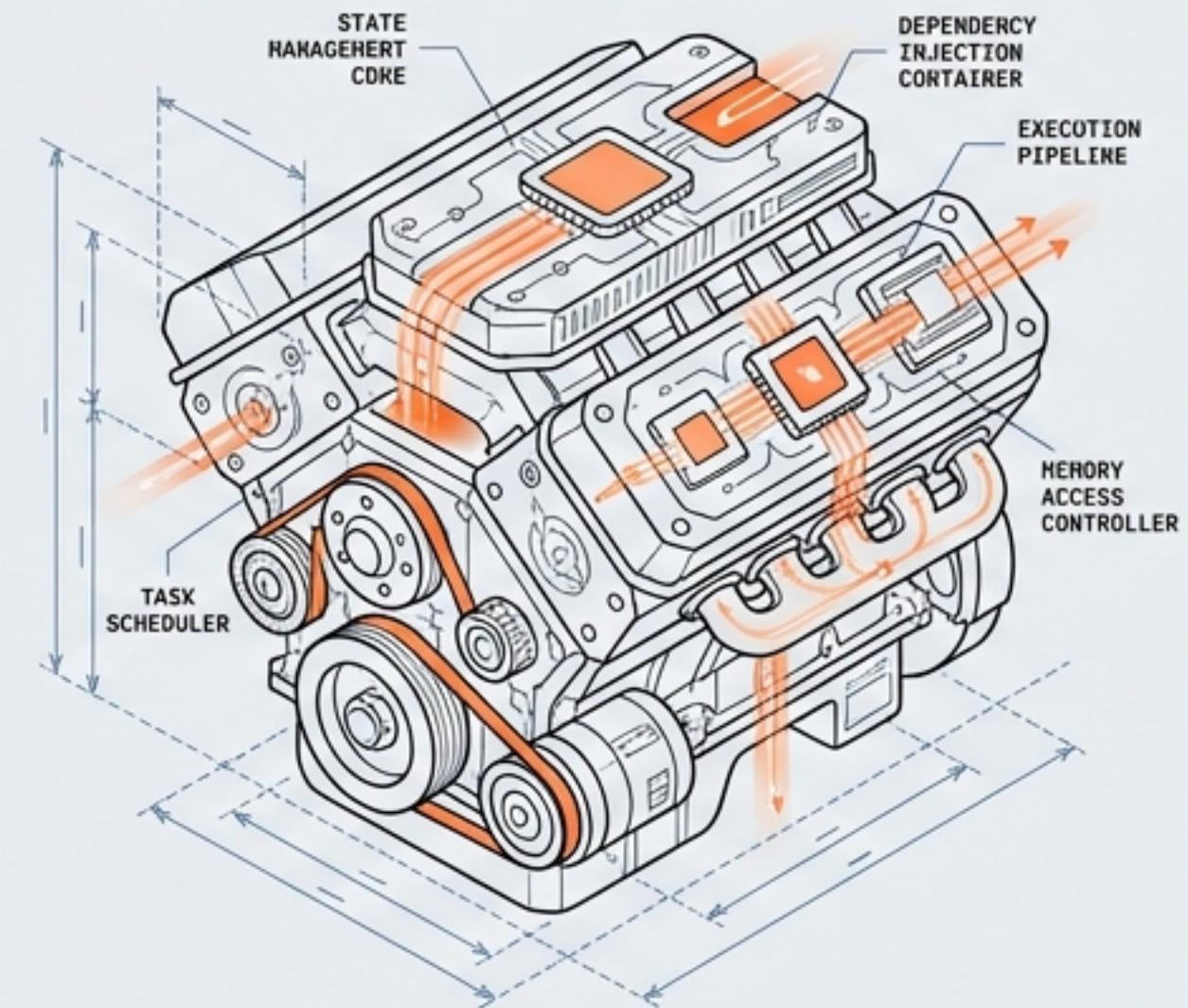


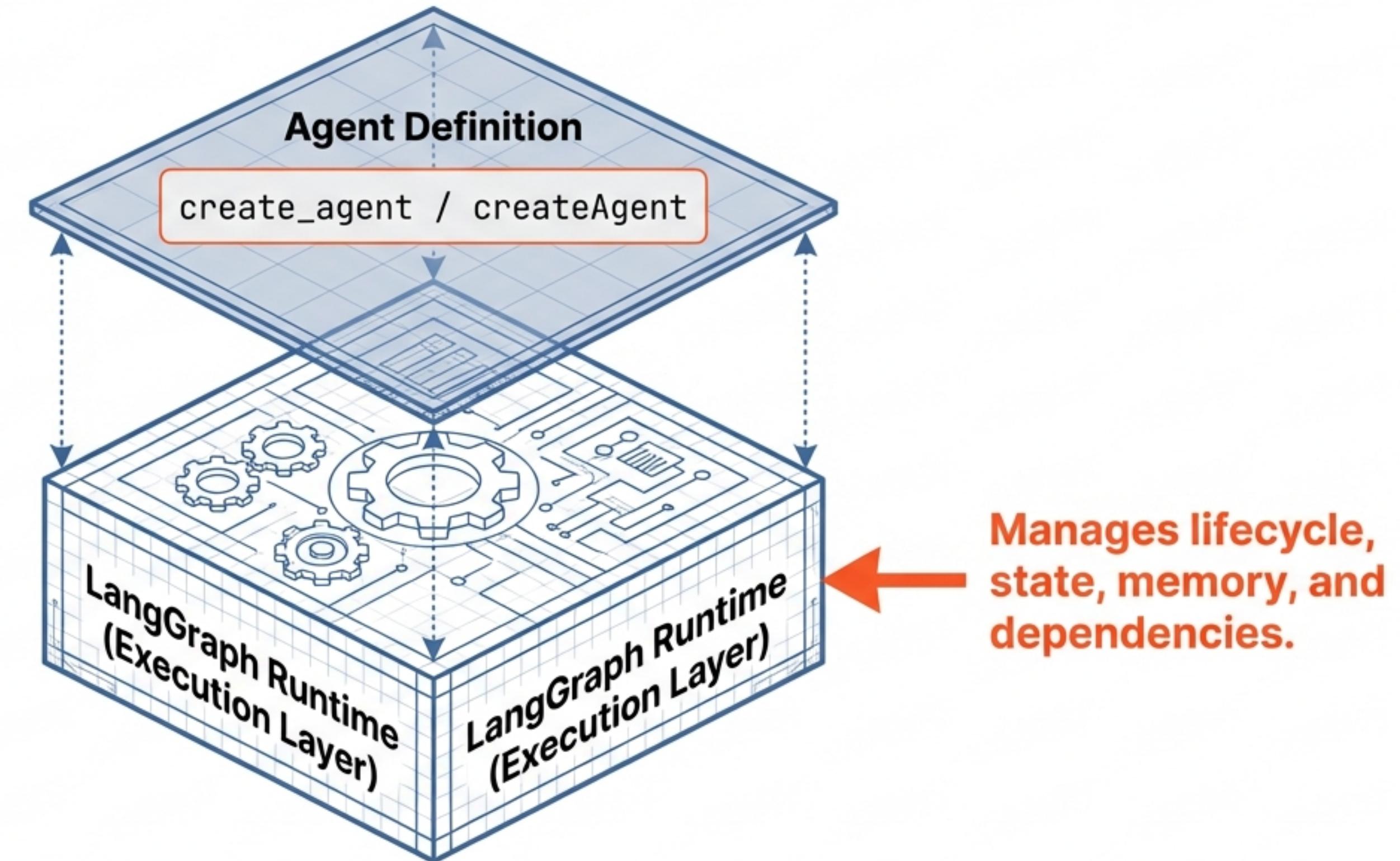
LangGraph Runtime: The Engine Behind `create_agent`

Architecture, State Management, and Dependency Injection for AI Agents.



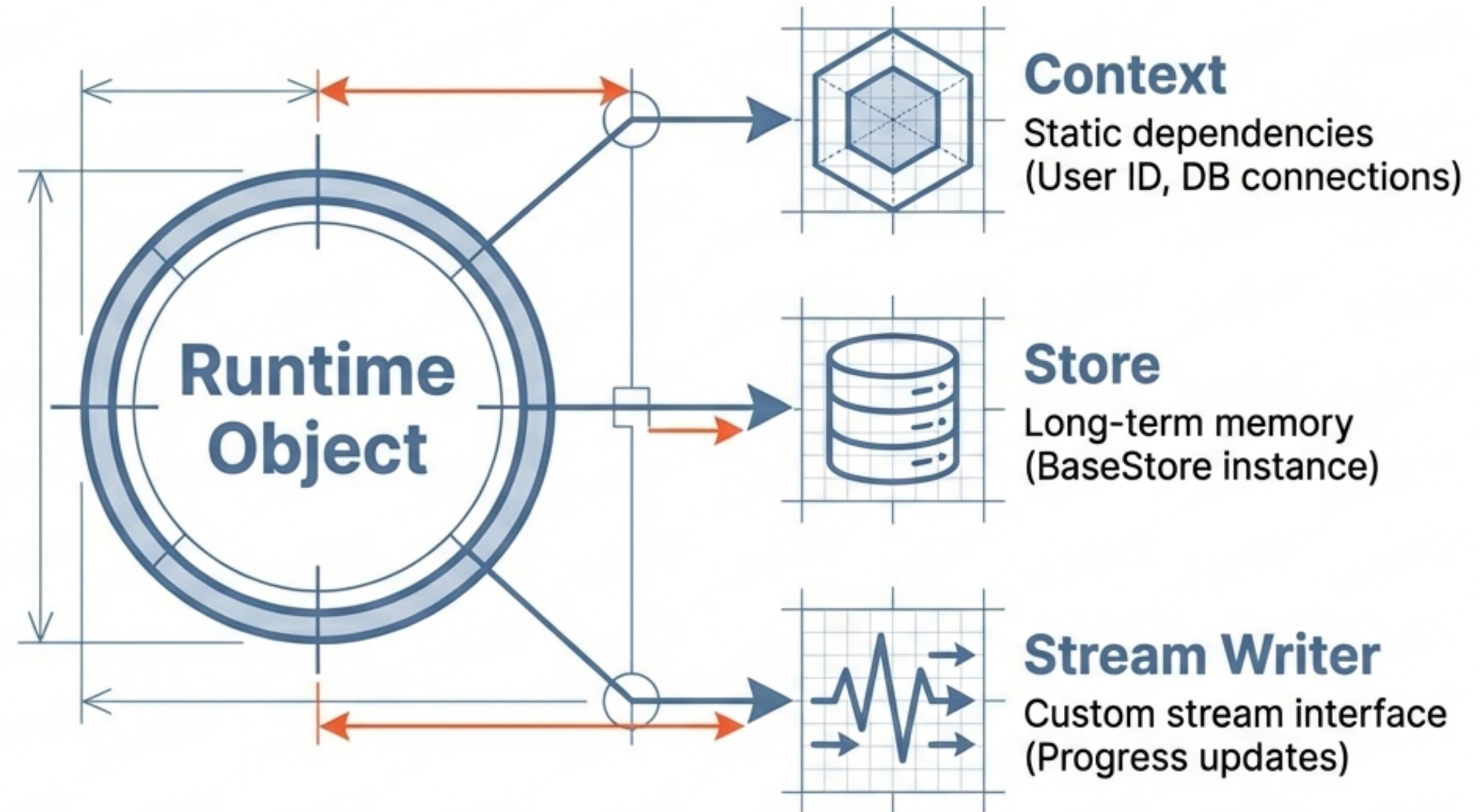
The Runtime Ecosystem

The `create_agent` function creates the definition, but the Runtime **manages the execution lifecycle, holding the state, memory, and dependencies.**



Anatomy of the Runtime Object

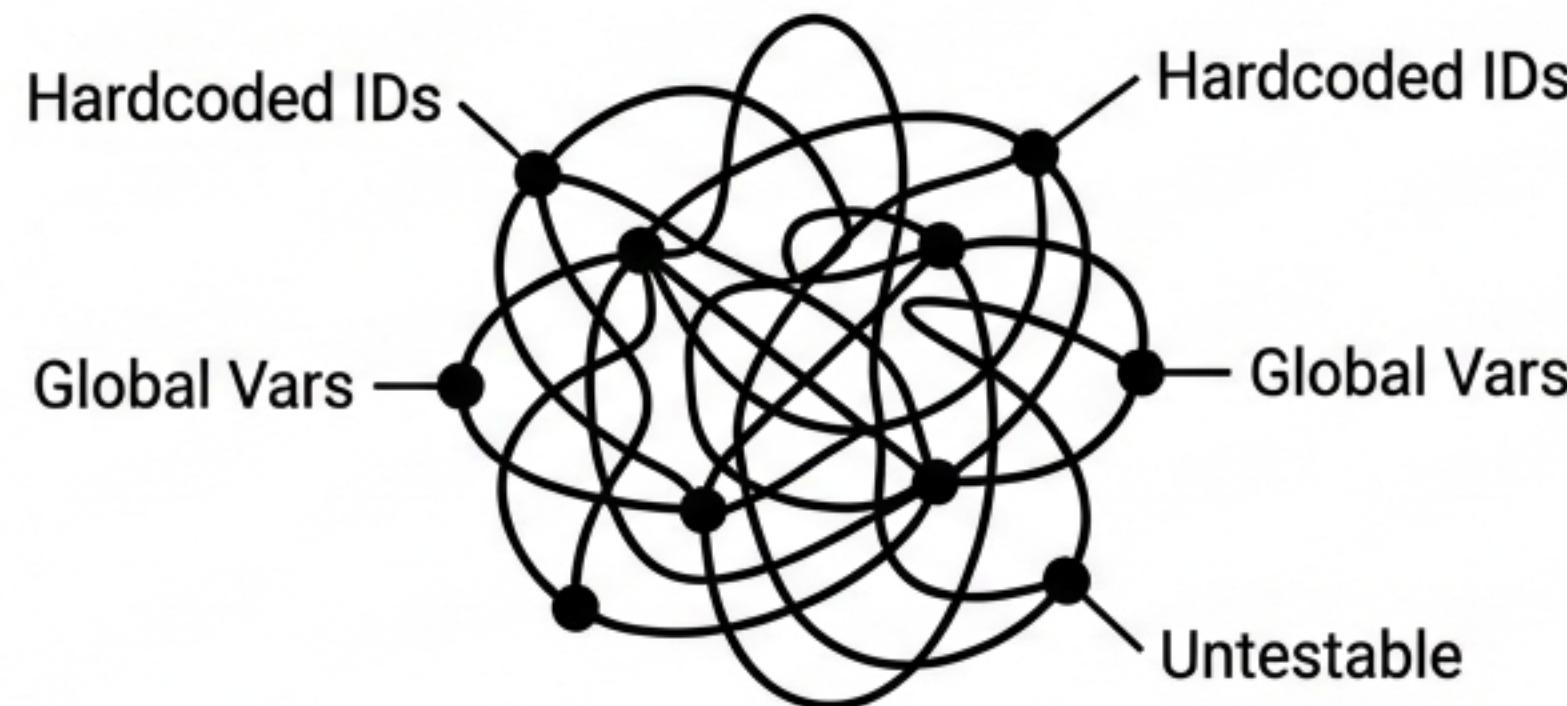
The Runtime object exposes three critical capabilities to your application logic. It serves as the **bridge between your code and the execution environment**.



The Power of Dependency Injection

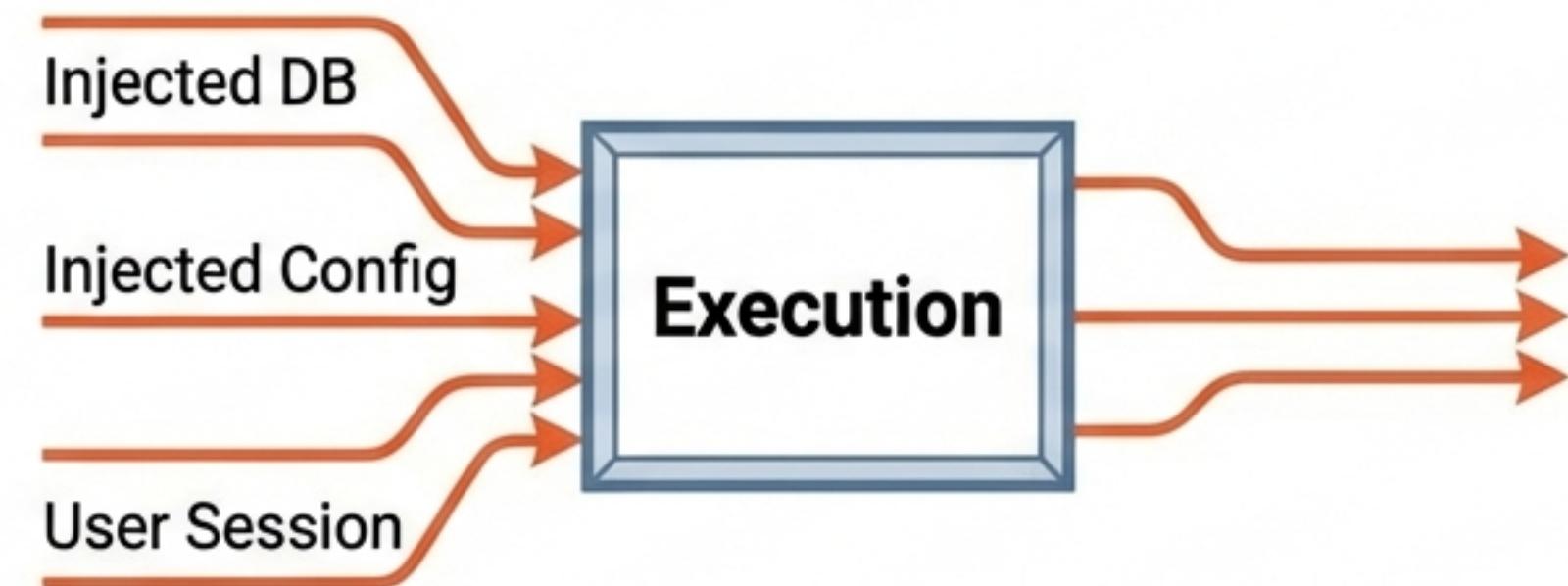
Stop hardcoding. Start injecting.

Global State (The Old Way)



Brittle & Messy

Runtime Context (The New Way)



Stateless, Testable, Flexible

Runtime context allows you to attach values directly to the execution flow, ensuring tools remain independent of the environment.

Defining the Context Schema

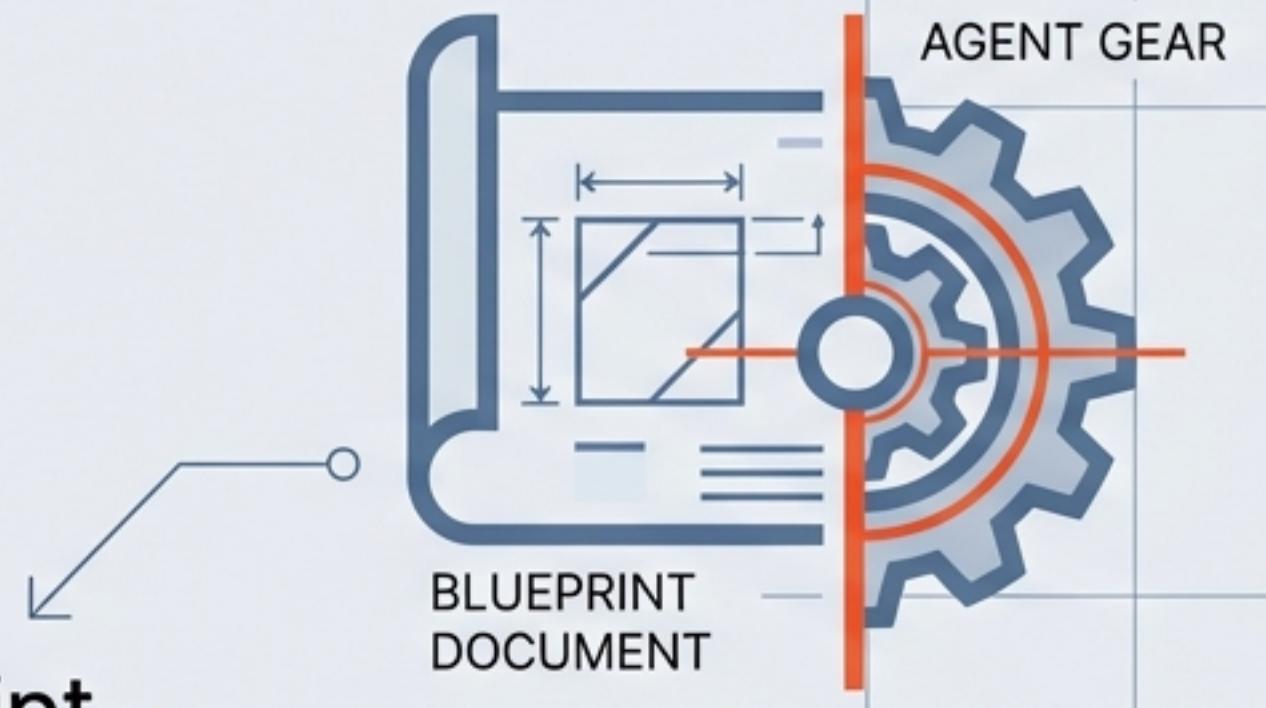
Before creating an agent, define the structure (contract) of the context it will hold.

Python

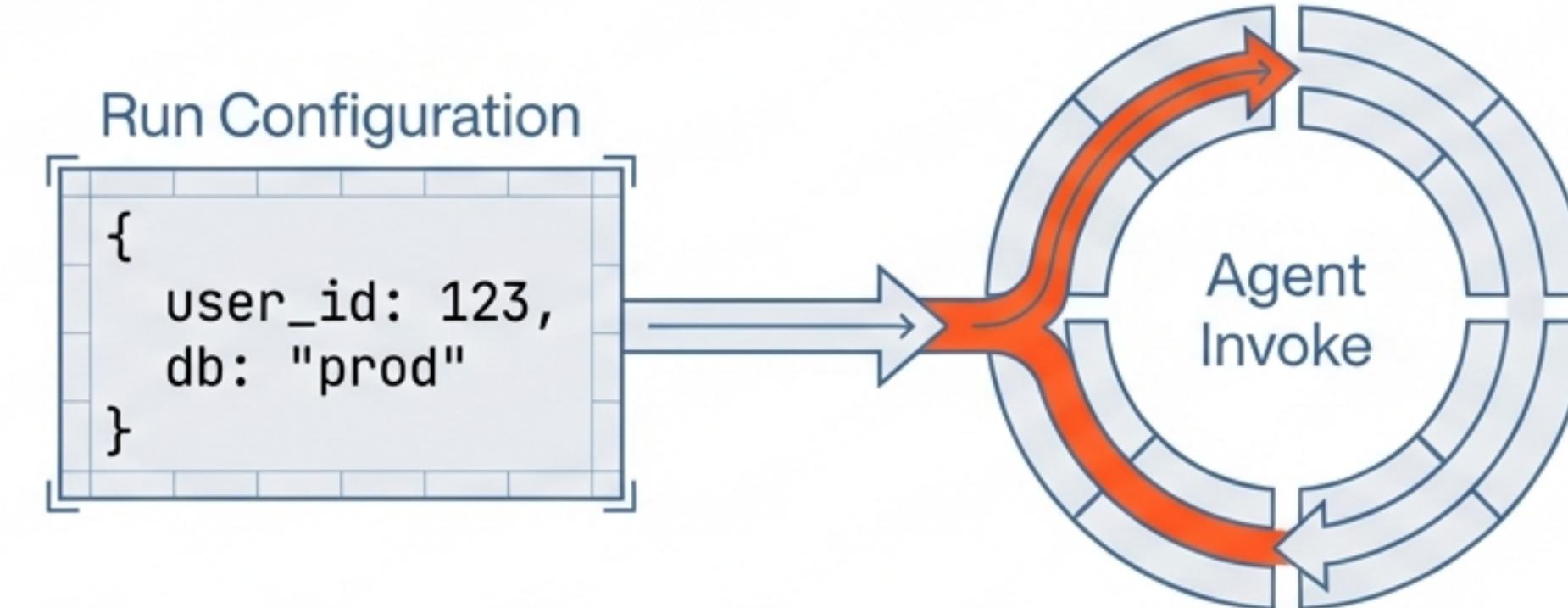
```
# Python
create_agent(
    ...,
    context_schema=MyContext
)
```

JavaScript

```
// JavaScript
createAgent(
    ...,
    { contextSchema: MyContext }
)
```



Invoking with Context

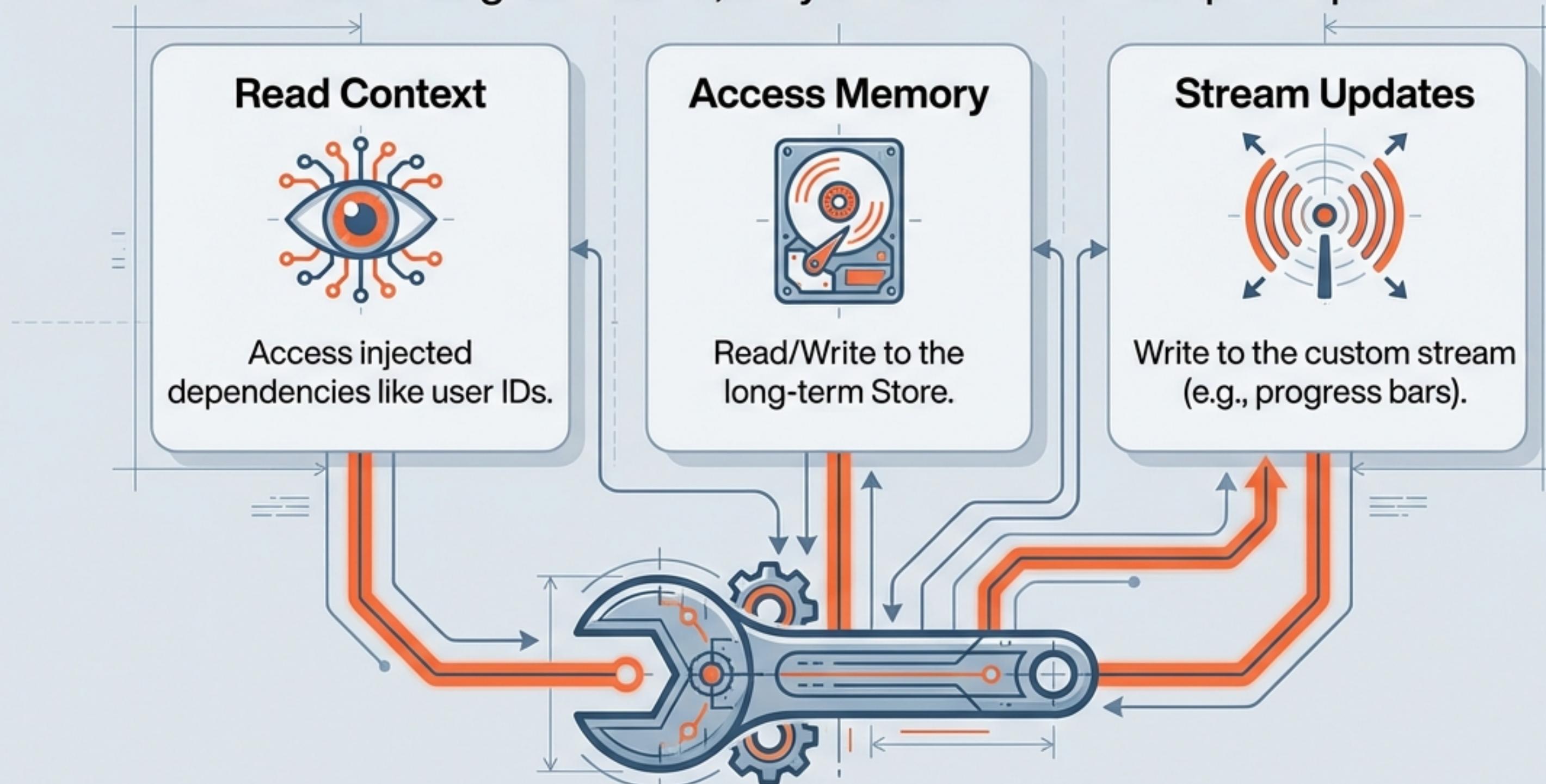


```
agent.invoke(  
    inputs,  
    context={  
        "user_id": "123",  
        "db_connection": db_conn  
    }  
)
```

Dependency Injection
Happens Here

Access Pattern A: Inside Tools (Capabilities)

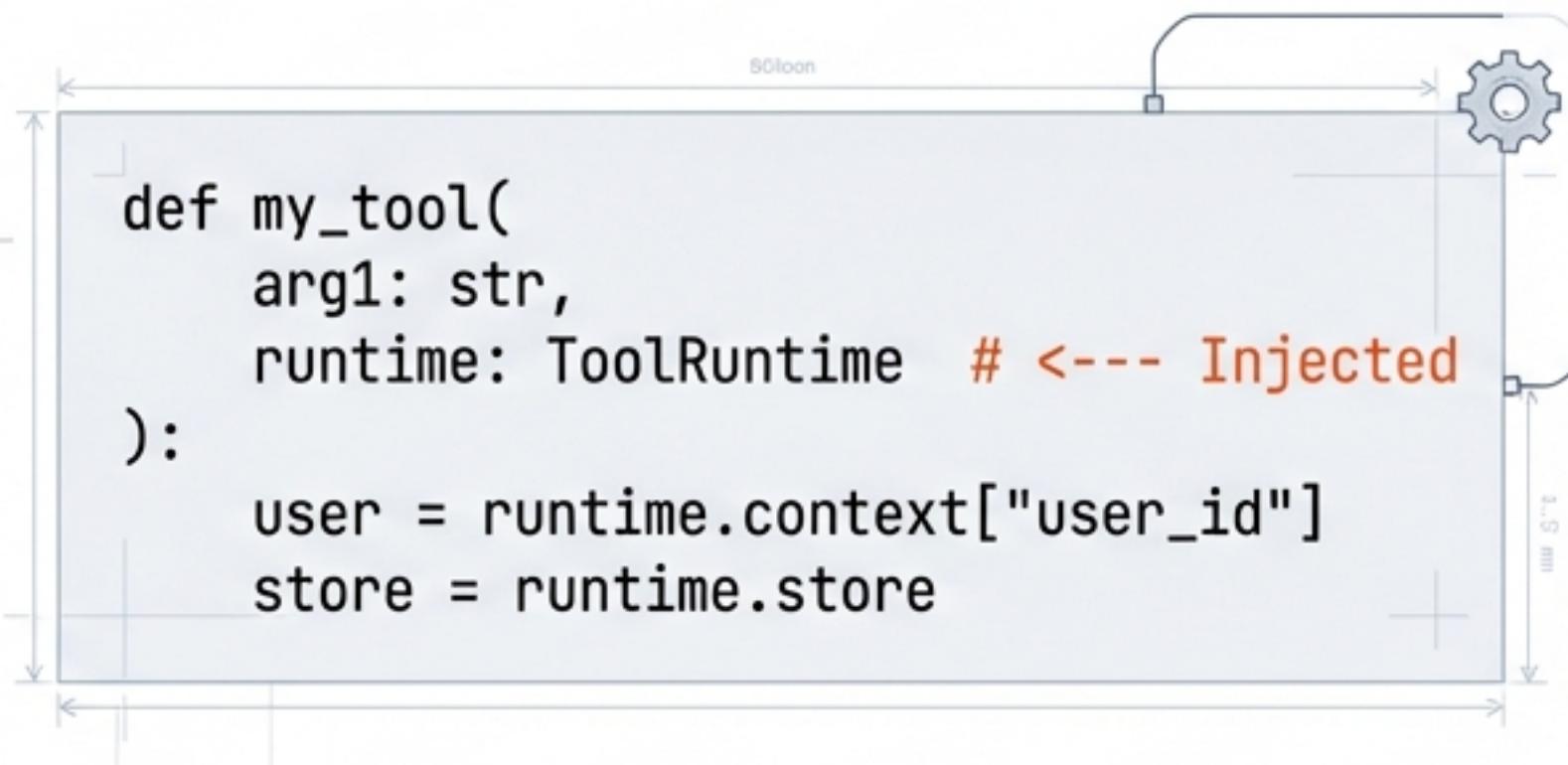
Tools are no longer isolated; they are context-aware participants.



Access Pattern A: Inside Tools (Syntax)

Python

Type hint with `ToolRuntime`.



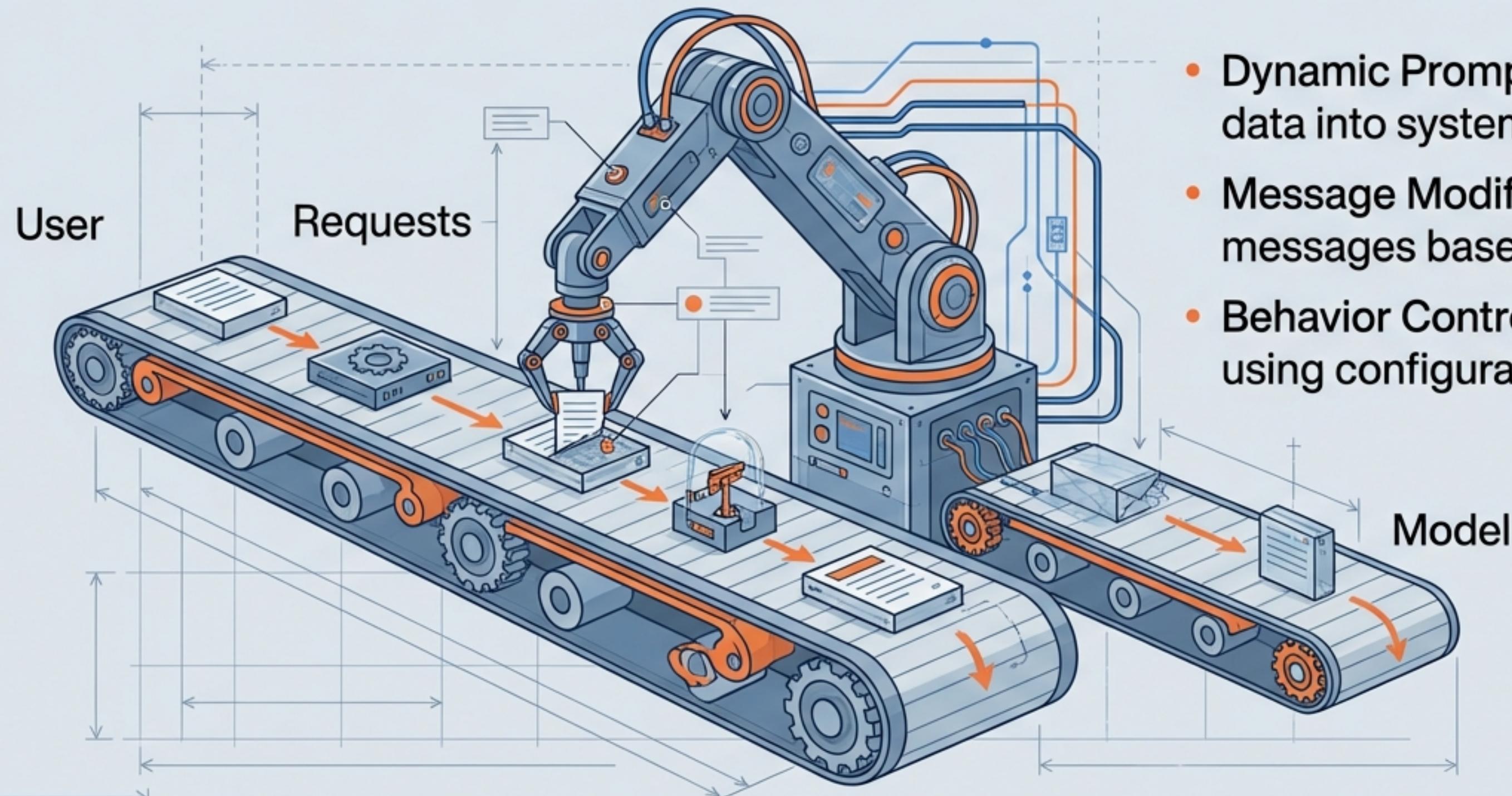
JavaScript

Destructure `runtime` from the second argument.



Access Pattern B: Inside Middleware (Capabilities)

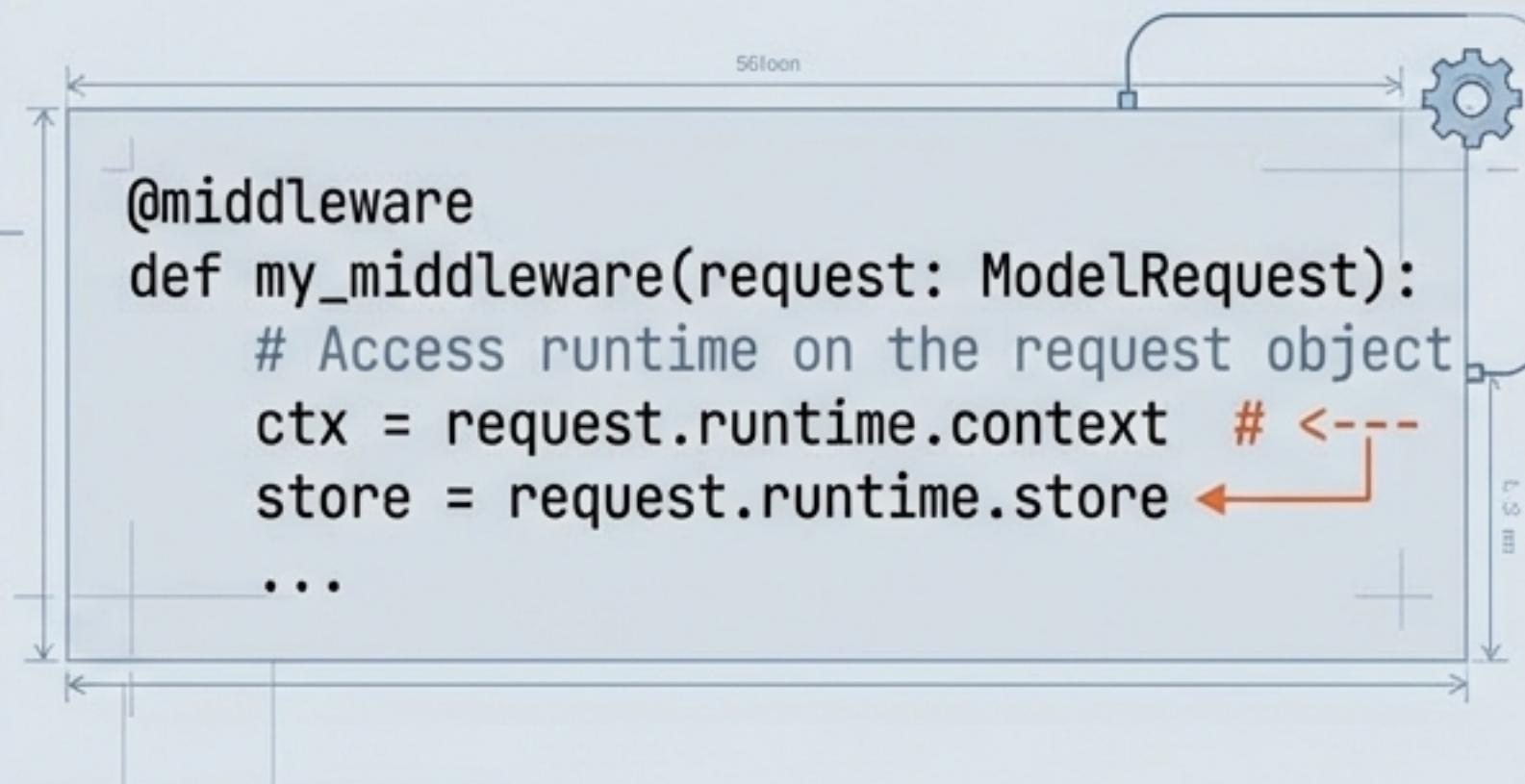
Middleware sits between the user and the model. Runtime access enables:



Access Pattern B: Inside Middleware (Syntax)

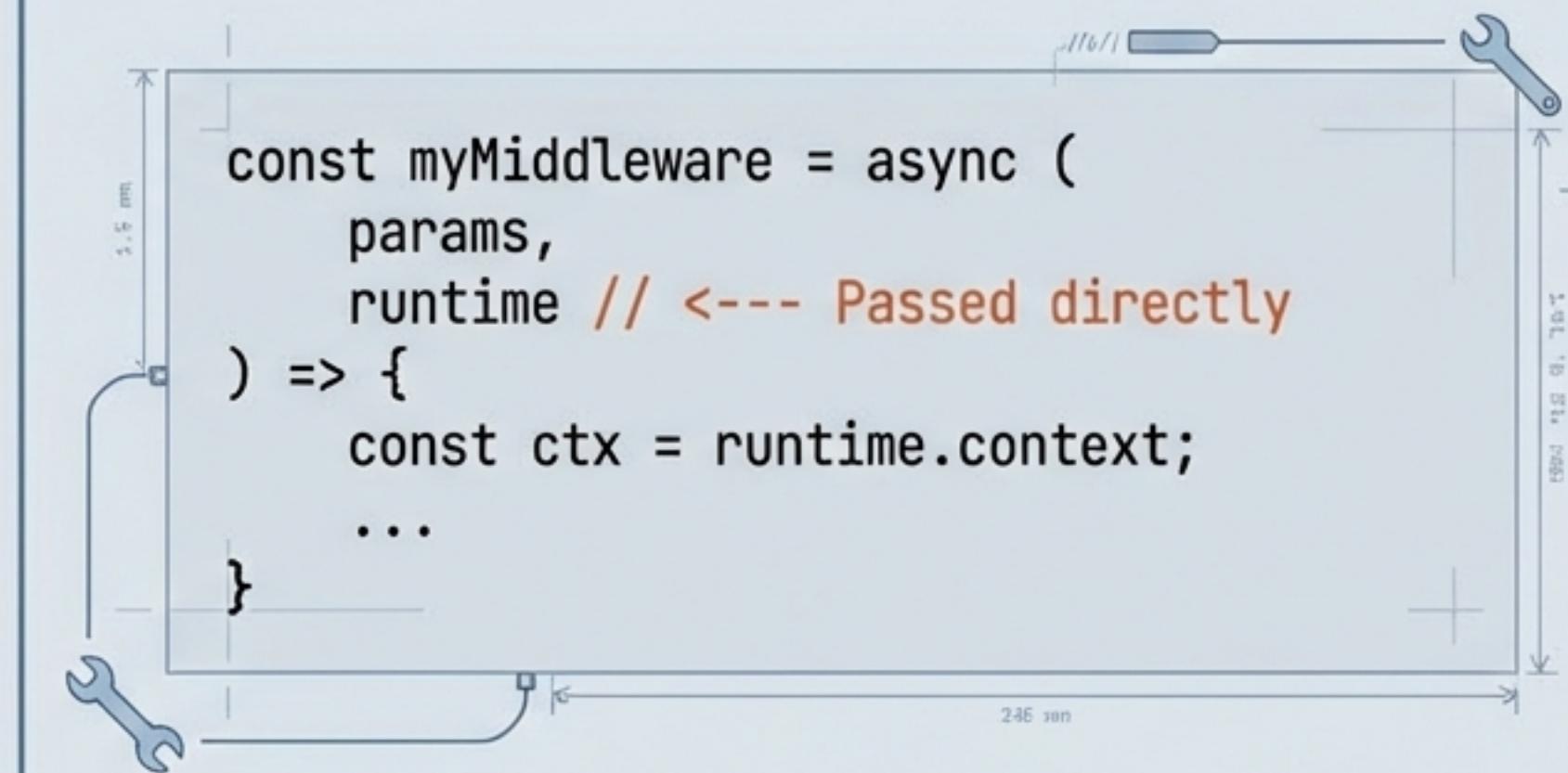
Python

Access via `request.runtime`.



JavaScript

Access via `runtime` parameter.



Architectural Reference Guide

Component	Definition	Tool Access Syntax	Middleware Access Syntax
Context	Static dependencies & config	runtime.context runtime.context ↪	request.runtime.context # <--
Store	Long-term Memory (BaseStore)	runtime.store runtime.store ↪	request.runtime.store # <--
Stream Writer	Custom stream channel	runtime.writer runtime.writer ↪	request.runtime.writer # <--

LangGraph Runtime: Stateless, Testable, Flexible.