

Agentic AI Workflows

From Direct Generation
to Iterative Intelligence

Vanilson Burégio

wxrks

What is an Agentic Workflow?

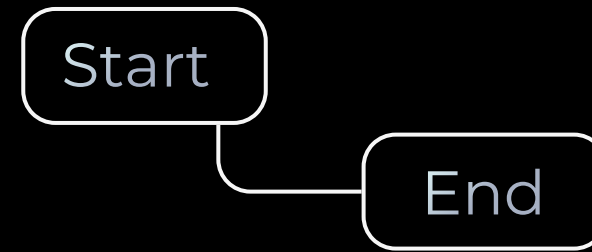
An Agentic AI workflow is a process where an LLM-based app executes multiple steps to complete a task.





Example

Planning a 5-day trip to Tokyo



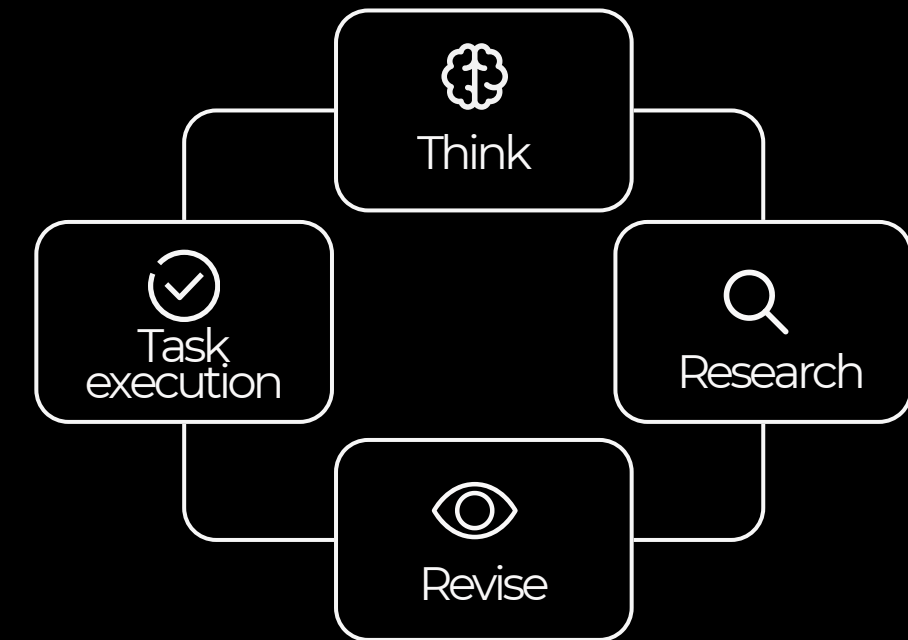
Single-shot Prompting Approach

Generate complete itinerary in one go

No research or verification

Generic recommendations

May miss key details or conflicts



Agentic Approach

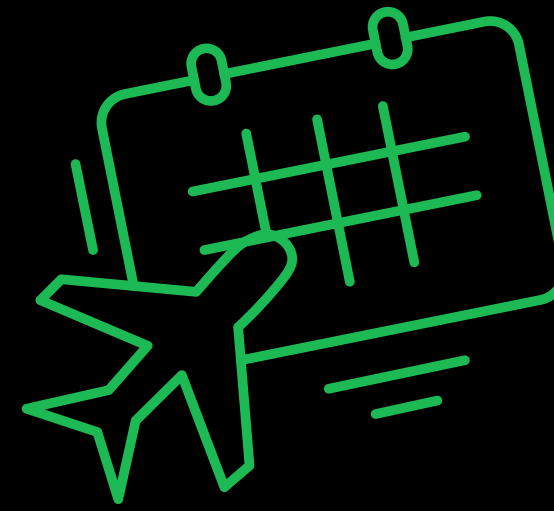
Research destinations and activities

Check availability and hours

Verify logistics and timing

Iterate and optimize the plan

Agentic Trip Planning Process



Planning a 5-day trip to Tokyo I



Research top attractions & neighborhoods



Check opening hours & seasonal events



Create initial day-by-day itinerary (draft)



Review travel times & logistics (review draft)



Add restaurant recommendations



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

● ● ●

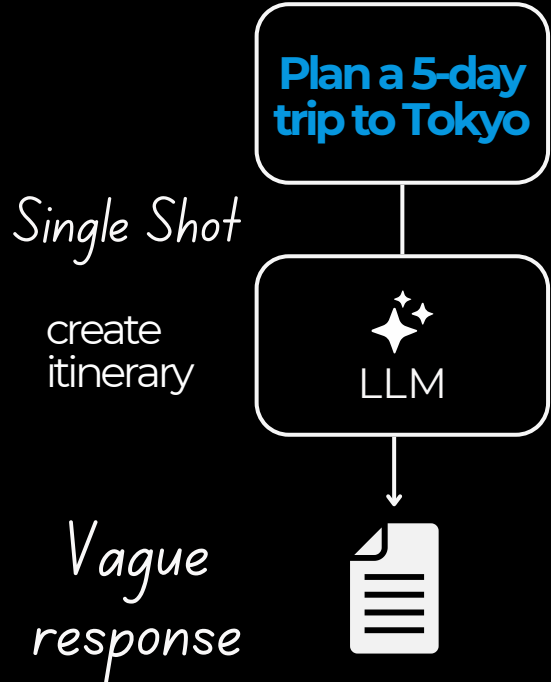
Highly Autonomous

Plans entire workflow:

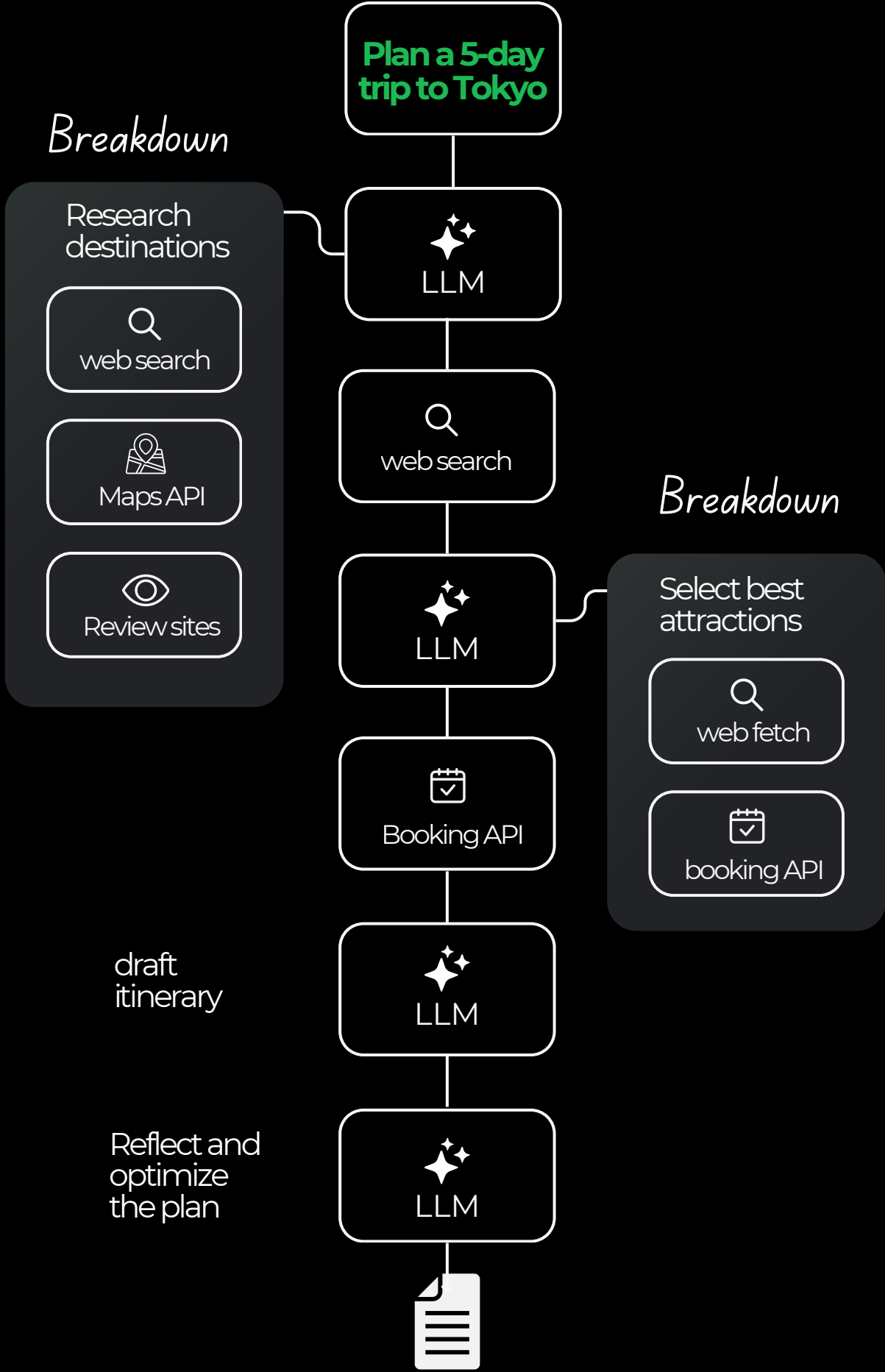
Decides steps dynamically

Adapts to weather,
finds alternatives,
rebooks

Less Autonomous



More Autonomous



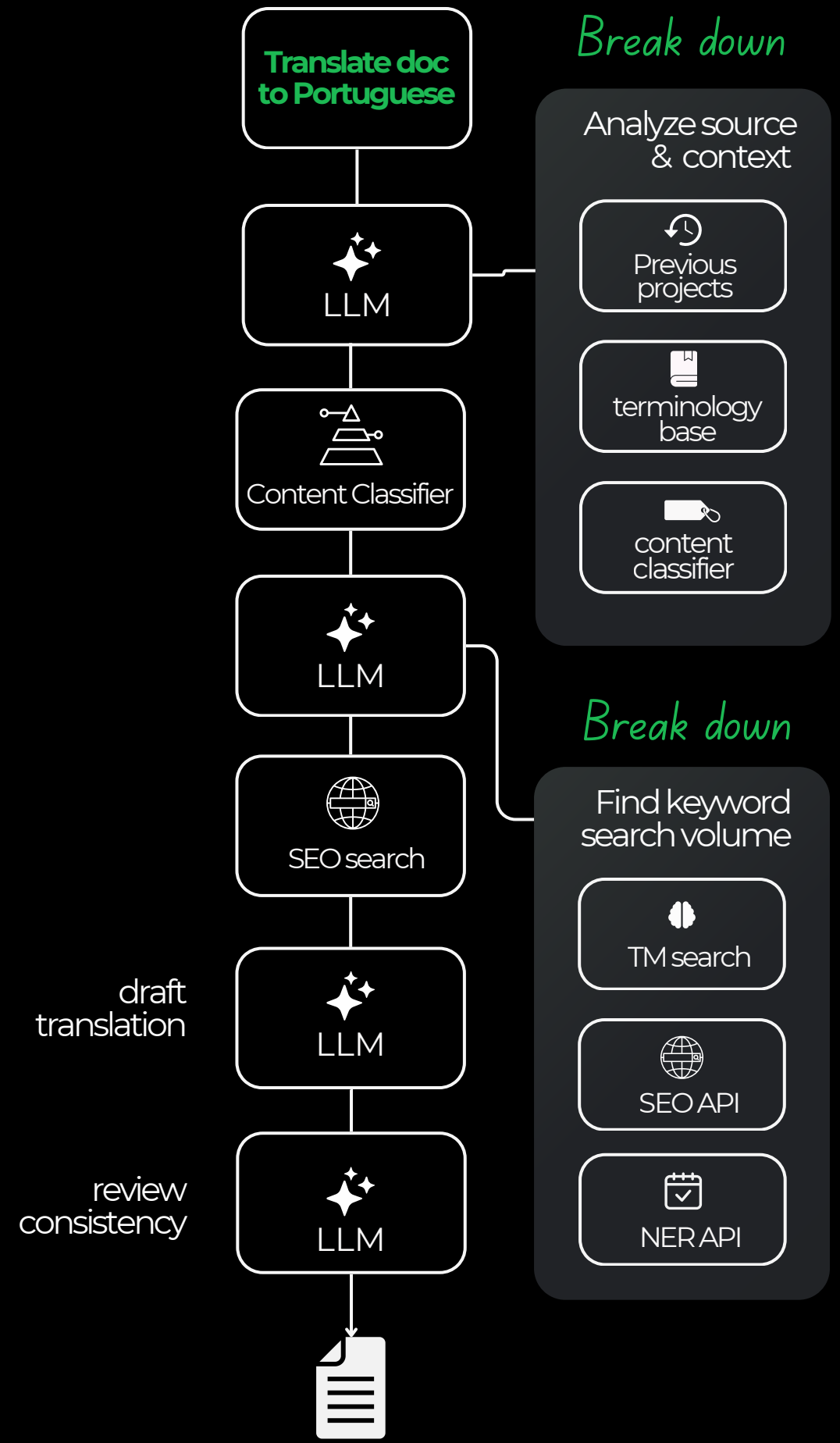
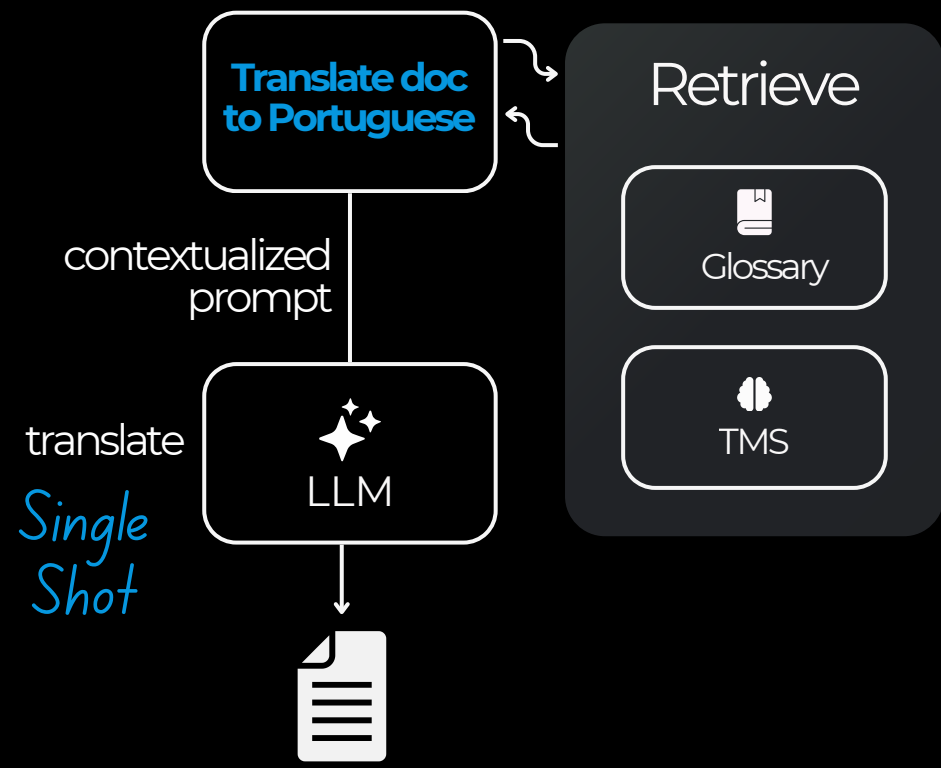
Degree of Autonomy

Less Autonomous

Versus

More Autonomous

Degree of Autonomy





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

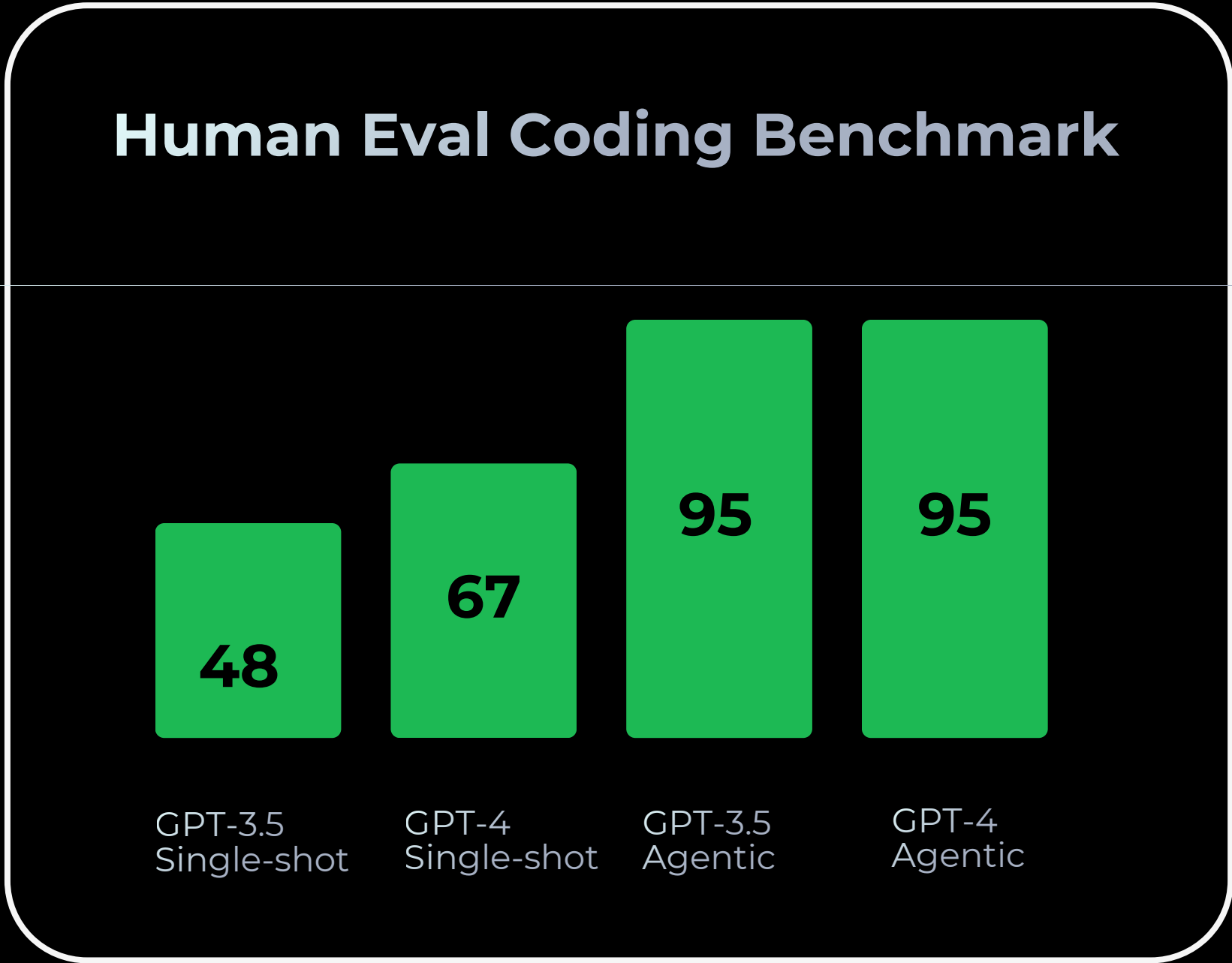
Common in chat-based or open-ended interactions

Performance needs to go beyond single-shot prompting

When iterative reasoning, reflection, or tool use is required to outperform a one-pass solution

Benefits

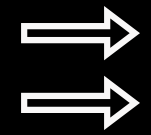
Dramatic Performance Gains



Agentic workflows often outperform even upgrading to the next generation model.

Additional Benefits

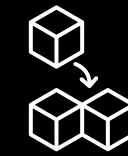
Parallelism



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

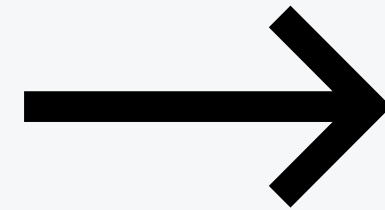
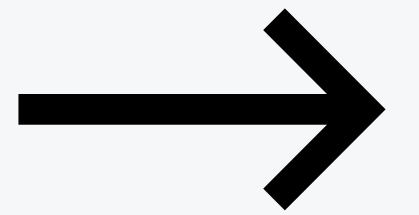
Modularity



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



→ Deccom

→ position

→ Task Decomposition

Breaking trip planning into discrete, implementable steps:

Plan Tokyo trip

- Research destinations
- Check availability
- Check itinerary
- Optimize route

Create itinerary

- Draft day 1
- Add meals
- Check travel times
- Add alternatives

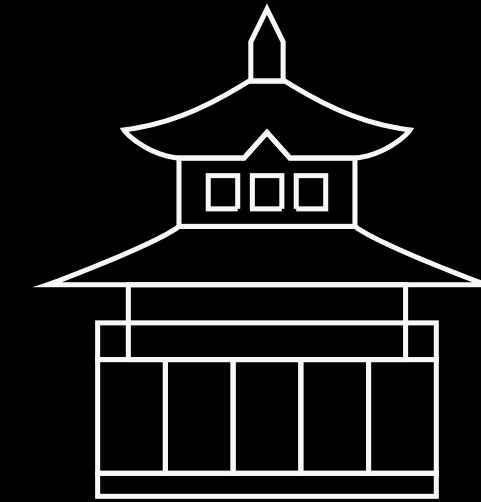
Key Question

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



→
User Query

“Plan my first day in Tokyo visiting temples”

→
Research Phase

Search top temples
→ Senso-ji,
Meiji Shrine

→
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 (45 min 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

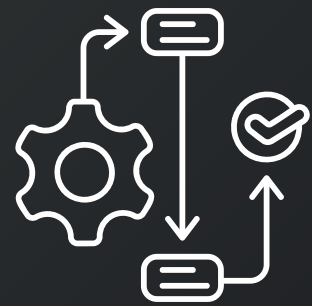
Travel APIs

- 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



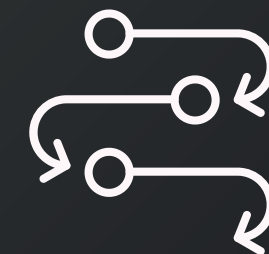
Tool Use

Call APIs for maps, bookings, weather, reviews



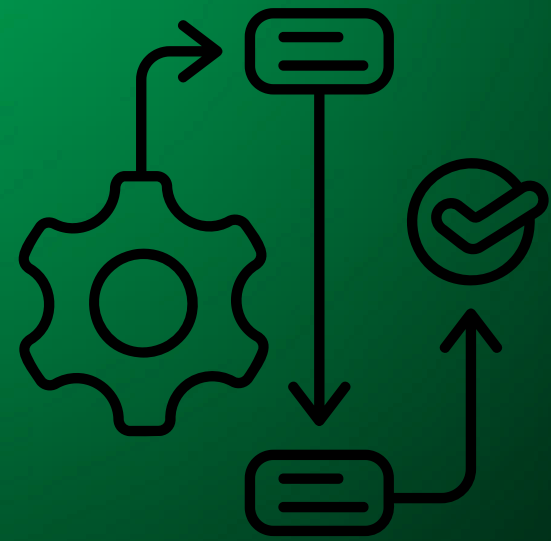
Planning

Determine steps:
→ research
→ verify
→ book
→ optimize



Multi-Agent

Specialist agents:
researcher,
logistics,
food expert



Pattern 1

Reflection

Generate Itinerary v1 →

Initial Plan

Morning Senso-ji
→ Afternoon

Critique Plan →

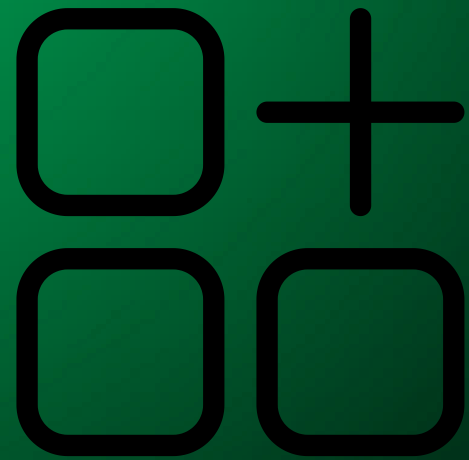
Critique

“Inefficient route.
Asakusa and Senso-ji
are in same area!”

Improved Itinerary v2

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

Maps API
→ Calculate travel time

Weather API
→ Check rain forecast

Booking API
→ Check hotel availability

Review Scraper
→ Aggregate ratings

Translation
→ Convert addresses



Pattern 3

Planning

LLM autonomously determines the workflow sequence

“Plan a cherry blossom viewing trip to Japan in April” I

Research ✓
cherry blossom forecast
(April 2026)

Prioritize ✓
best viewing locations
(Kyoto, Tokyo, Osaka)

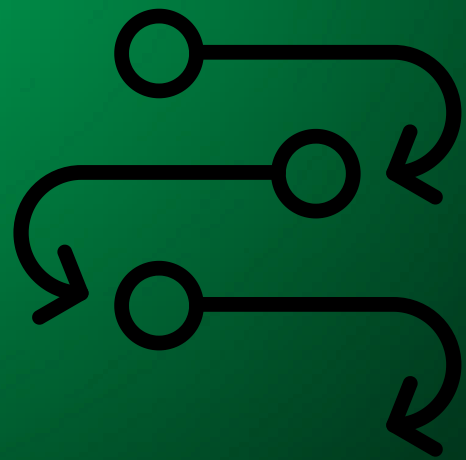
Check ✓
peak bloom dates by city

Book ✓
hotels near viewing spots

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

Plan 10-day
multi-country
Europe trip



Research Agent

Find attractions
Read reviews
Check events



Logistics Agent

- Book trains
- Check timings
- Route optimization

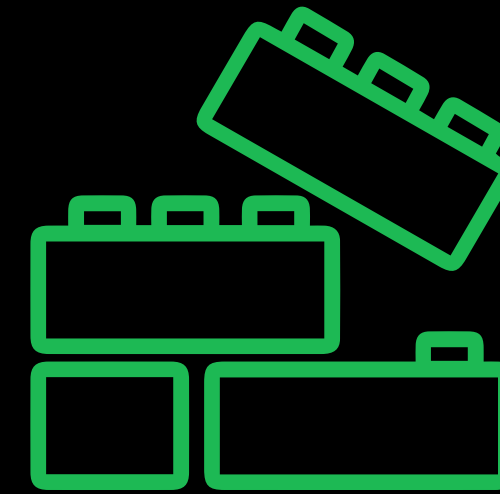


Food Agent

- Find restaurants
- Dietary preferences
- Local cuisine

Agents collaborate:
→ Researcher finds options
→ Logistics checks feasibility
→ Food adds meals

What are the implementation Building Blocks?



Skills

Procedural knowledge

The HOW

Tools

Executable capabilities

The WHAT

MCP

External connectivity

The WHERE

Subagents

Task specialists

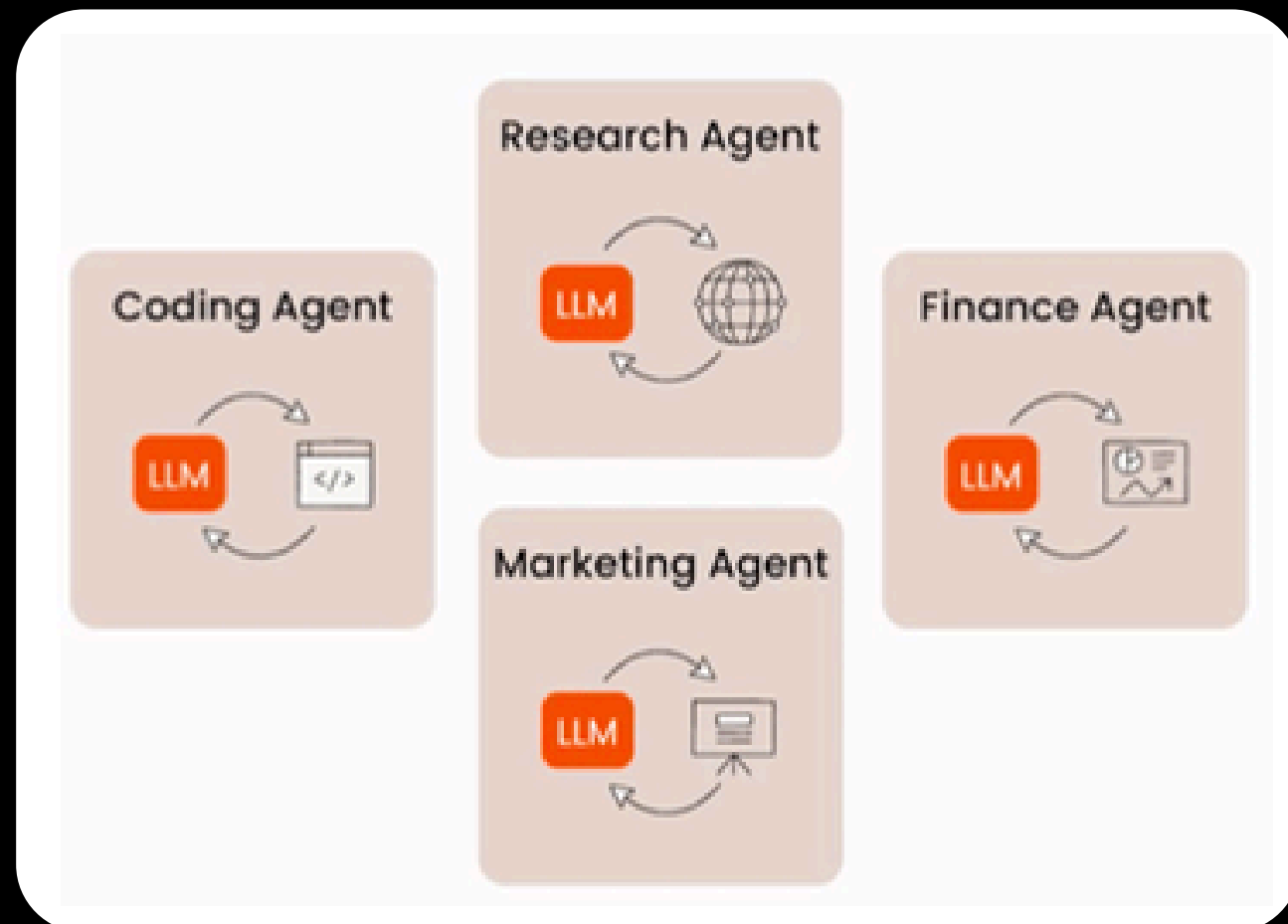
The WHO

Models

Specialist Agents

Domain-specific Agents

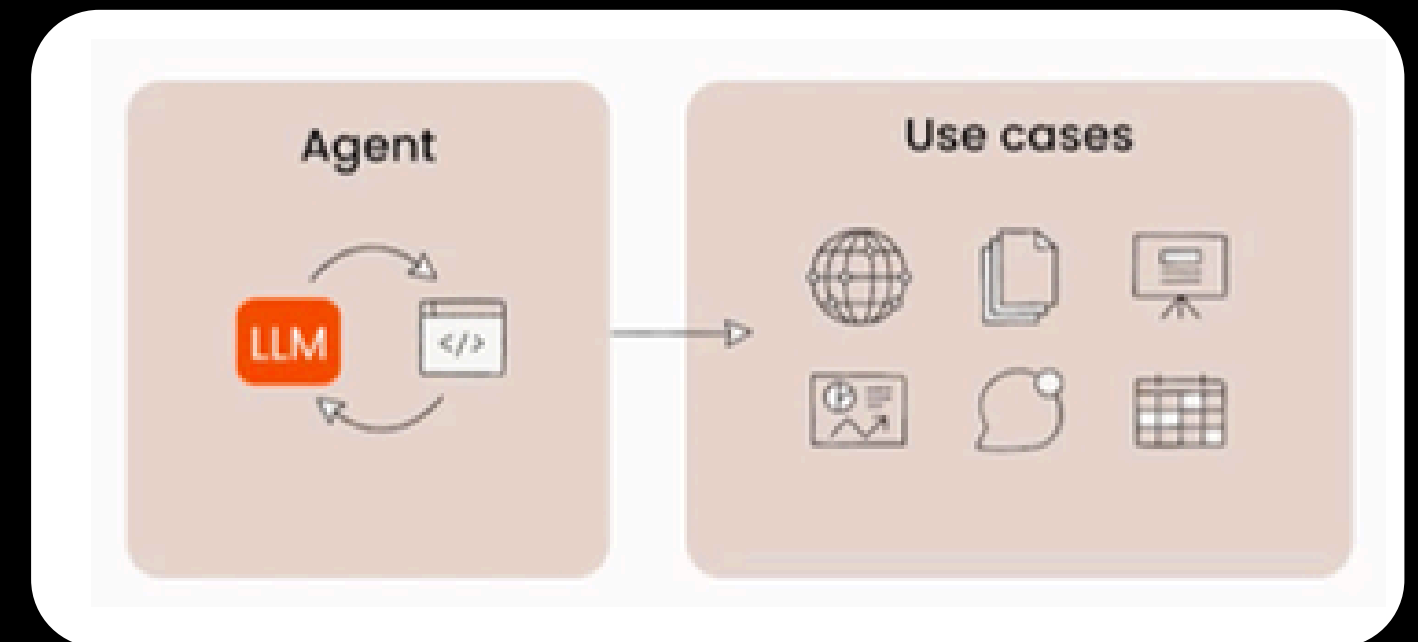
Single-purpose agents



Generalist Agents

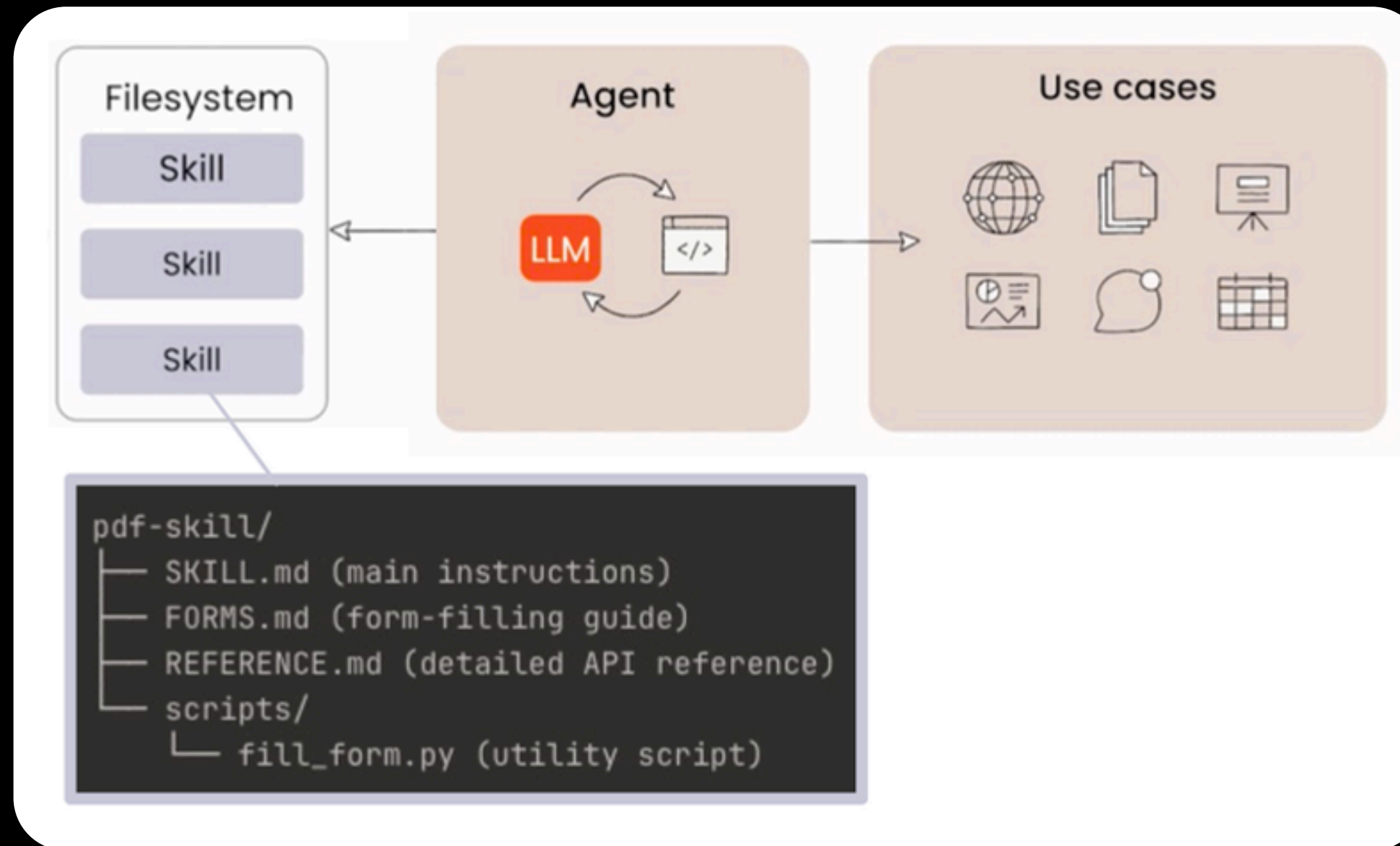
General-purpose Agents

One agent, many use cases



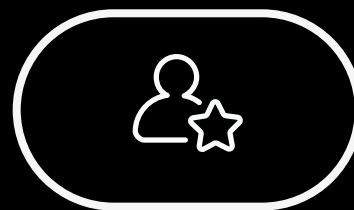
✓ Code as universal interface

General-purpose Agents with Skills

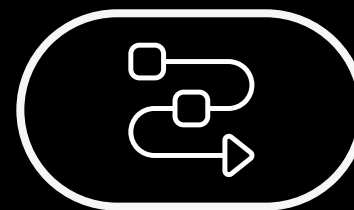


What Are Agent Skills?

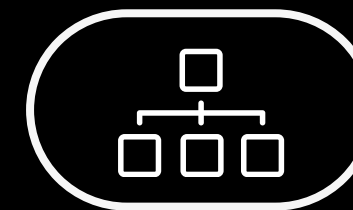
Modular, dynamically-loadable units of procedural knowledge that transform general-purpose agents into domain specialists.



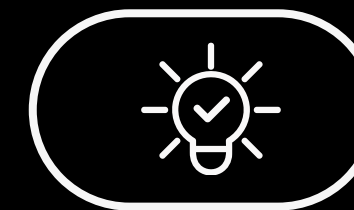
Domain Expertise



Repeatable Workflows



Org-Specific Context



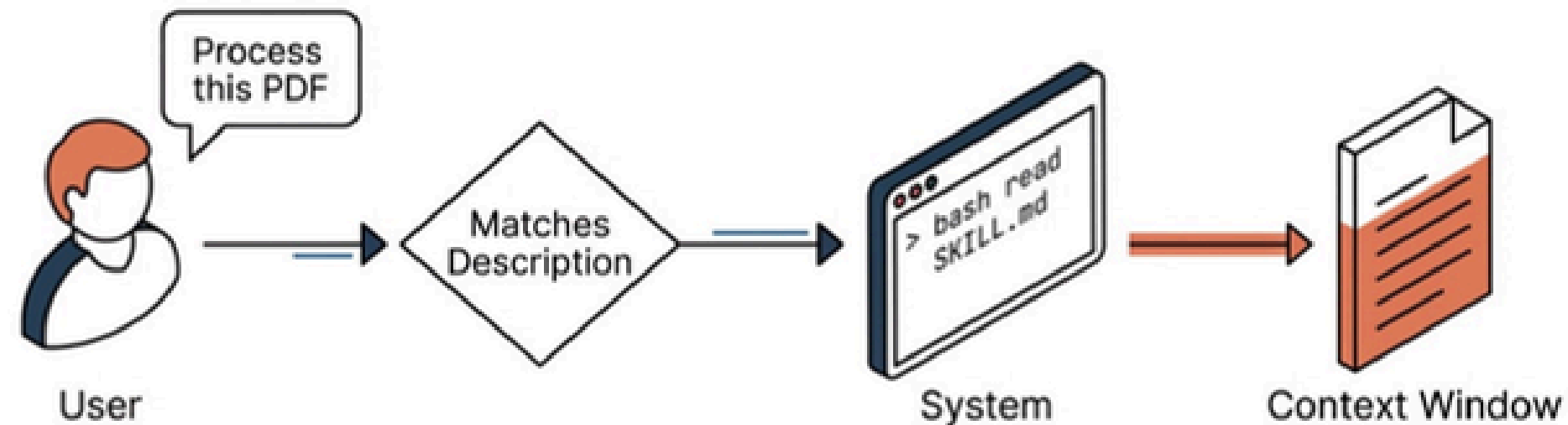
New Capabilities

Skills encode how to do things, not just what to say.
They transform agents from conversational to operational systems.

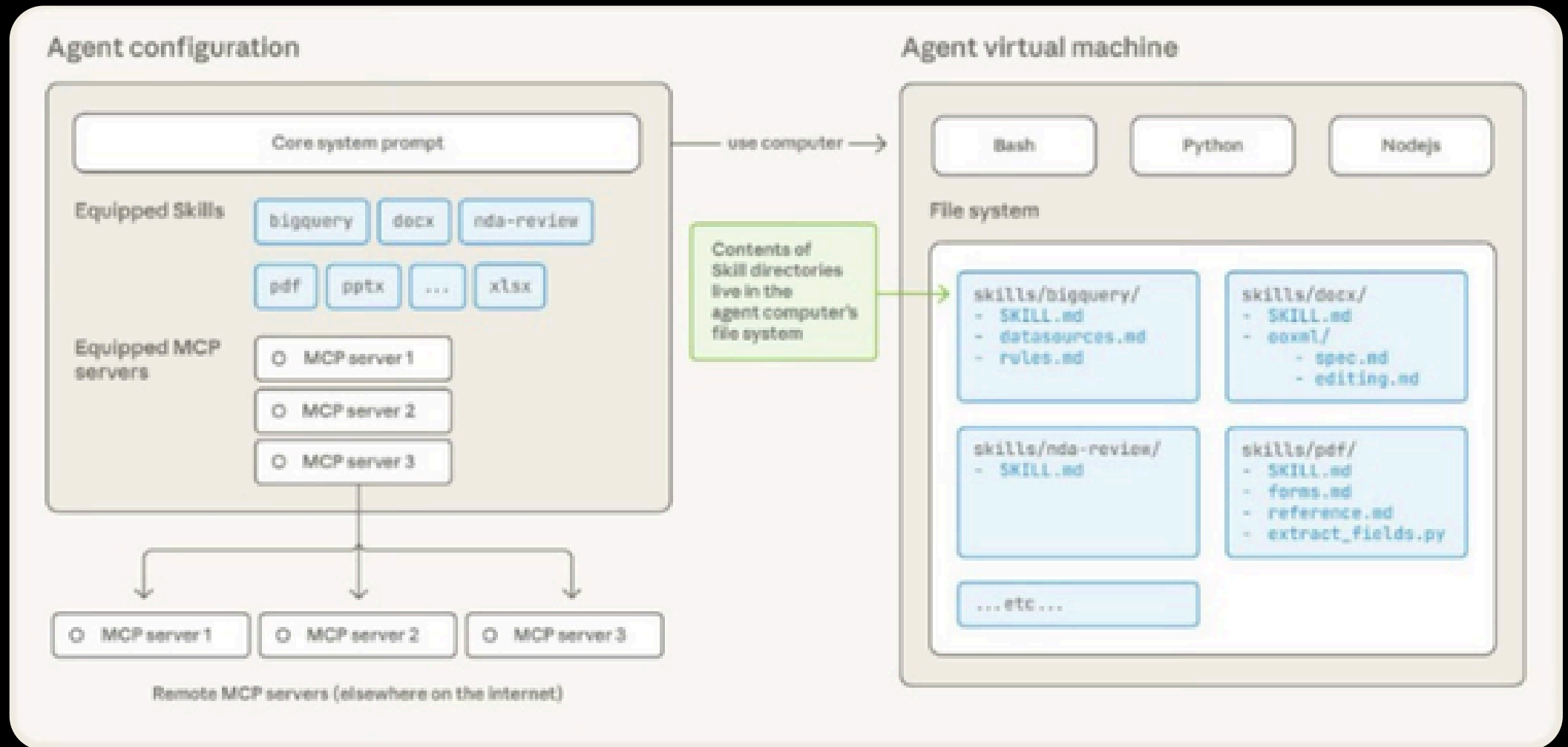
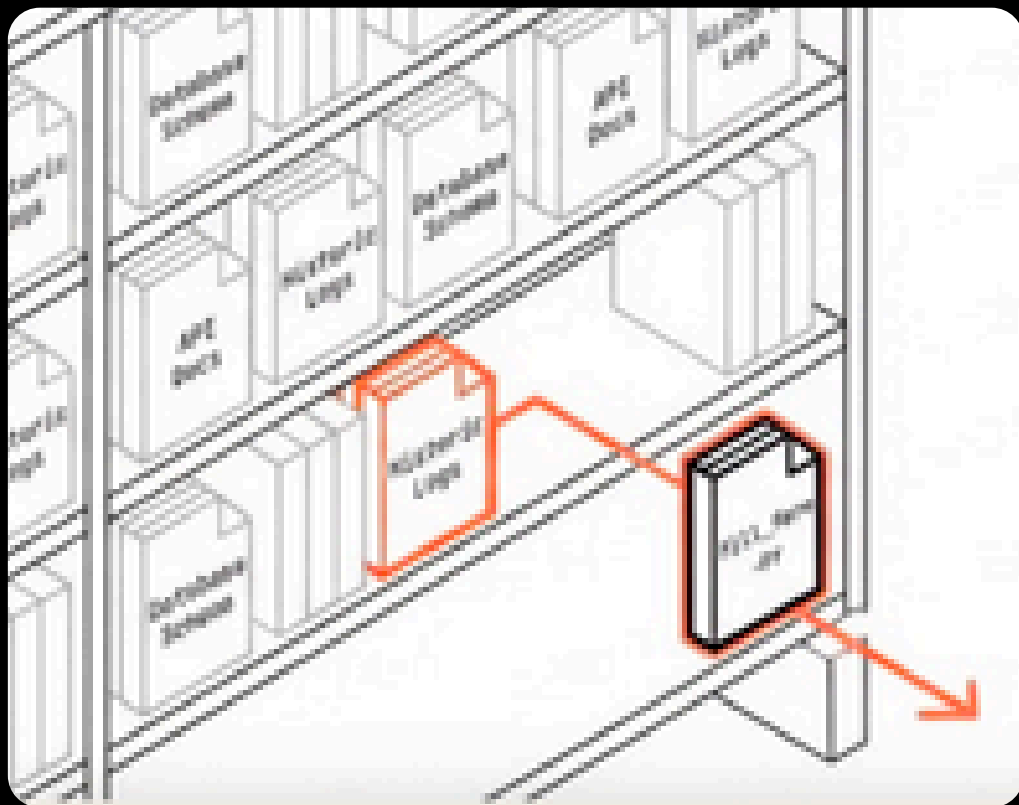
Skill Structure

```
---  
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]
```

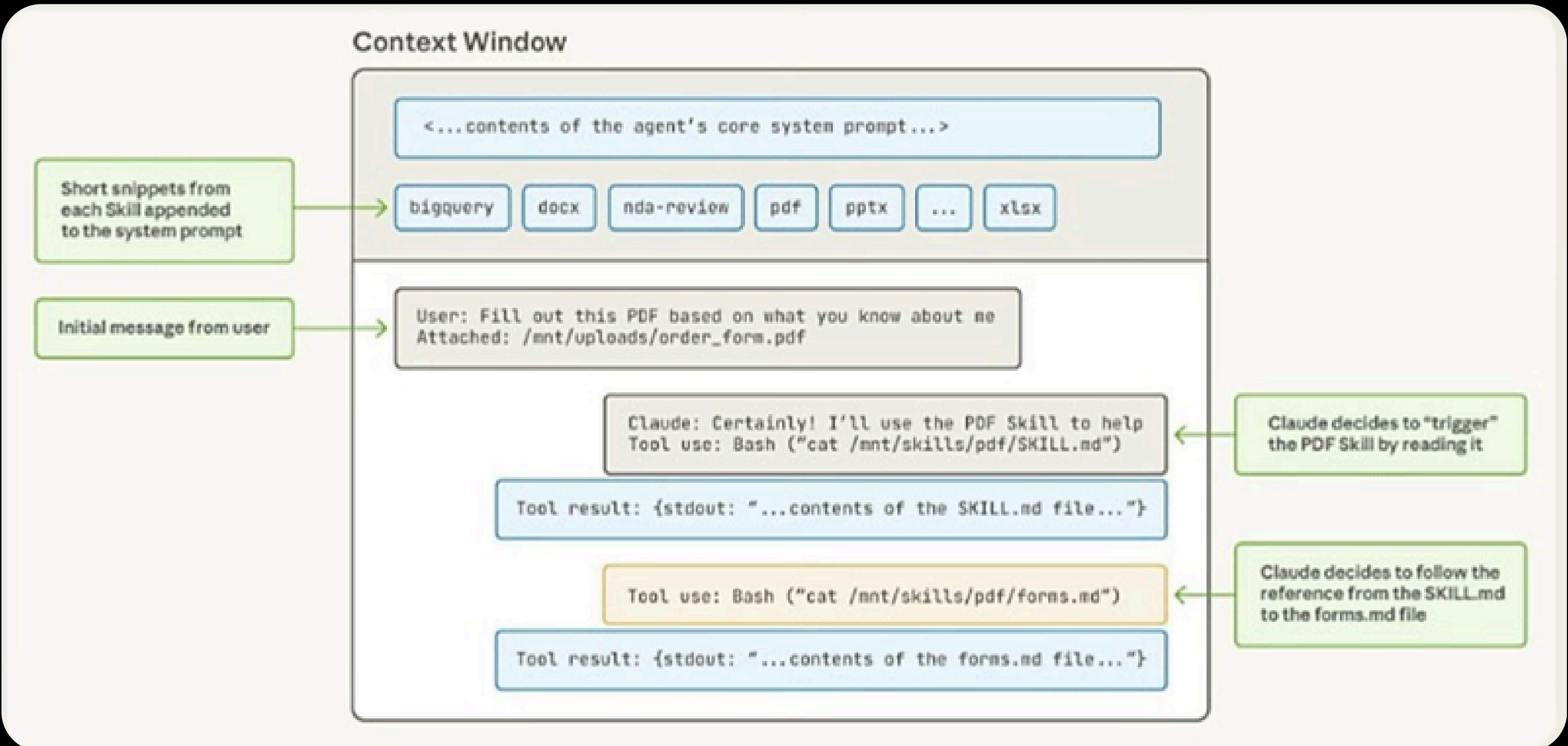
This text triggers
the skill loading



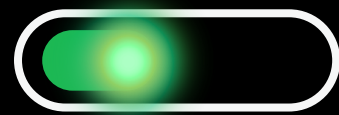
“Unlimited” Resource Library



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
weatherdependent 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

Four design patterns

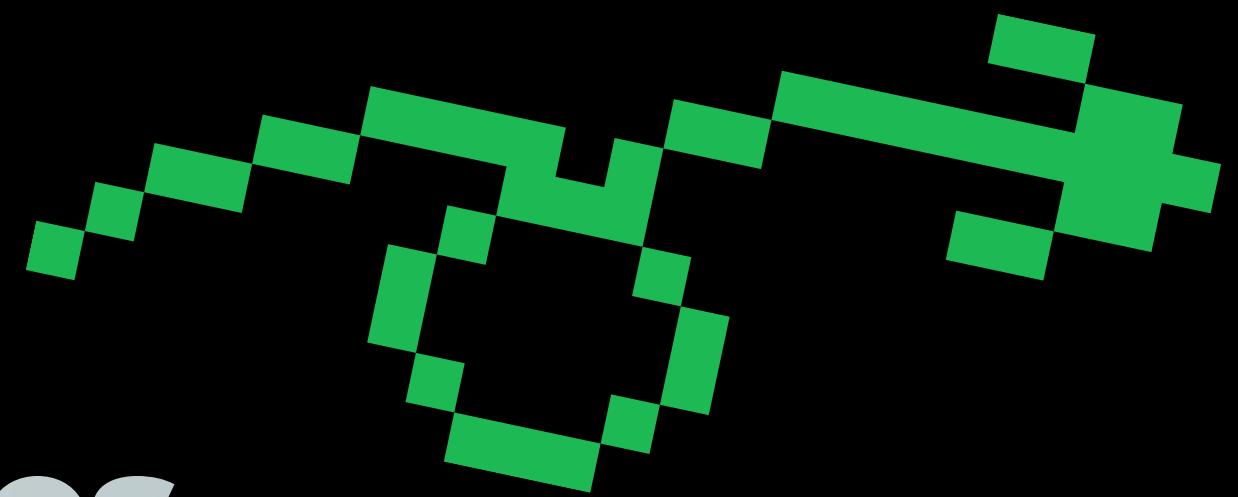
enable sophisticated workflows:
Reflection,
Tool Use,
Planning,
Multi-Agent

Start simple

and add complexity gradually

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

wxrks