

Building AI Agents in Kotlin with Koog

Maria Tigina

Set up the project

Setting up the project

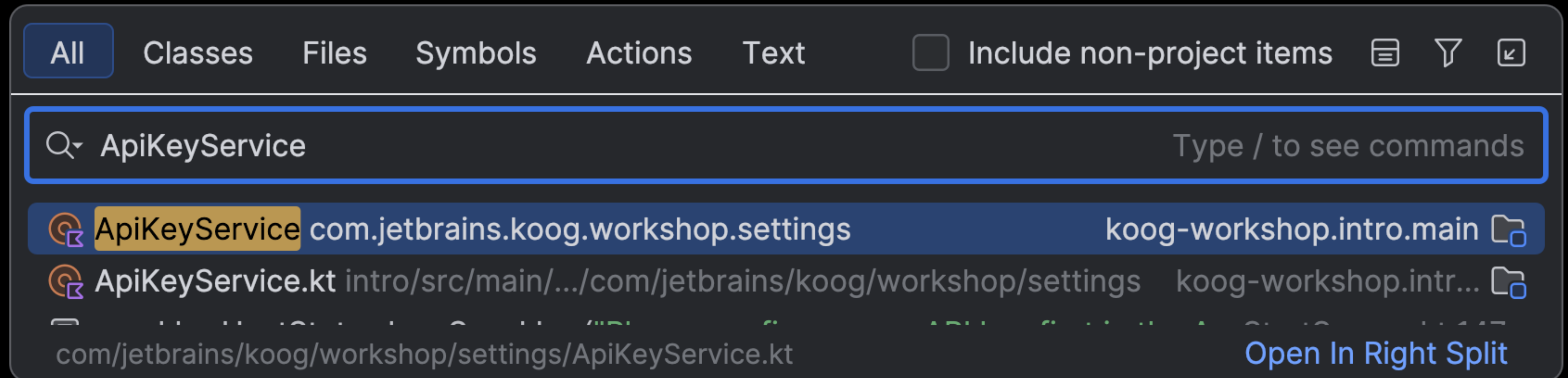
Clone <https://github.com/nomisRev/koog-workshop>

Set up API keys environment variables:

- KOOG_WORKSHOP_ANTHROPIC_API_KEY
- KOOG_WORKSHOP_OPENAI_API_KEY



shift
shift





← Weather Forecast

Debug:



View ▾

Hi, I'm a weather agent. I can provide weather forecasts for any location.



What's the weather in Munich today?



I don't have real-time data, but I can give you an idea of the typical weather in Munich around this time of year, **early November**. It tends to be cool, with average temperatures ranging from 3°C to 10°C (37°F to 50°F). You can expect cloudy days with occasional rain, so it's a good idea to bring a warm coat and an umbrella. For the most accurate and up-to-date weather, please check a reliable weather website or app.



Type a message...



← Weather Forecast

Debug: ☐ View ▾

Hi, I'm a weather agent. I can provide weather forecasts for any location.



What's the weather in Munich today?



I don't have access to real-time weather data, so I can't tell you the current weather conditions in Munich today.

To get accurate, up-to-date weather information for Munich, I'd recommend:

1. **Weather websites:** Check [weather.com](https://www.weather.com), [AccuWeather](https://www.accuweather.com), or [weather.gov](https://www.weather.gov)
2. **Weather apps:** Use your phone's built-in weather app or download apps like Weather Underground or Dark Sky
3. **Search engines:** Simply search "Munich weather" in Google or Bing
4. **Local sources:** Check Deutscher Wetterdienst (DWD), Germany's national weather service

These sources will give you current conditions, hourly forecasts, and extended outlooks for Munich. Is there anything specific about Munich's weather patterns or climate that I can help you understand instead?



Type a message...



What is AI Agent

What is AI agent?

You are a banking assistant with access to banking operations.

You can send money by writing ONLY the following JSON:

```
{  
  "name": "send_money",  
  "params": {  
    "to": <recepient id>,  
    "amount": <amount to be sent>,  
  }  
}
```

Understood. I'm ready to assist with sending money.

Ask anything



Search



Reason



Deep research



Create image



What is AI agent?

You are a banking assistant with access to banking operations.

You can send money by writing ONLY the following JSON:

```
{
  "name": "send_money",
  "params": {
    "to": <recepient id>,
    "amount": <amount to be sent>,
  }
}
```

And you can get list of my contacts using:

```
{
  "name": "list_contacts",
}
```

I will provide them in the format of:

```
[
  {
    "name": "Daniel",
    "id": "daniel_kraig_007"
  },
  ...
]
```

Got it! I'm ready to help you with banking operations like sending money or listing your contacts. Just let me know what you'd like to do.

What is AI agent?

Send 100\$ to my **mom**

To proceed, I need to list your contacts to identify your wife's recipient ID.

Sending the following to get your contacts:

json

Copy

Edit

```
{  
  "name": "list_contacts"  
}
```

```
[  
  {  
    "name": "mom",  
    "id": "aleksandra_12345"  
  },  
  {  
    "name": "Daniel",  
    "id": "daniel_kraig_007"  
  }  
]
```



What is AI agent?

```
"name": "mom",
"id": "aleksandra_12345"
},
{
  "name": "Daniel",
  "id": "daniel_kraig_007"
}
]
```

Thanks! Now sending \$100 to your **mom**

json

 Copy

 Edit

```
{
  "name": "send_money",
  "params": {
    "to": "aleksandra_12345",
    "amount": 100
  }
}
```

Ask anything



Search



Reason



Deep research



Create image



AI Tool

**Any function the model
can decide to call**

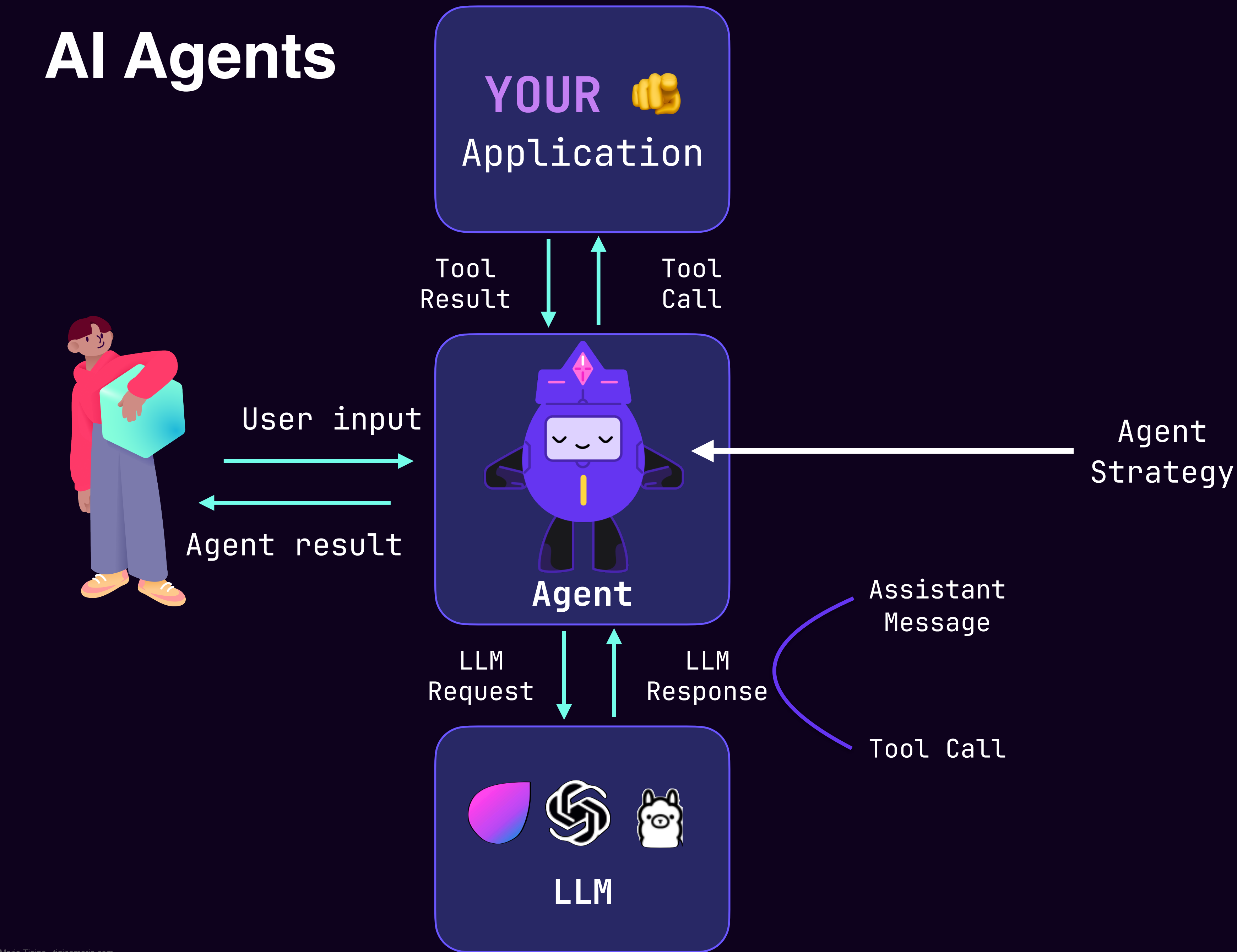
**LLM becomes “the brain”.
It “decides” what tools to
call and when to accomplish
the user’s request**

AI Agent

**A model using tools
to solve user's problem**

Basic AI Agent

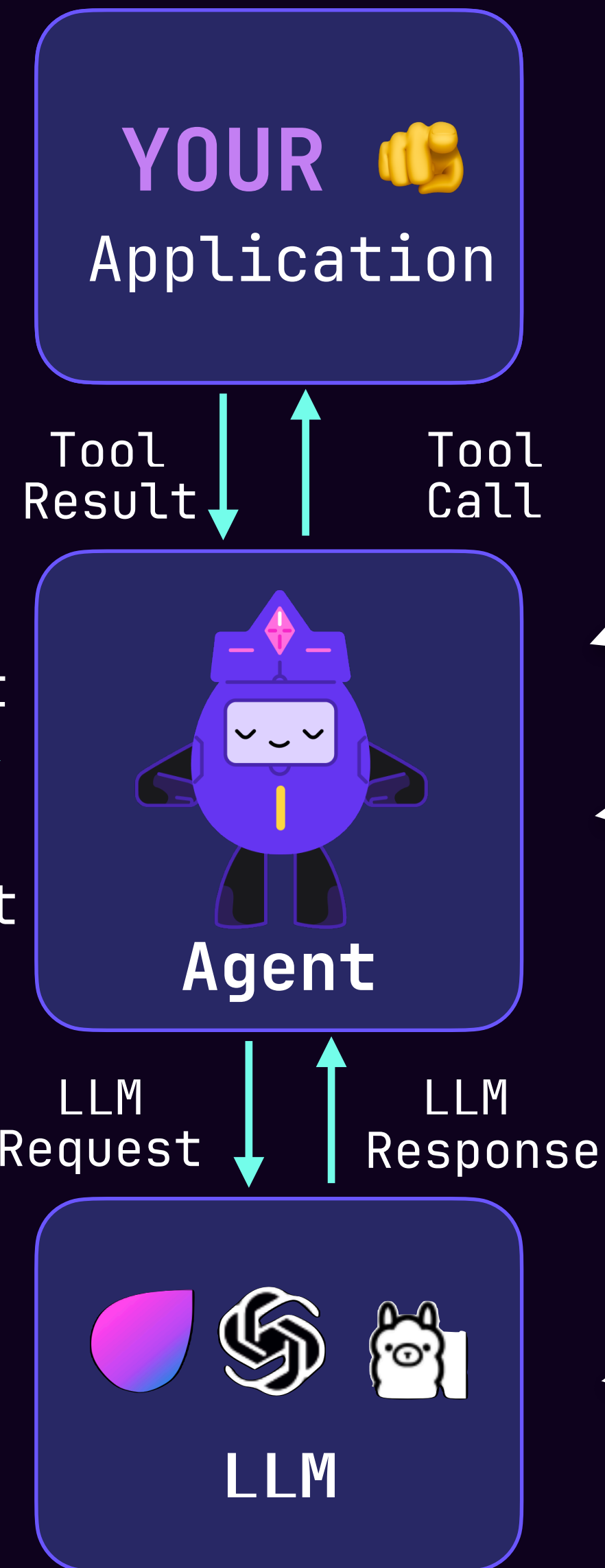
AI Agents



Koog AI Agent



User input
Agent result



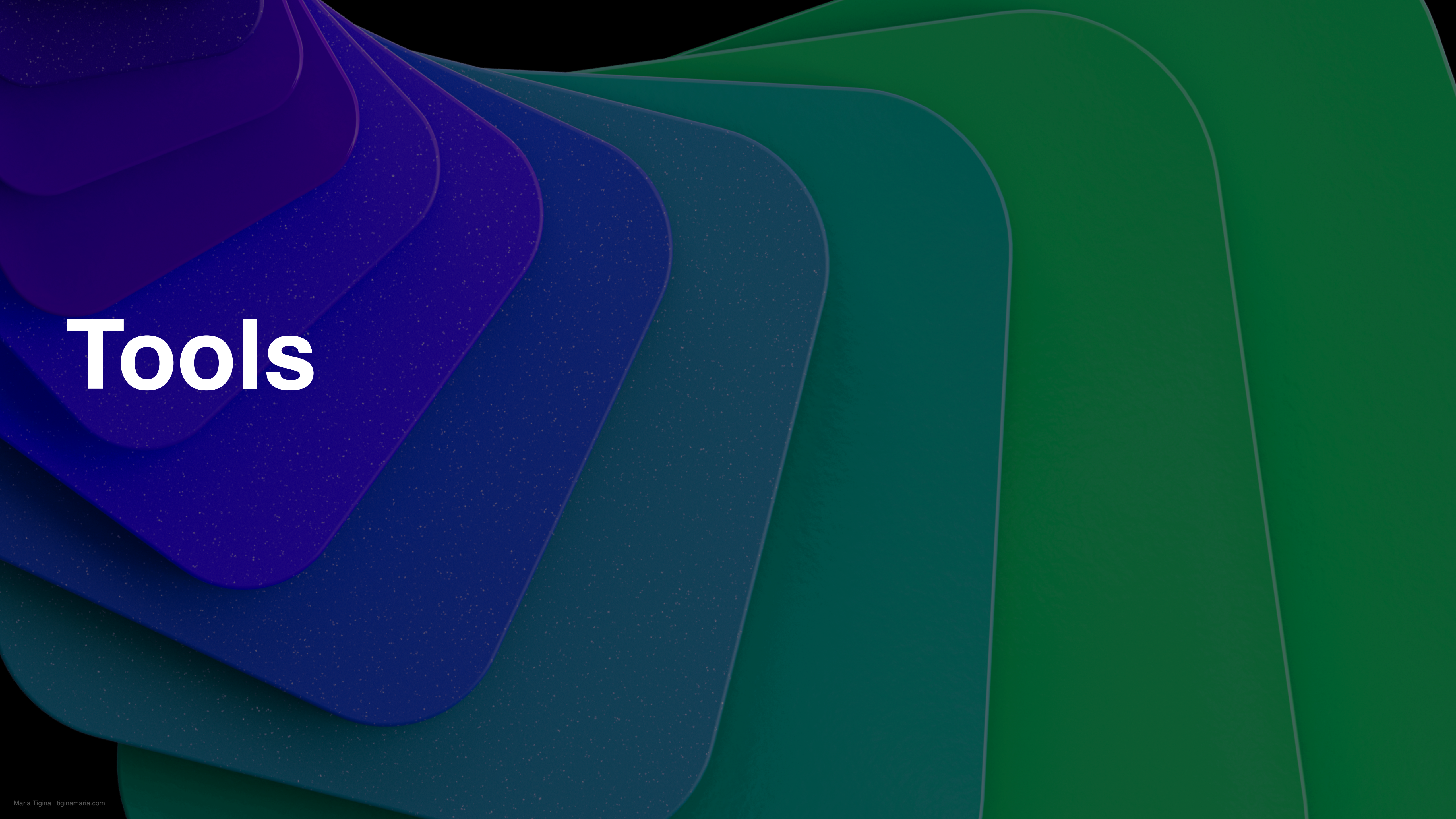
```
val agent = AIAgent(
```

```
    toolRegistry = ToolRegistry {  
        tools(BankingTools())  
    },
```

```
    strategy = strategy<String, String>("my-strategy") {},
```

```
    agentConfig = AIAgentConfig(  
        model = OpenAIModels.Chat.GPT5_4Pro,  
        prompt = prompt("id") {  
            system("You are a helpful assistant")  
        },  
        maxAgentIterations = 100  
    ),
```

```
    promptExecutor = MultiLLMPromptExecutor(  
        OpenAILLMClient("OPENAI_KEY"),  
        AnthropicLLMClient("ANTHROPIC_KEY"),  
    )  
)
```

Tools

Developing tools

Tools = Kotlin
methods, AI agent can
call

- You define methods
- LLM constructs arguments
- You control outputs + logic
- LLM controls the order

(if no strategy provided)

Developing tools

```
fun sendEmail(  
    recipientEmail: String,  
    text: String  
): String {...}
```


Developing tools

```
@Tool
@LLMDescription("Sends email and waits for reply")
fun sendEmail(
    @LLMDescription("Email of the recipient")
    recipientEmail: String,
    @LLMDescription("Content of the email message")
    text: String
): String {...}
```


More tools:

```
class AccountReadTools(private val userId: String) : ToolSet {

    @Tool
    @LLMDescription("Returns a list of transactions for the current user")
    fun getLatestTransactions(
        startDate: Instant?,
        @LLMDescription("Optional transaction status filter (if null -- all statuses will be returned)")
        status: Transaction.Status? // ? = null ==>>> all statuses
    ): List<Transaction> { ... }

    @Tool
    @LLMDescription("Get account balance (in USD) for the current user")
    fun getAccountBalance(): @LLMDescription("Get account balance (in USD) for the current user") Int { ... }

    @Tool
    @LLMDescription("Get account information for the current user: name, credentials, balance, etc.")
    fun readAccountInfo(): UserAccountInfo { ... }

}
```


Tools: passing to an agent

```
val bankingTools = ToolRegistry {  
    tool(::sendMoney)  
    tools(BankingTools(userID))  
}
```

```
val agent = AIAgent(  
    promptExecutor = executor,  
    llmModel = OpenAIModels.Chat.GPT5,  
    systemPrompt = "You're a banking assistant...",  
    toolRegistry = bankingTools,  
)
```


Coding Time!



← Weather Forecast

Debug:



View ▾

Hi, I'm a weather agent. I can provide weather forecasts for any location.



What's the weather in Munich today?



I don't have real-time data, but I can give you an idea of the typical weather in Munich around this time of year, early November. It tends to be cool, with average temperatures ranging from 3°C to 10°C (37°F to 50°F). You can expect cloudy days with occasional rain, so it's a good idea to bring a warm coat and an umbrella. For the most accurate and up-to-date weather, please check a reliable weather website or app.

Type a message...



⬅ Weather Forecast

Hi, I'm a weather agent. I can provide weather forecasts for any location.

Request to LLM

Messages

System	You are a helpful weather assistant. ...	
User	What's the weather in Munich today?	

Available tools

add_datetime(date, days, hours, minutes)current_datetime(timezone)weather_forecast(location, date, days, granularity)

Tool Call

current_datetimetimezone: "Europe/Berlin"

Request to LLM

Messages

System	You are a helpful weather assistant. ...	
User	What's the weather in Munich today?	
ToolCall	current_datetime	{"timezone":"Europe/Berlin"}
Result	current_datetime	"Current datetime: 2026-05-19T20:33:54+02:00, Date: 2026-05-19, Time: 20:33:54, Timezone: Europe/Berlin"

Available tools

add_datetime(date, days, hours, minutes)current_datetime(timezone)weather_forecast(location, date, days, granularity)

Type a message...

➤

Debug: ☐

View ▾

☒ Tools

☒ LLM Calls

☐ Nodes

☒ Tasks

Task. Provide weather tools to the weather agent



The tools are already implemented in WeatherTools.
Add missing `@Tool` and `@LLMDescription` annotations.

Pass these tools to the agent in WeatherAgentProvider.

Run WeatherAgentToolsTest.kt to verify it works.

```
ToolRegistry {  
    tools(WeatherTools())  
}
```




← Weather Forecast

Debug:



View ▾

Hi, I'm a weather agent. I can provide weather forecasts for any location.



What's the weather in Munich today?



The weather in Munich today is rainy, with temperatures ranging from 4.2°C to 18.3°C. There is a slight precipitation expected with about 0.1 mm of rain. Make sure to carry an umbrella if you're heading out!

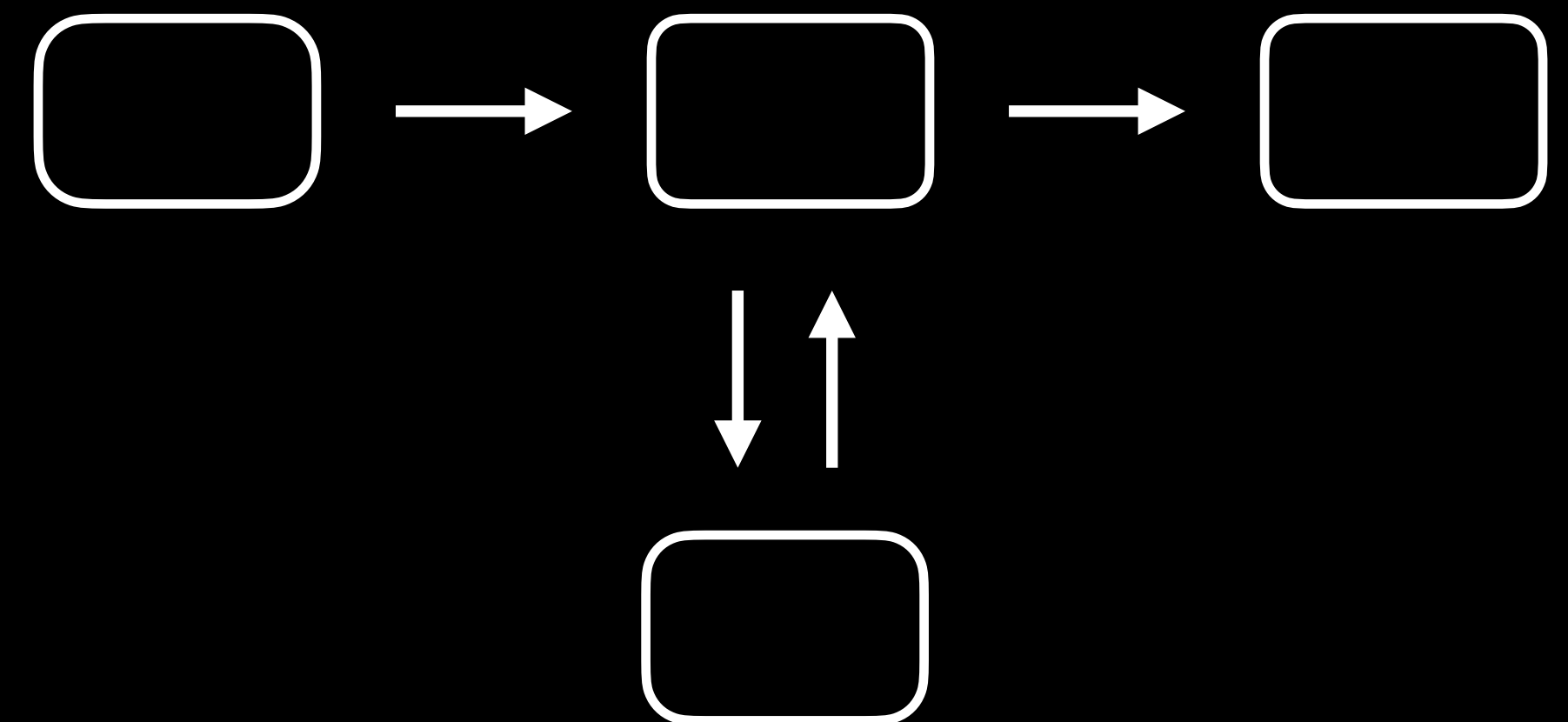
Type a message...



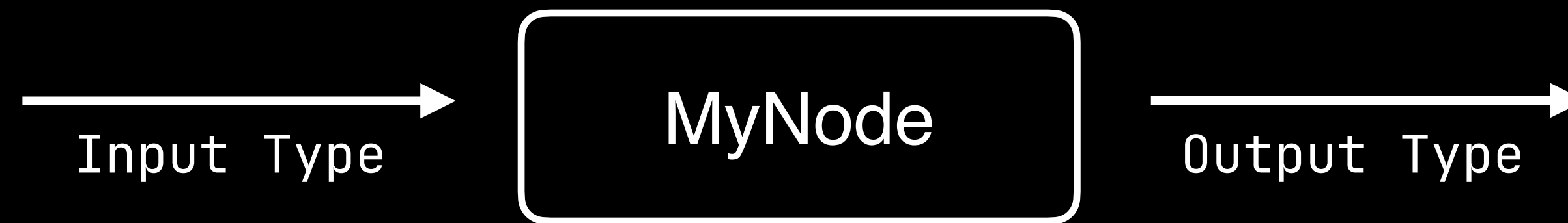
Agent Strategy

Graph Strategy

- Defines how an agent behaves
- Represents the agent's workflow as a directed graph of nodes and edges

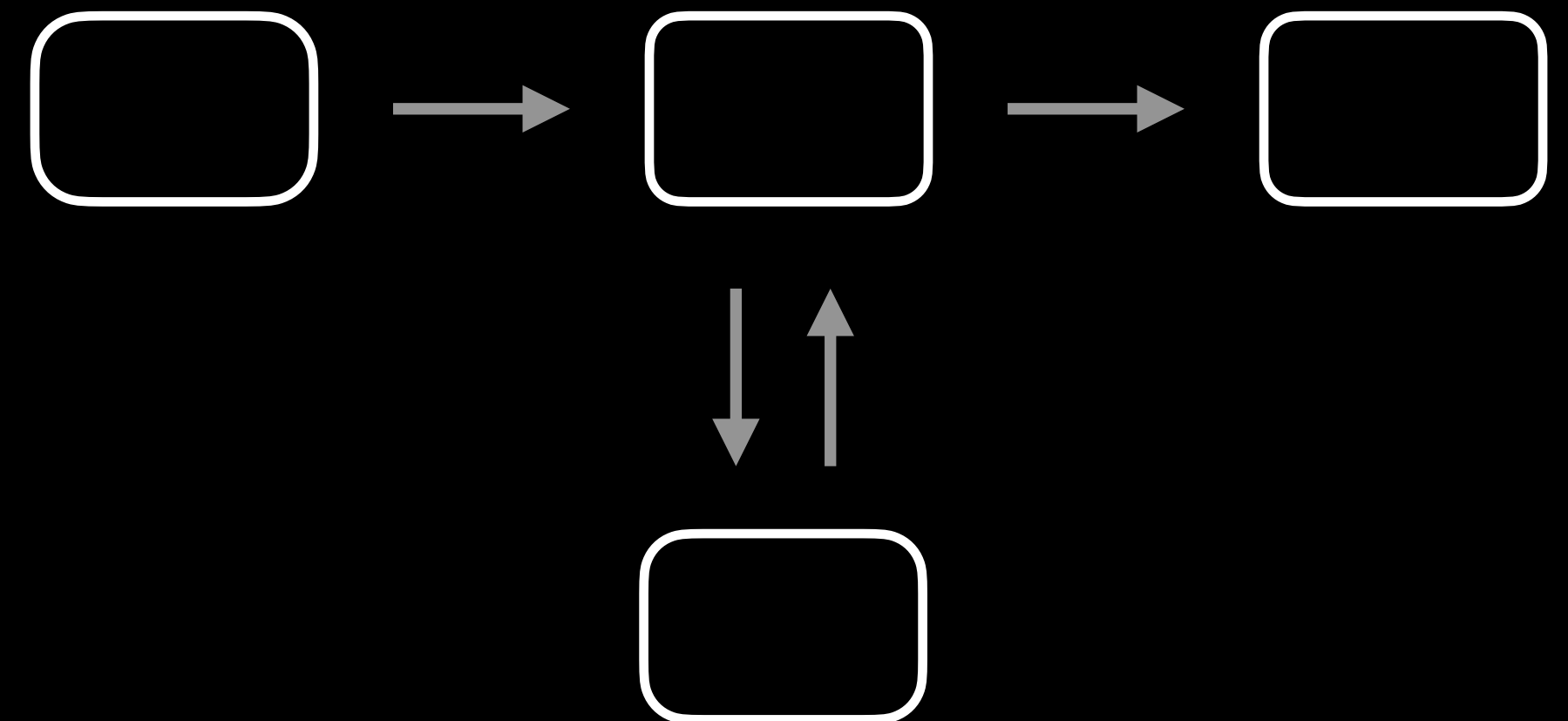


Each node has input and output types



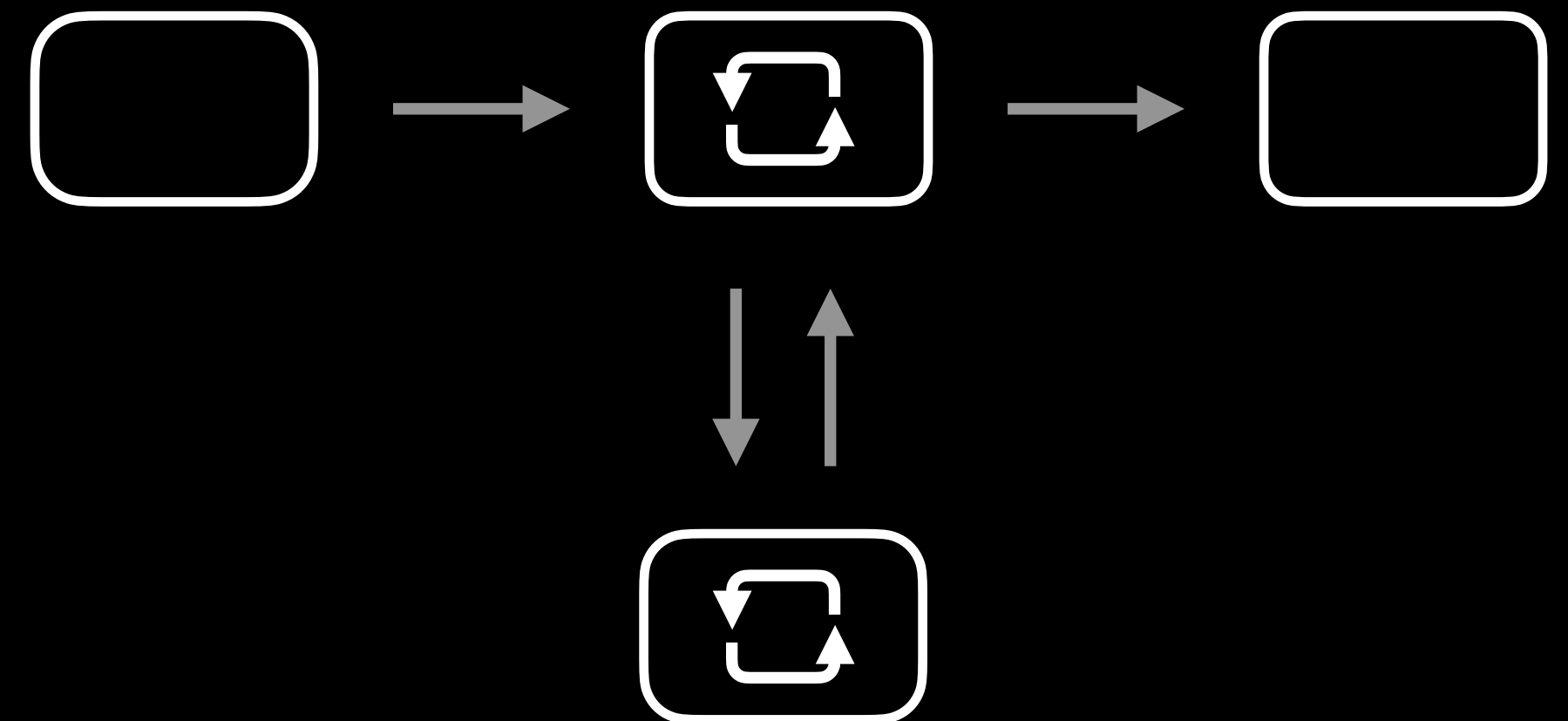
Nodes: low-level

- Calling the LLM
- Executing tools
- Parsing structured output



Nodes: high-level

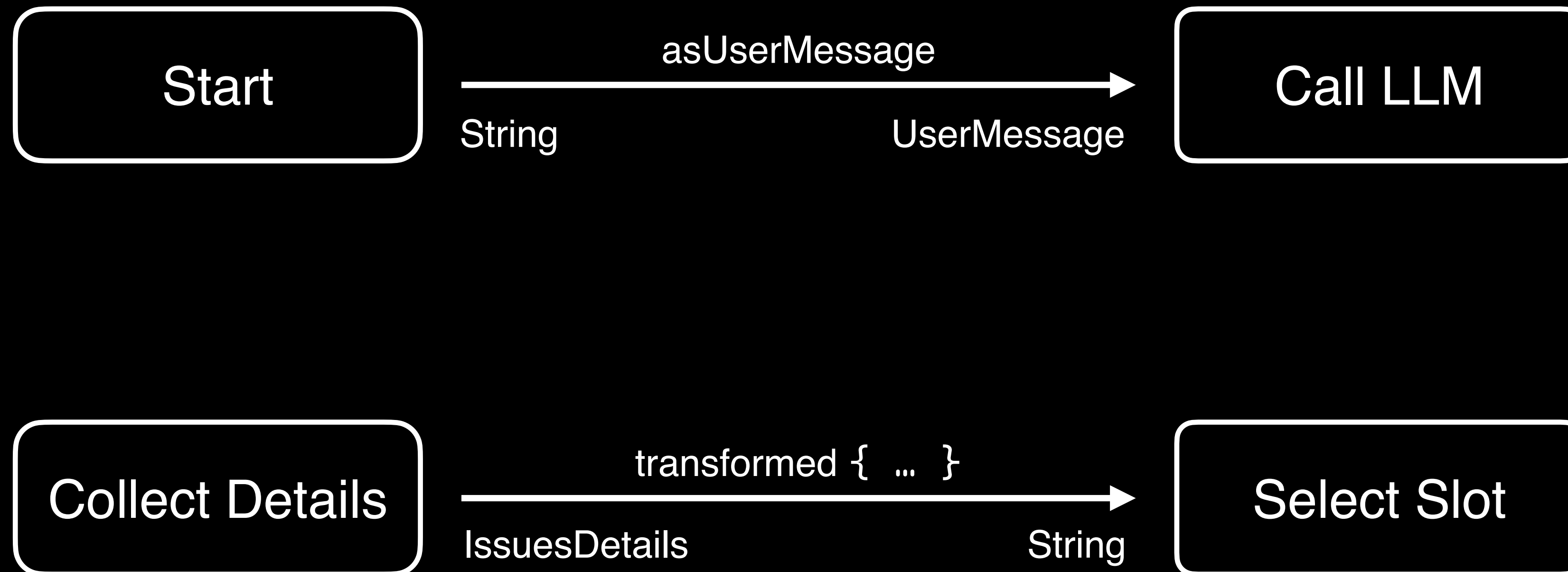
- You can nest an entire strategy in a node using subgraphs
- Enables sub-graph (~sub-agent) composition



Edge: transition between nodes



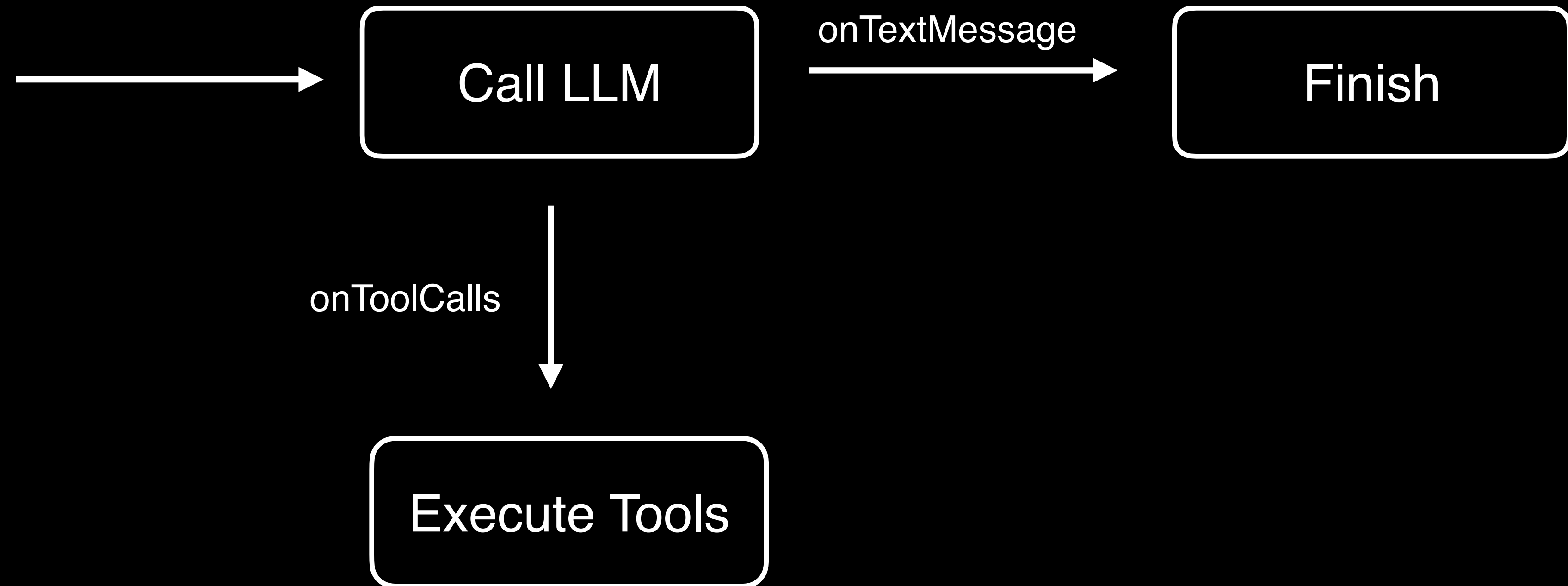
Edge transformations



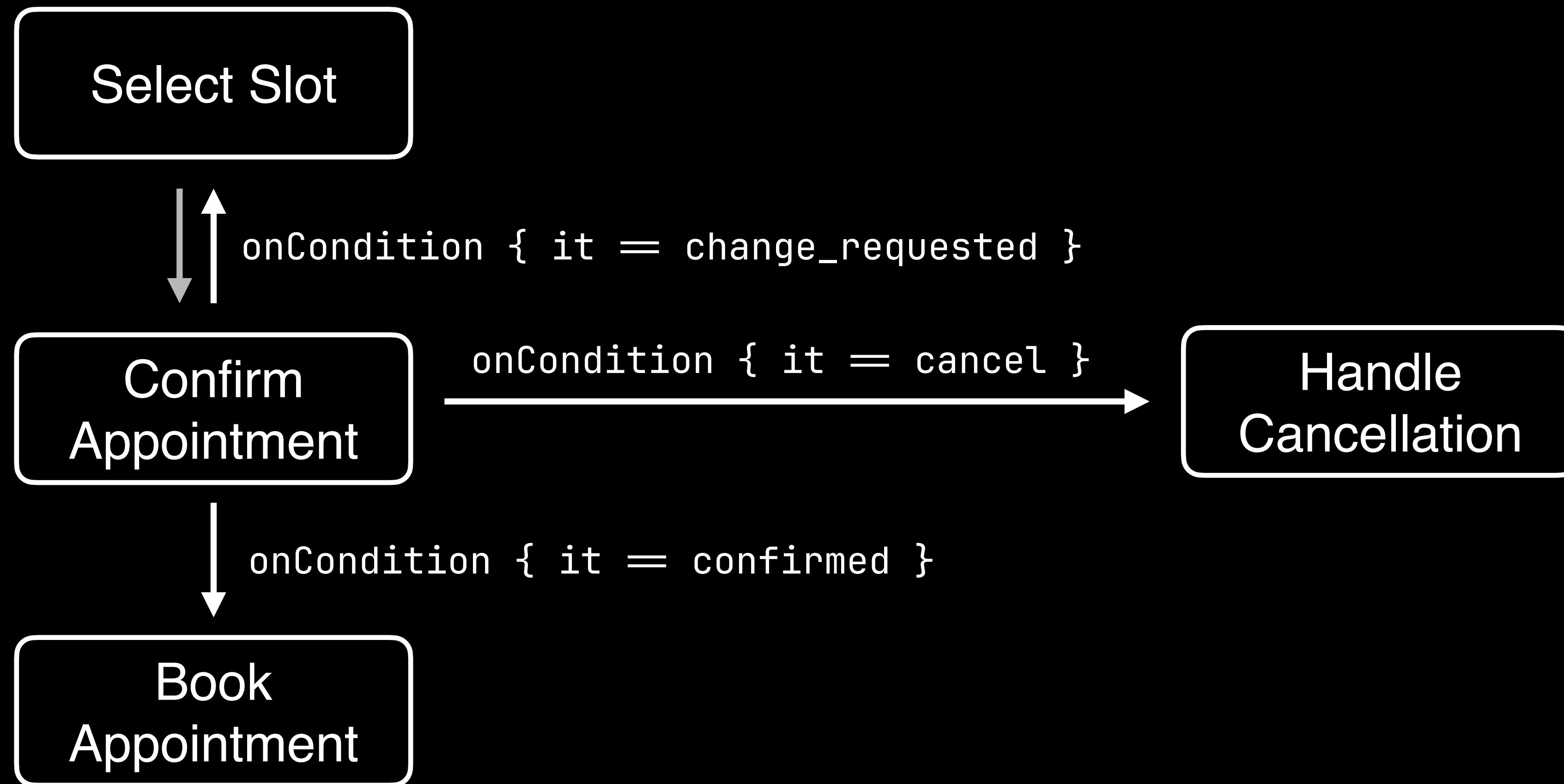
Edge conditions



Edge conditions

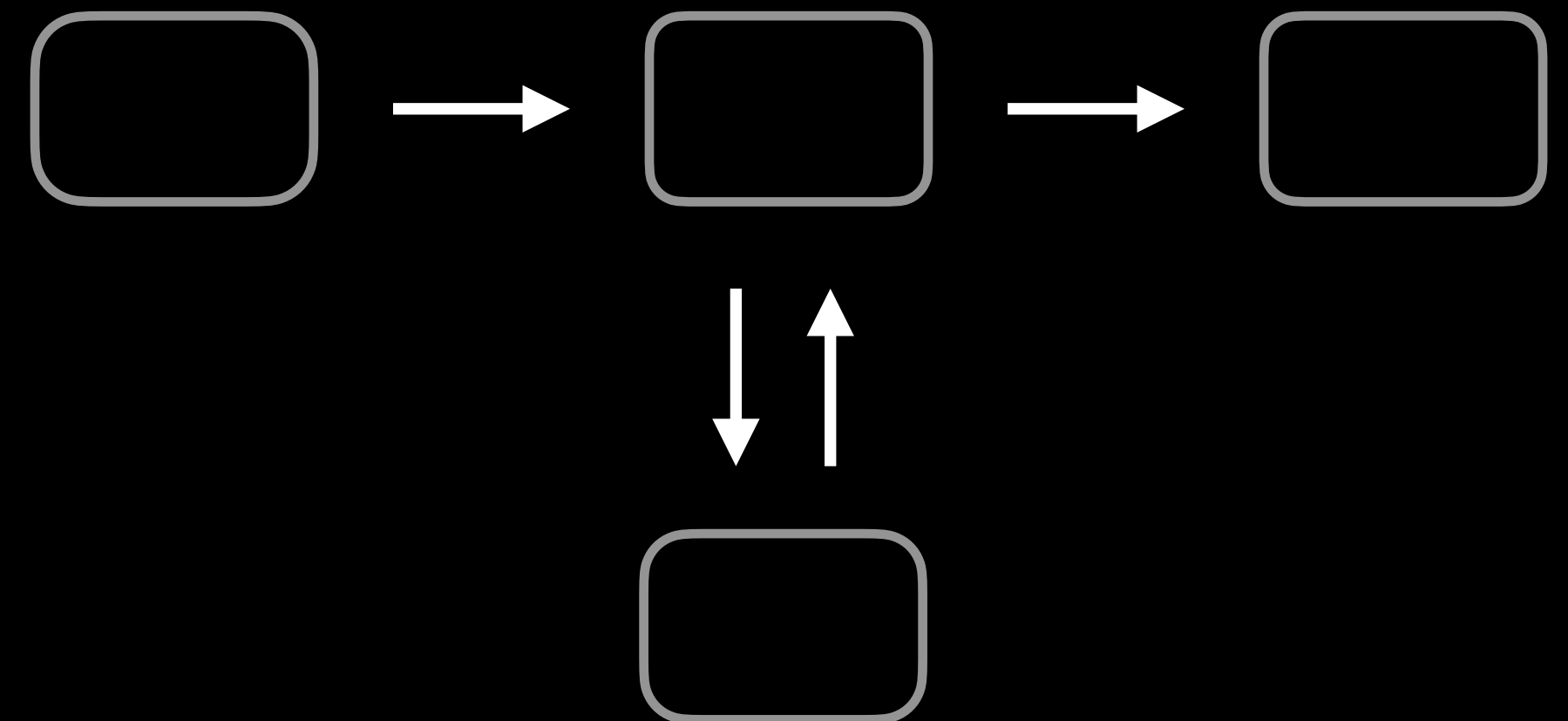


Edge conditions



Edges

- Define transitions between nodes
- Each edge can have:
 - transformations (transformed)
 - conditions (onToolCalls, onTextMessage, onCondition)



Graph nodes: any code

```
val toInt by node<String, Int> { it.toInt() }
```

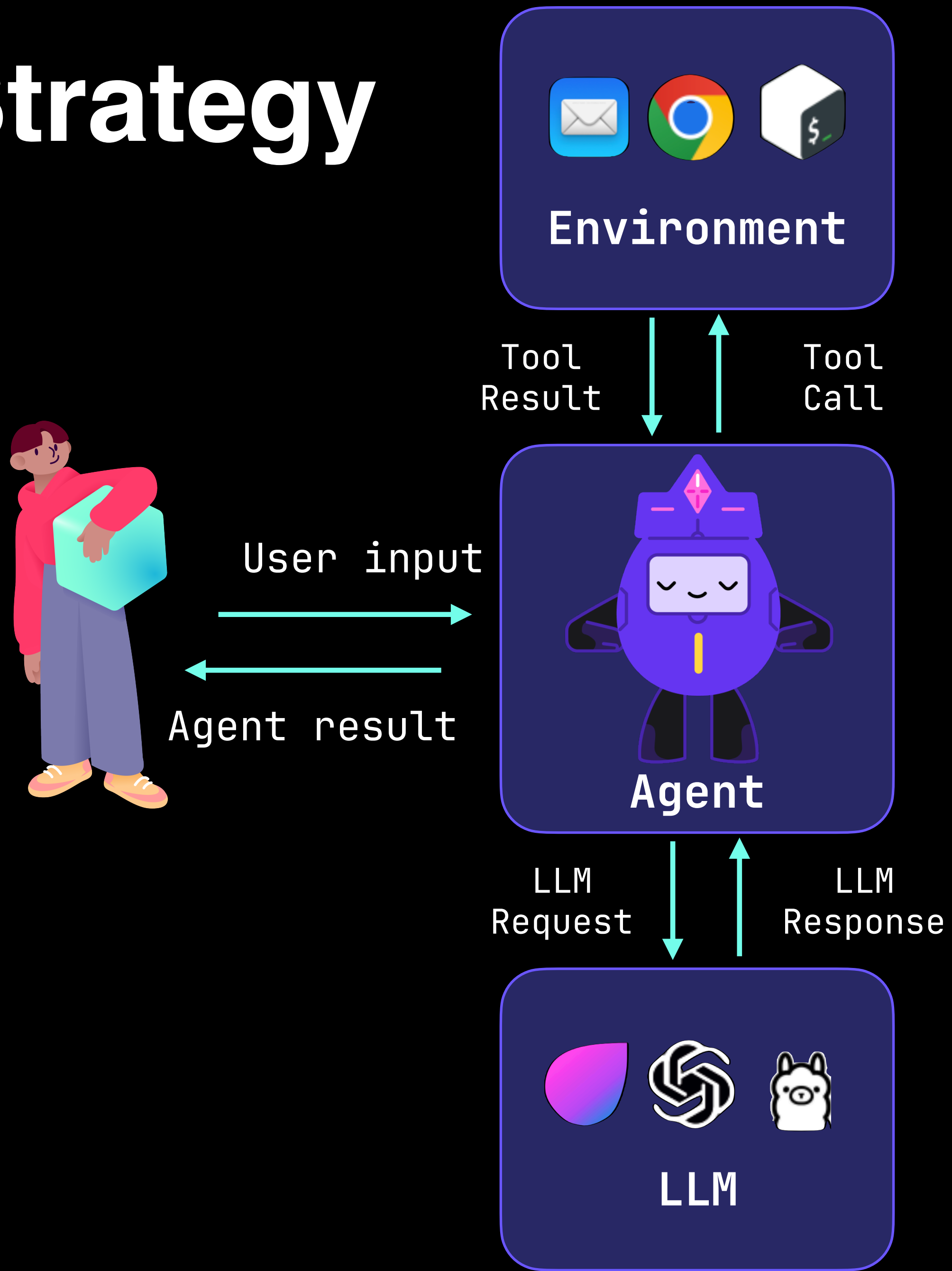
```
val increment by node<Int, Int> { it + 1 }
```

```
edge(nodeStart forwardTo toInt)
```

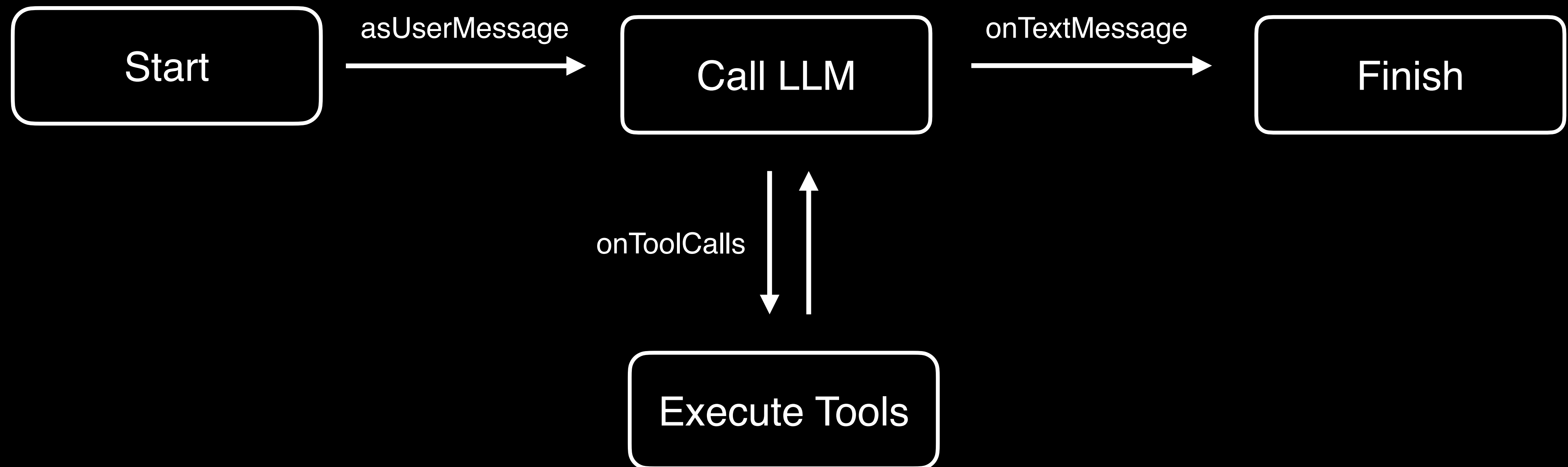
```
edge(toInt forwardTo increment)
```

```
edge(increment forwardTo nodeFinish)
```

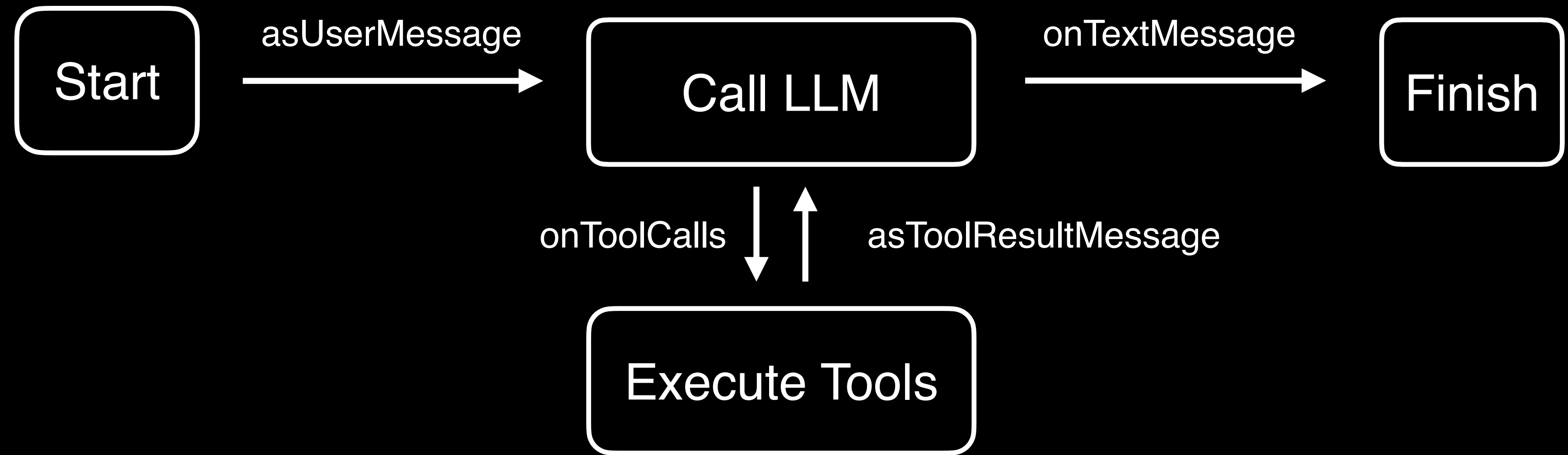

Basic Strategy



Basic Strategy



Basic Strategy



```
// Built-in nodes:
```

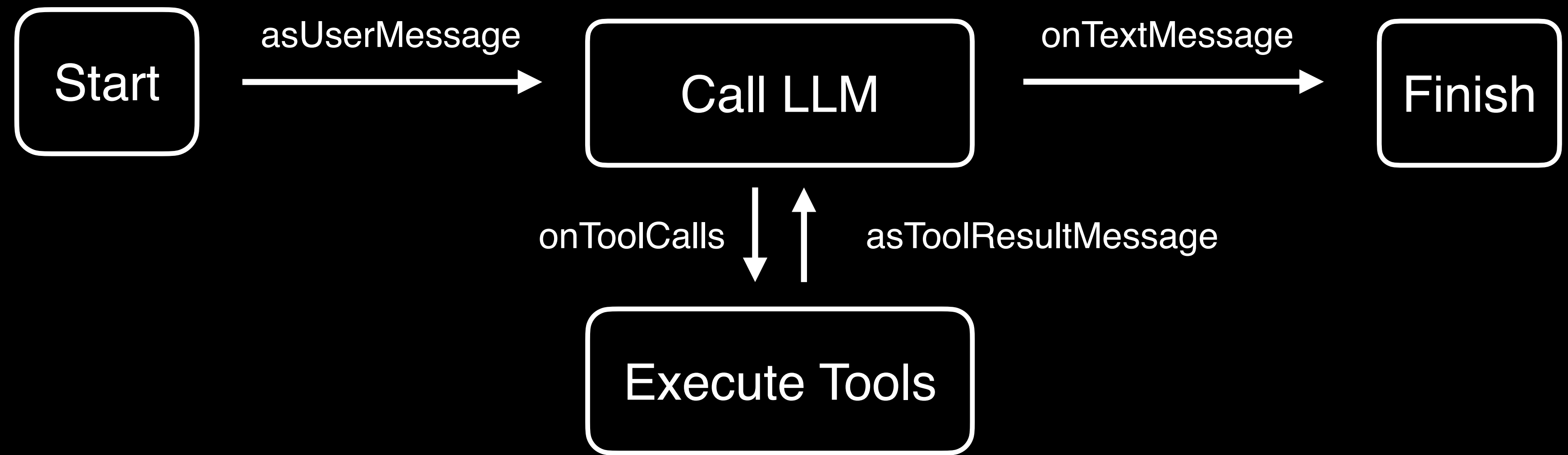
```
nodeStart: StartNode<TInput>
```

```
nodeFinish: FinishNode<TOutput>
```

```
val nodeCallLLM: AIAgentNodeBase<Message.User, Message.Assistant>
```

```
val nodeExecuteTools: AIAgentNodeBase<ToolCalls, ReceivedToolResults>
```


Basic Strategy



```
// Built-in nodes:
```

```
nodeStart: StartNode<String>
```

```
nodeFinish: FinishNode<String>
```

```
val nodeCallLLM: AIAgentNodeBase<Message.User, Message.Assistant>
```

```
val nodeExecuteTools: AIAgentNodeBase<ToolCalls, ReceivedToolResults>
```

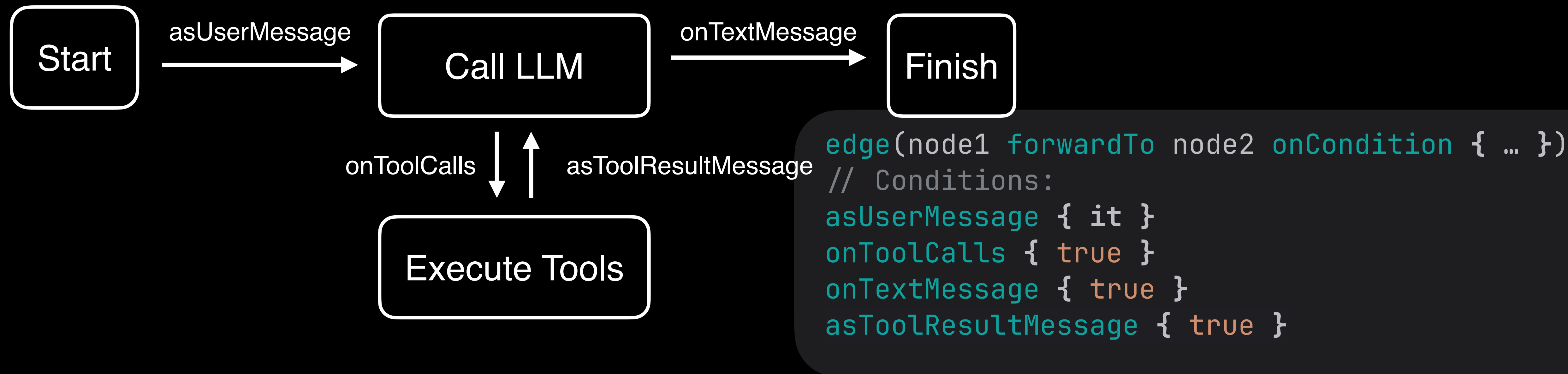

Coding Time!

Task. Implement basic strategy by hand

Open WeatherStrategy and complete its by-hand implementation.

Pass `strategy = basicSingleRunStrategyByHand()` to `WeatherAgent`.

Run WeatherAgentStrategyTest.kt to verify the agent runs as before.



Solution



```
val nodeCallLLM by nodeLLMSendMessage()
val nodeExecuteTools by nodeExecuteTools()

edge(nodeStart forwardTo nodeCallLLM asUserMessage { it })
edge(nodeCallLLM forwardTo nodeExecuteTools onToolCalls { true })
edge(nodeExecuteTools forwardTo nodeCallLLM asToolResultMessage { true })
edge(nodeCallLLM forwardTo nodeFinish onTextMessage { true })
```