Strands Agent Design Patterns

Version 0.1.x | Generated: 2025-06-18

# Pattern 1: Research Agent

The Research Agent pattern combines web search, document analysis, and fact-checking capabilities to provide comprehensive research assistance.

## Implementation

Key components:

* • Web search tool for finding information
* • Document parser for analyzing sources
* • Fact-checking tool for verification
* • Citation manager for references

### Code Example

from strands import Agent, tool  
  
@tool  
def search\_web(query: str) -> List[str]:  
 """Search the web for information."""  
 # Implementation here  
 pass  
  
research\_agent = Agent(  
 system\_prompt="You are a research assistant.",  
 tools=[search\_web, parse\_document, check\_facts]  
)

# Pattern 2: Customer Service Agent

The Customer Service Agent pattern handles user inquiries, accesses knowledge bases, and escalates complex issues.

## Key Features

* • FAQ search capability
* • Order status lookup
* • Ticket creation system
* • Sentiment analysis
* • Escalation workflow

## Best Practices

1. Always maintain a professional tone  
2. Search knowledge base before responding  
3. Collect necessary information before escalating  
4. Provide clear next steps to users  
5. Log all interactions for quality assurance

# Pattern 3: Code Assistant

The Code Assistant pattern helps developers write, debug, and optimize code across multiple programming languages.

## Core Components

* • Code analysis tool
* • Syntax checker
* • Test generator
* • Documentation writer
* • Performance profiler

## Implementation Strategy

The Code Assistant should be able to:  
1. Understand code context and requirements  
2. Generate code following best practices  
3. Explain complex code segments  
4. Suggest optimizations and improvements  
5. Help with debugging and error resolution