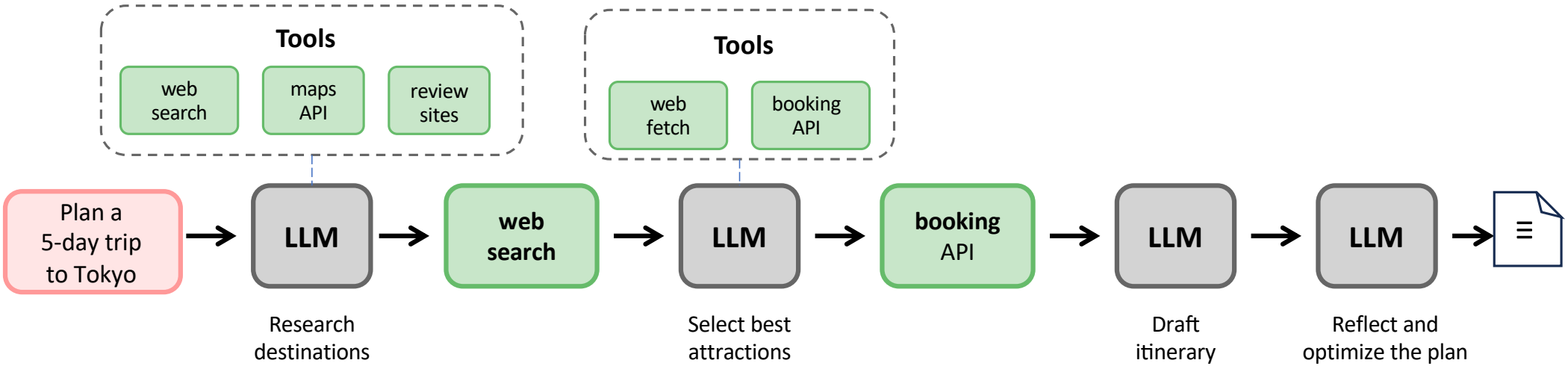
Degree of Autonomy

Less autonomous



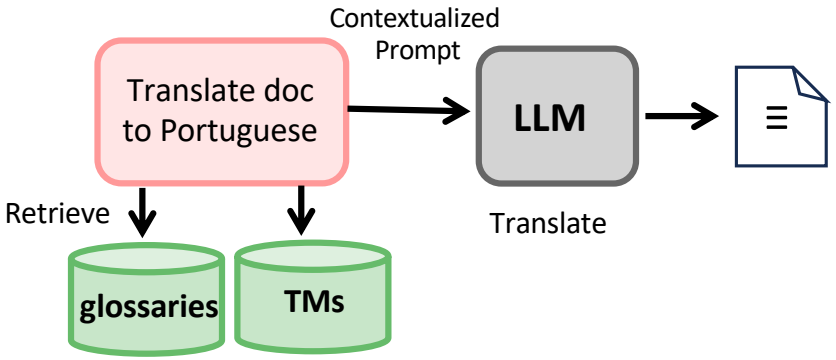


More autonomous

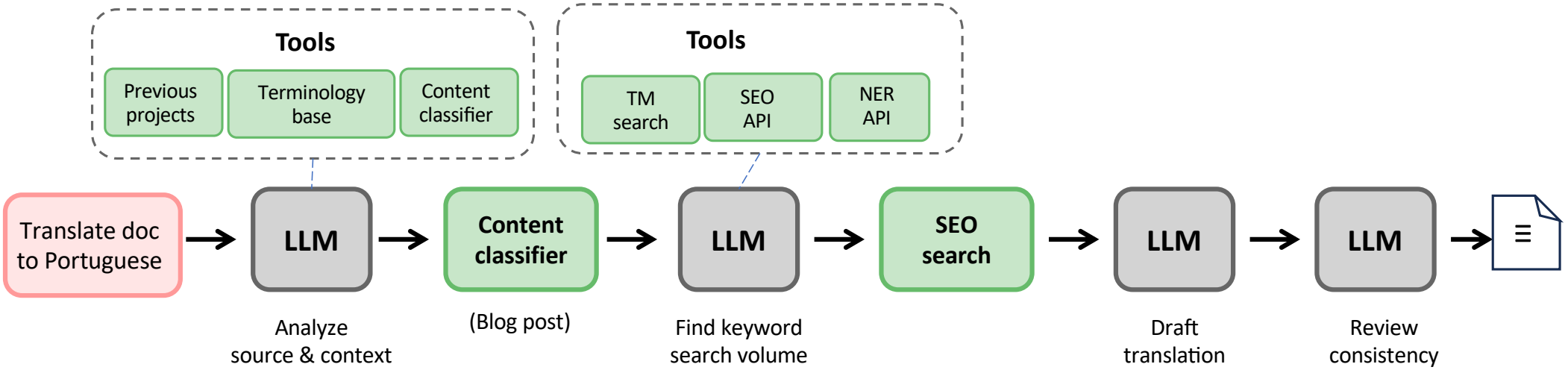


Degree of Autonomy (translation context)

Less autonomous



More autonomous



When to Delegate Autonomy to an Agentic System?

When to Delegate Autonomy to an Agentic System?

- **The workflow is not deterministic**
  The next step depends on what happens during execution.

- **Step execution depends on many factors that are hard to encode as rules**
  *Context, **unknown**, incomplete or changing information.*

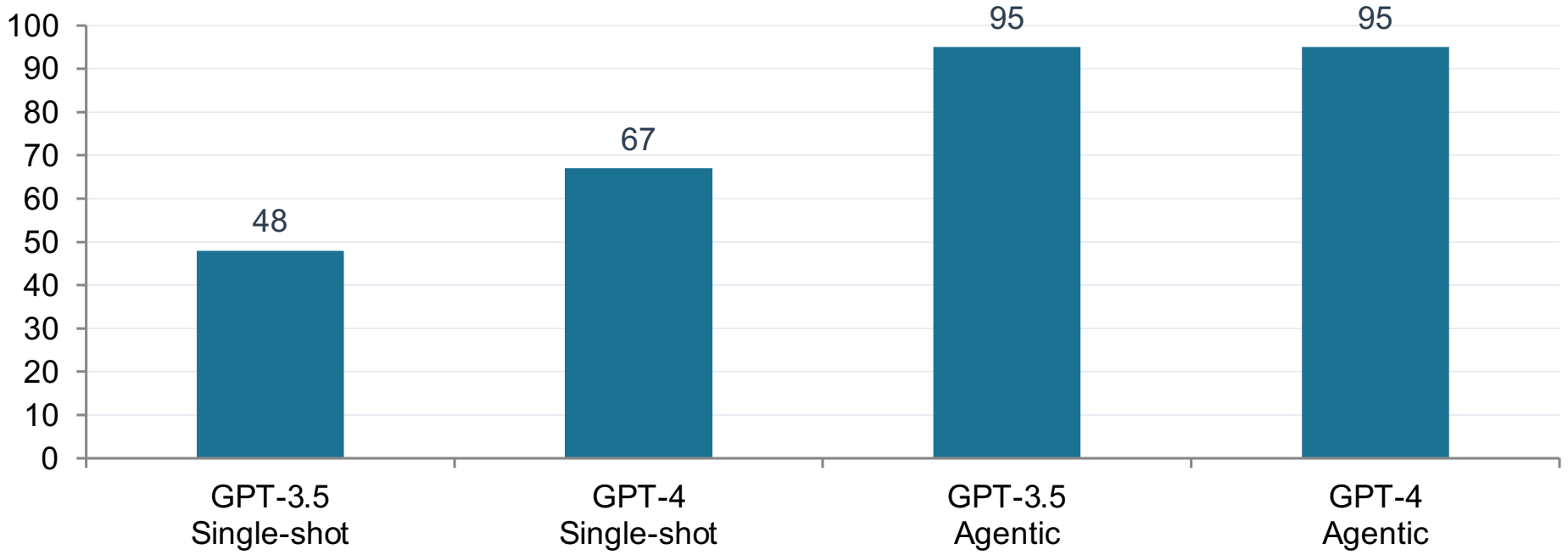
- **Rule-based implementations would be costly or complex**
  Too many if–else conditions to build and maintain.

- **User intent must be interpreted**
  Common in chat-based or open-ended interactions.

- **Performance needs to go beyond single-shot prompting**
  When iterative reasoning, reflection, or tool use is required to outperform a one-pass solution.

Benefits: Dramatic Performance Gains

Human Eval Coding Benchmark



Agentic workflows often outperform even upgrading to the next generation model

Additional Benefits

Parallelism

Trip Planning Example:

- Check 10 hotel availabilities simultaneously
- Fetch reviews from multiple sources in parallel
- Query weather data for all destinations
- Verify opening hours for 20+ attractions at once

Modularity

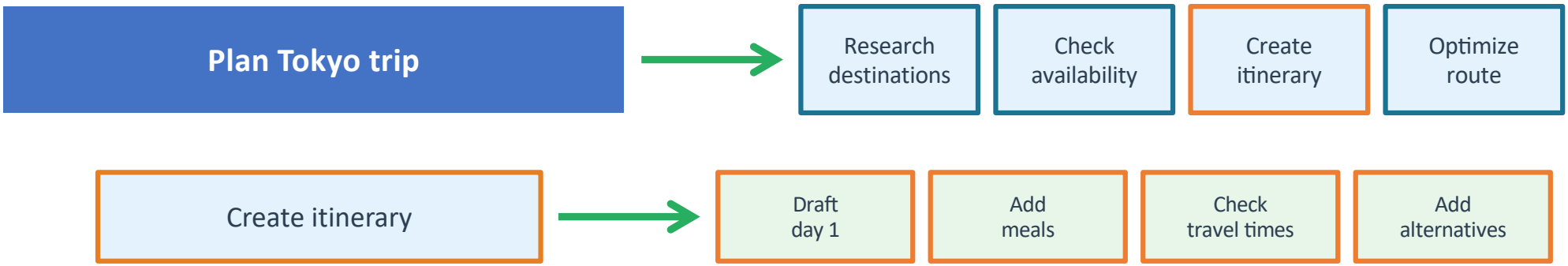
Swap Components:

- Different map providers (Google, Apple, Mapbox)
- Multiple booking APIs (Expedia, Booking.com)
- Various review sources (TripAdvisor, Yelp)
- Test different LLMs for each step

Designing Agentic Workflows is all about **Decomposition**

Task Decomposition

Breaking trip planning into discrete, implementable steps



Key Question: Can each step be done by an LLM, code, function call, or tool?

✓ Search attractions → Web search API

✓ Check hours → Google Maps API

✓ Calculate routes → Mapping service

✓ Draft schedule → LLM with context

Detailed Example: Day 1 Planning

Draft
day 1

Planning the first day in Tokyo with agentic workflow

User Query	"Plan my first day in Tokyo visiting temples"
Research Phase	Search top temples → Senso-ji, Meiji Shrine found
Verify Details	Check opening hours: Senso-ji 6am-5pm, Meiji Shrine sunrise-sunset
Add Context	Find nearby: Nakamise Shopping Street, Ueno Park
Create Draft	Morning: Senso-ji → Afternoon: Meiji Shrine → Evening: Harajuku
Review & Optimize	Check travel times: 45min between locations, add lunch at 12:30pm
Final Output	Optimized itinerary with times, addresses, directions, alternatives

Building Blocks for Trip Planning

AI Models

- LLMs for planning
- Recommendation engines
- Image recognition
- Translation models

Travel APIs

- Google Maps/Places
- Booking.com API
- TripAdvisor
- Weather APIs

Data Tools

- Review aggregation
- Price comparison
- Real-time availability
- User preferences DB

Four Key Agentic Design Patterns

Reflection

Review itinerary for conflicts, improve recommendations

Tool Use

Call APIs for maps, bookings, weather, reviews

Planning

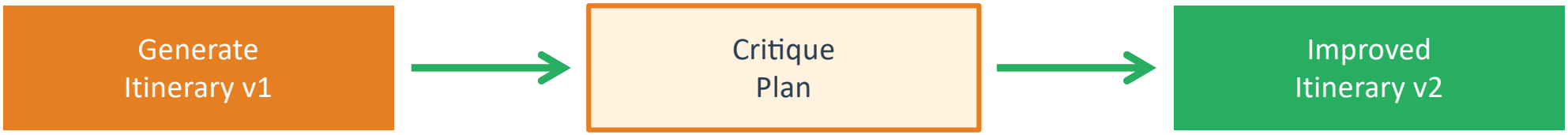
Determine steps: research → verify → book → optimize

Multi-Agent

Specialist agents: researcher, logistics, food expert

Pattern 1: Reflection

Applied to trip planning



Example Reflection Process:

Initial Plan: *Morning Senso-ji → Afternoon Shibuya → Evening Asakusa*

Critique: "Inefficient route - Asakusa and Senso-ji are in same area!"

Improved Plan: *Morning Senso-ji + Asakusa area → Afternoon Shibuya → Evening Shinjuku*

Pattern 2: Tool Use

LLM decides which travel tools to use

Google Places

Find top-rated temples in Tokyo

Maps API

Calculate 45min travel Senso-ji → Shibuya

Weather API

Check rain forecast for outdoor activities

Booking API

Check hotel availability near station

Review Scraper

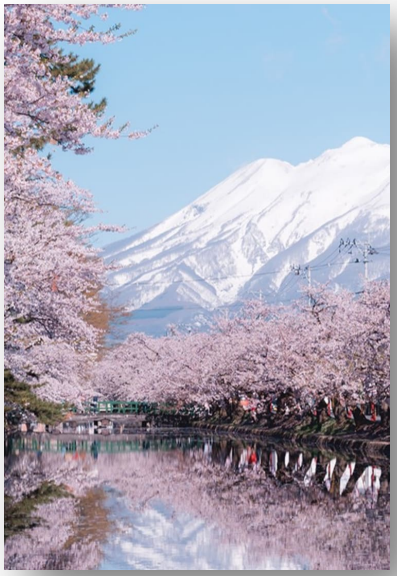
Aggregate ratings from 5 sources

Translation

Convert addresses to English

Pattern 3: Planning

LLM autonomously determines the workflow sequence



User: "Plan a cherry blossom viewing trip to Japan in April"

1 **Research**
Cherry blossom forecast for April 2026

→

2 **Prioritize**
Best viewing locations Kyoto, Tokyo, Osaka

→

3 **Check**
Peak bloom dates by city

→

4 **Book**
Hotels near top viewing spots

→

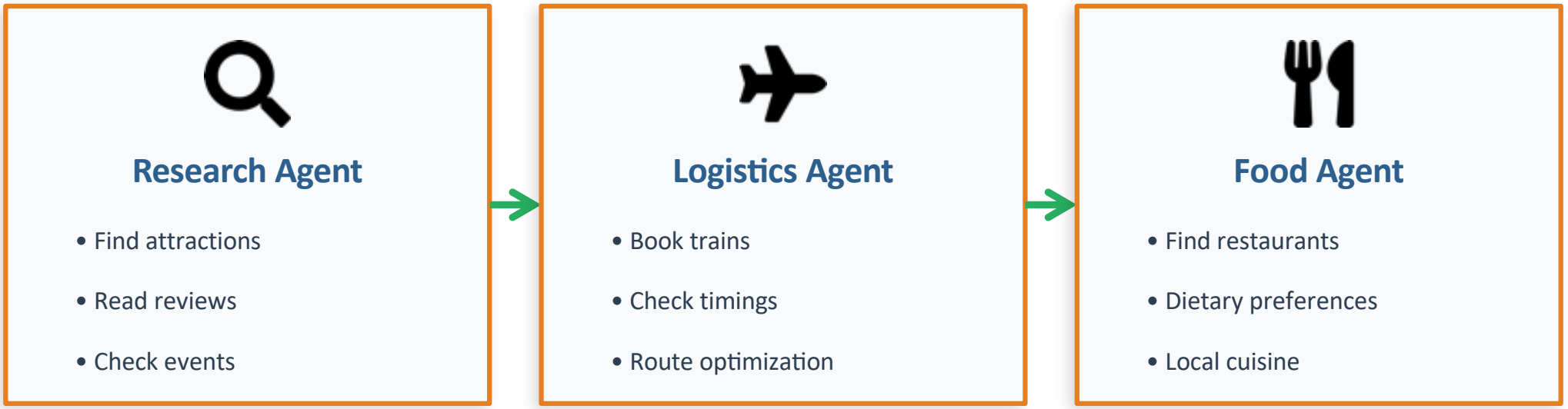
5 **Plan**
Daily itineraries around parks

Agent decides: research timing before location, then optimize bookings

Pattern 4: Multi-Agent Collaboration

Specialized agents working together on trip planning

Task: Plan 10-day multi-country Europe trip



Agents collaborate: researcher finds options → logistics checks feasibility → food adds meals

What are the implementation Building Blocks?

Implementation Building Blocks

Skills

Procedural knowledge

The HOW

Tools

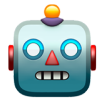
Executable capabilities

The WHAT

MCP

External connectivity

The WHERE



Subagents

Task specialists

The WHO

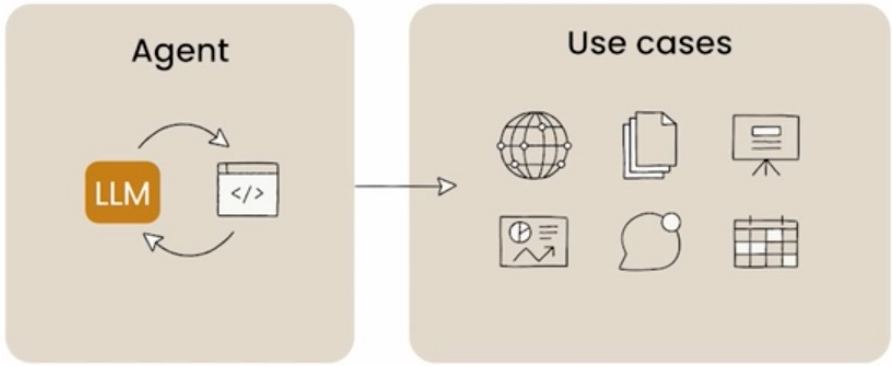
Models

Specialist Agents vs Generalist Agents

Domain-specific Agents

Single-purpose agents

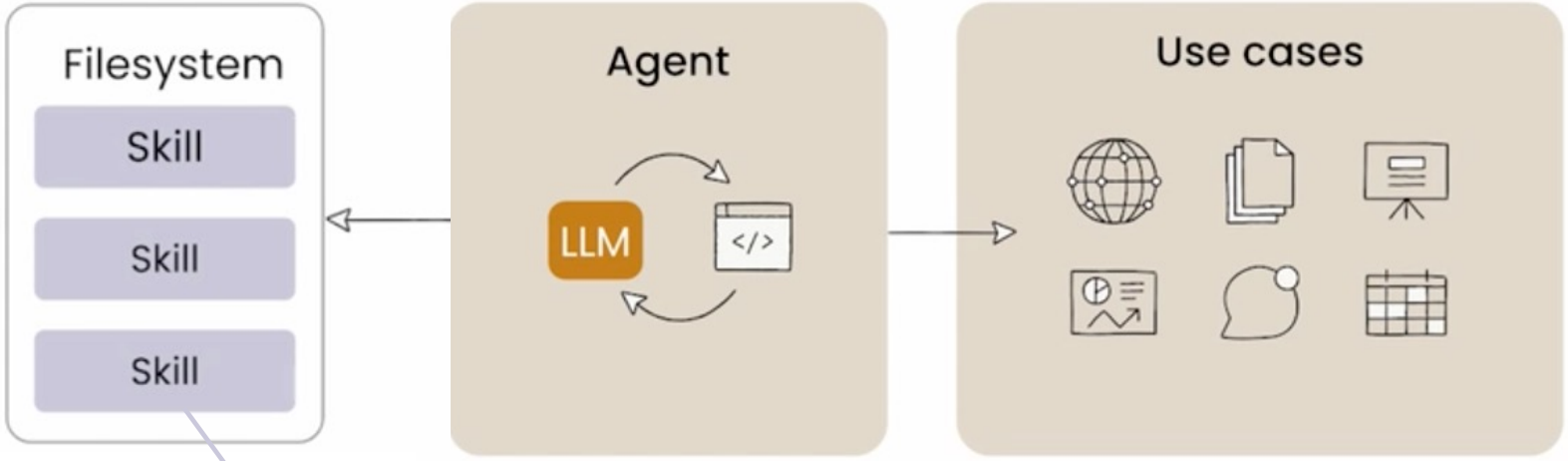
General-purpose Agents



✓ **Code as universal interface**

One agent, many use cases

General-purpose Agents with Skills



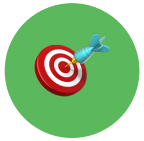

```
pdf-skill/
├── SKILL.md (main instructions)
├── FORMS.md (form-filling guide)
├── REFERENCE.md (detailed API reference)
└── scripts/
    └── fill_form.py (utility script)
```

What Are Agent Skills?

Modular, dynamically-loadable units of procedural knowledge that transform general-purpose agents into domain specialists

Domain Expertise

Repeatable Workflows



Org-Specific Context

New Capabilities

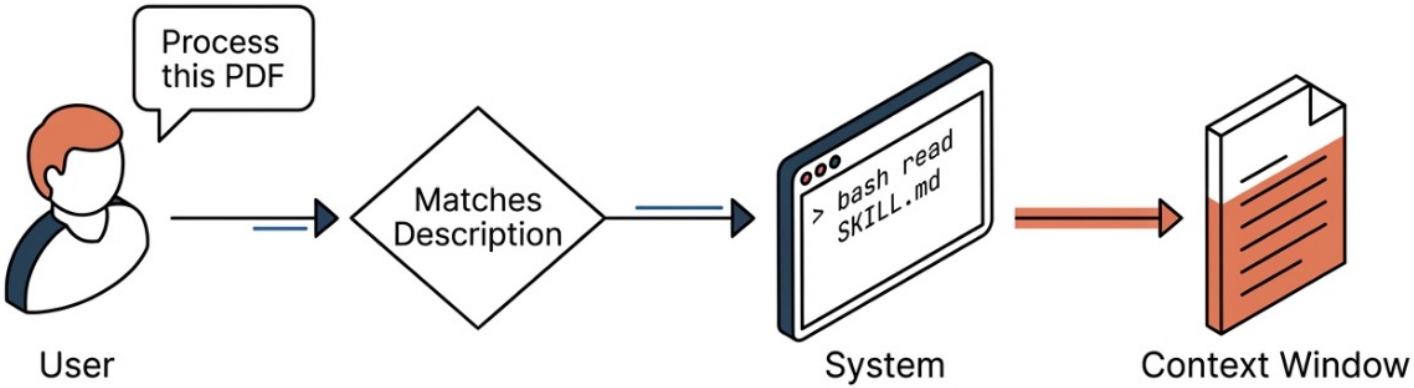
Skills encode how to do things*, not just what to say*

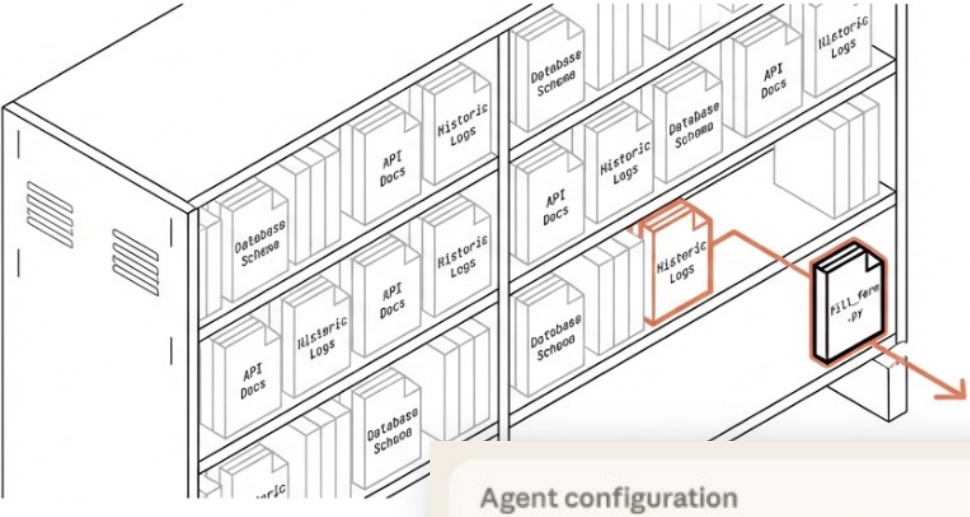
They transform agents from conversational to operational systems

Skill Structure

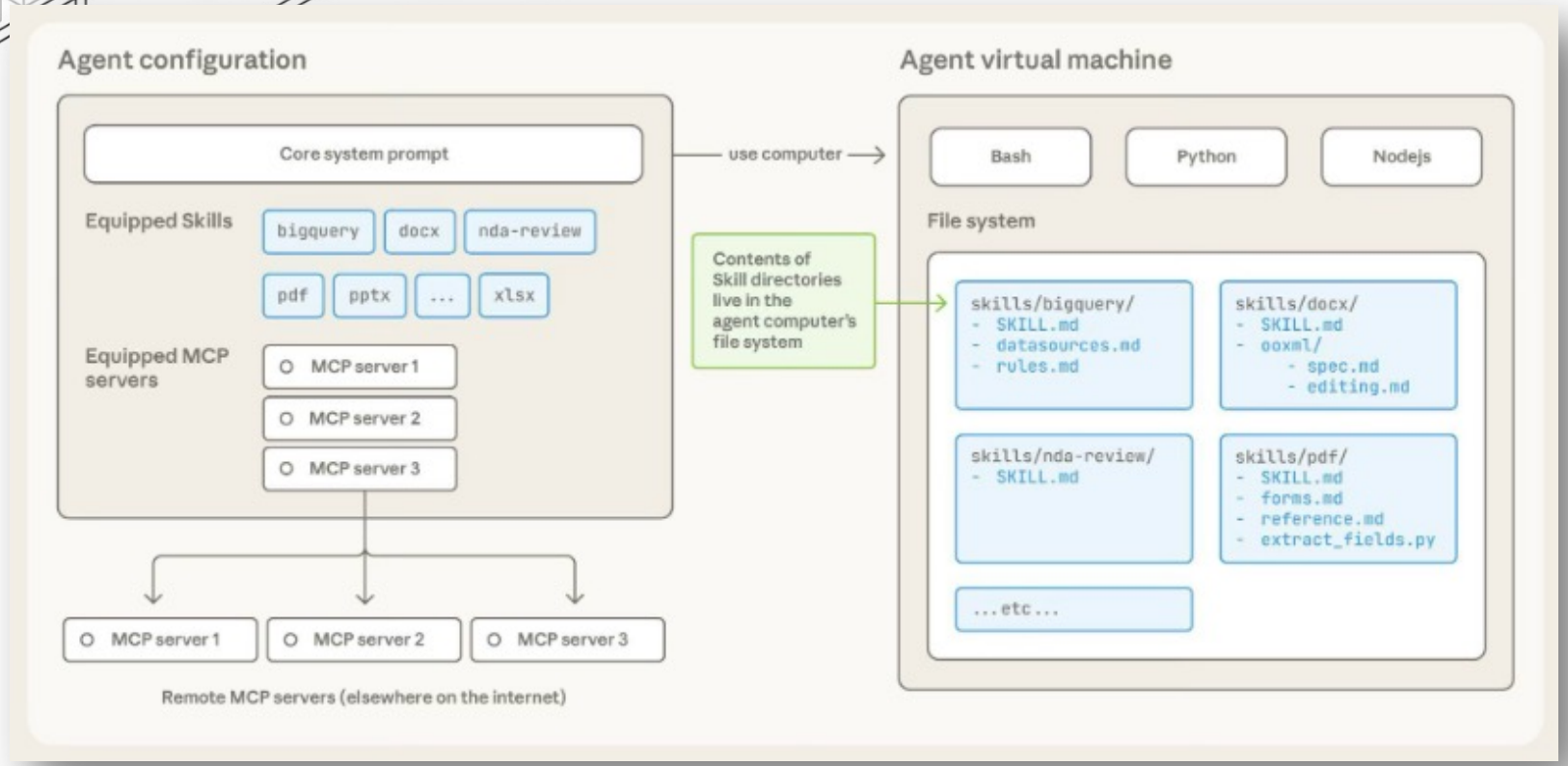
```
---
name: your-skill-name
description: Brief description of what this Skill does and when to use it
---

# Your Skill Name

## Instructions
[Clear, step-by-step guidance for Claude to follow]

## Examples
[Concrete examples of using this Skill]
```

This text triggers the skill loading

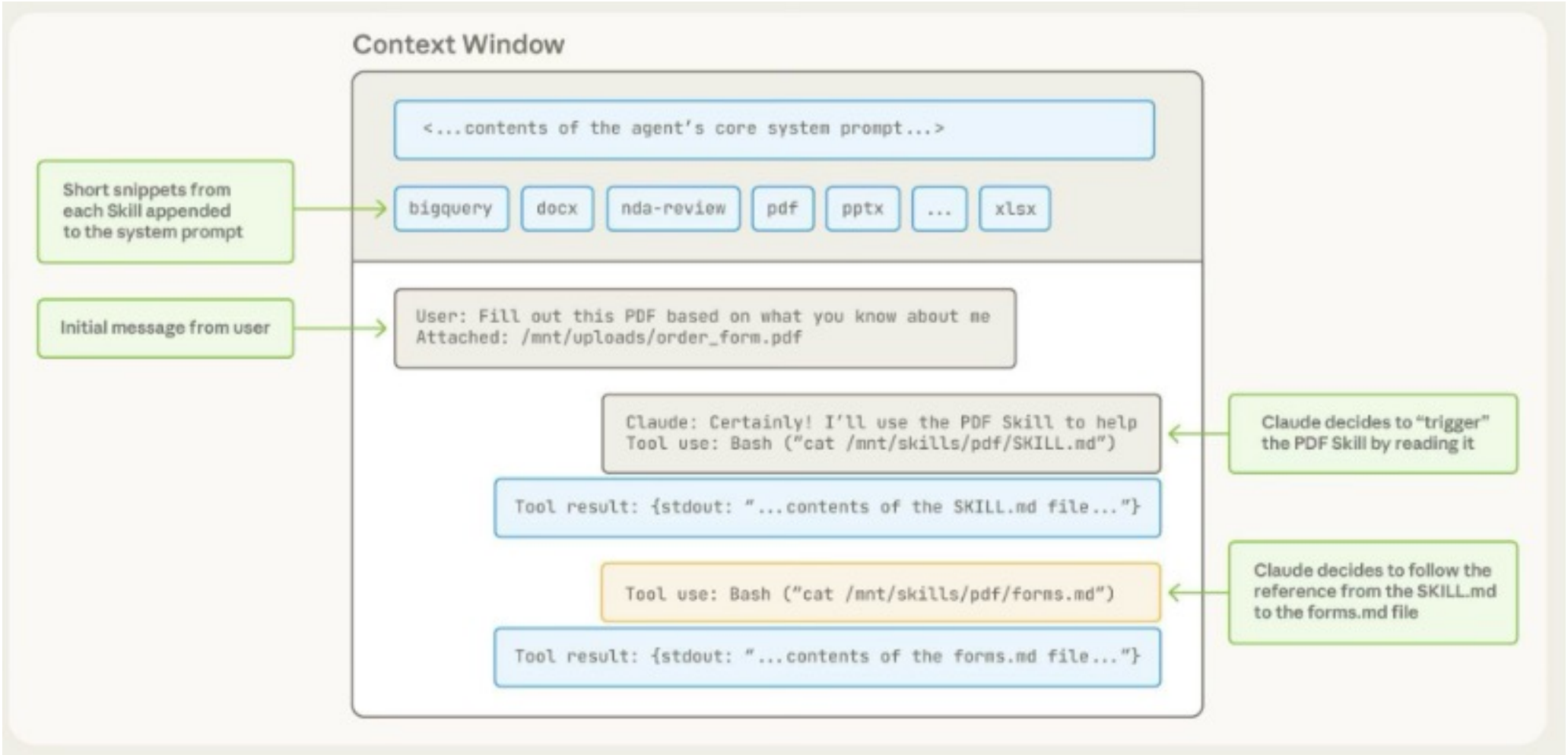




“Unlimited” resource Library



Skills in Claude



Implementation Difficulty Spectrum

Easier

Weekend city trip with popular attractions

- Pre-defined templates
- Standard destinations
- Clear step sequence
- Limited customization

Medium

Multi-country itinerary with logistics coordination

Harder

Adventure trip with weather-dependent activities & real-time rebooking

- Complex multi-city trips
- Dynamic replanning
- Real-time adaptations
- Handling uncertainties

Key Takeaways

Agentic workflows execute multiple steps iteratively – a workflow goes from generic to personalized

Research → Verify → Draft → Optimize produces superior itineraries with validated information

Break complex tasks into discrete steps: each step uses LLMs, APIs, or specialized tools

Four design patterns enable sophisticated workflows: Reflection, Tool Use, Planning, Multi-Agent

Start simple with clear processes, then add complexity as needed for better results

Next Round: From Concepts to Practice

How can we apply these concepts to both our *engineering practices* and *the system itself*?

Agentic AI Workflows

From Direct Generation to Iterative Intelligence

Vanilson Burégio