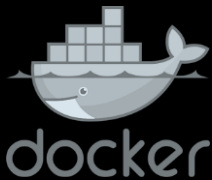




Enhanced Dialog Driven Interface

Transform Workflows & Elevate Collaboration
through ChatGPT & Conversational APIs



Hello!

I am Gregor Jarisch

E.D.I.

by LABS.AI

Head of R&D

Open Source Chatbot Middleware

gnowbe

CTO & Head of Product

Award winning Micro-Learning Solution



More about these projects:

eddi.labs.ai

|

gnowbe.com

1.

What is E.D.D.I ?

Short intro

E.D.D.I – Open Source Chatbot Platform

- © Enterprise-Certified & Cloud Native Restful Micro Service (powered by Quarkus 3)
- © Multiple Chatbots with Versioning for a smooth upgrading experience
- © Basic NLP & Behavior Rules, API Connector, Templating
- © Extension Oriented Plugin System

2.

Commercial Support

Expert Knowledge Saves U-Turns

Open Source + Membership Subscription

Open Source

- > Core E.D.D.I
 - > NLP, Rules, Output, API Connectors, ...
 - > Full-fledged API
- > Docker Image
- > Optimized for Kubernetes & Red Hat Openshift



kubernetes



docker

Commercial

- + Dedicated Support
- + Training / Consulting
- + Customization
- + Roadmap
- + Managed Service with
24/7 Support

25k

Lines of Code

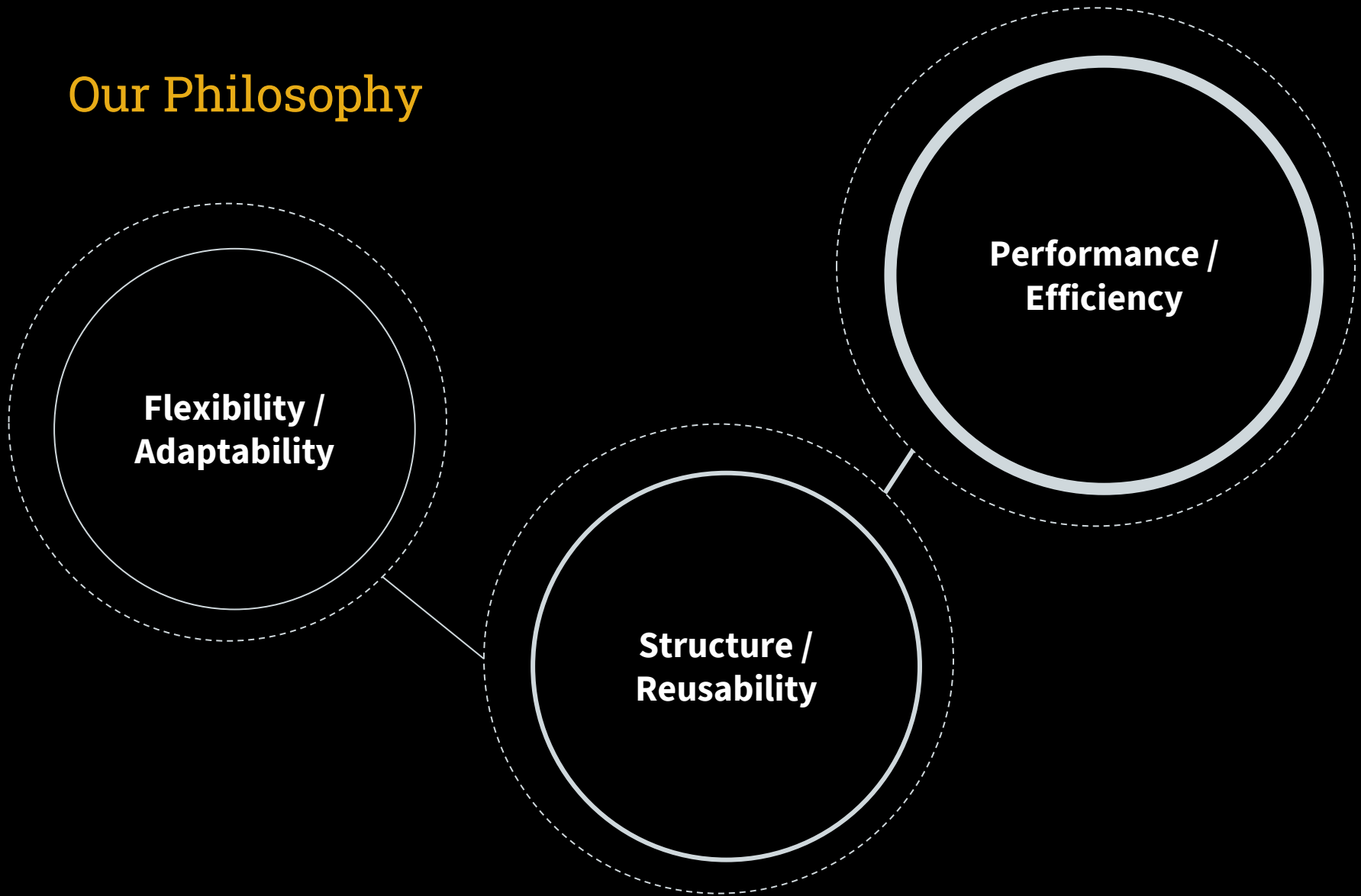
15

Years of Engineering

5

Development Iterations from Scratch

Our Philosophy

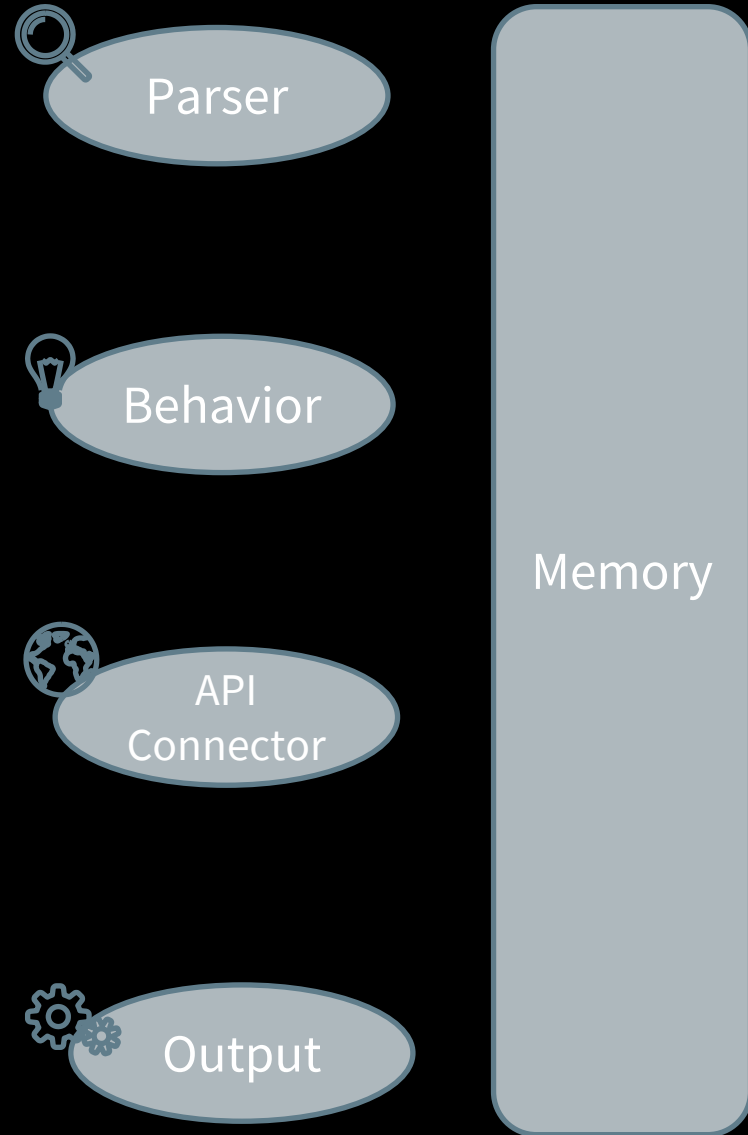
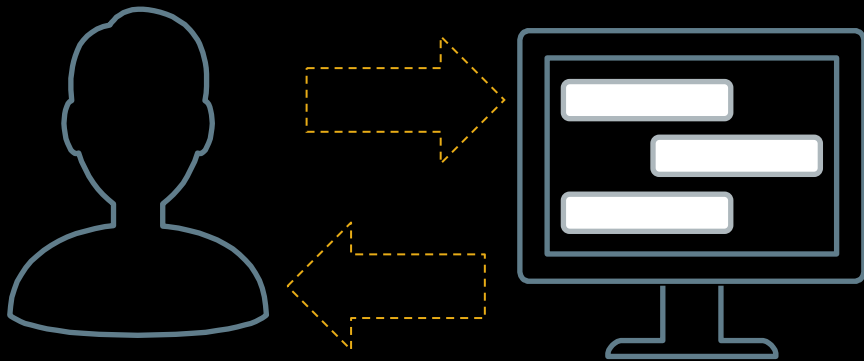


3.

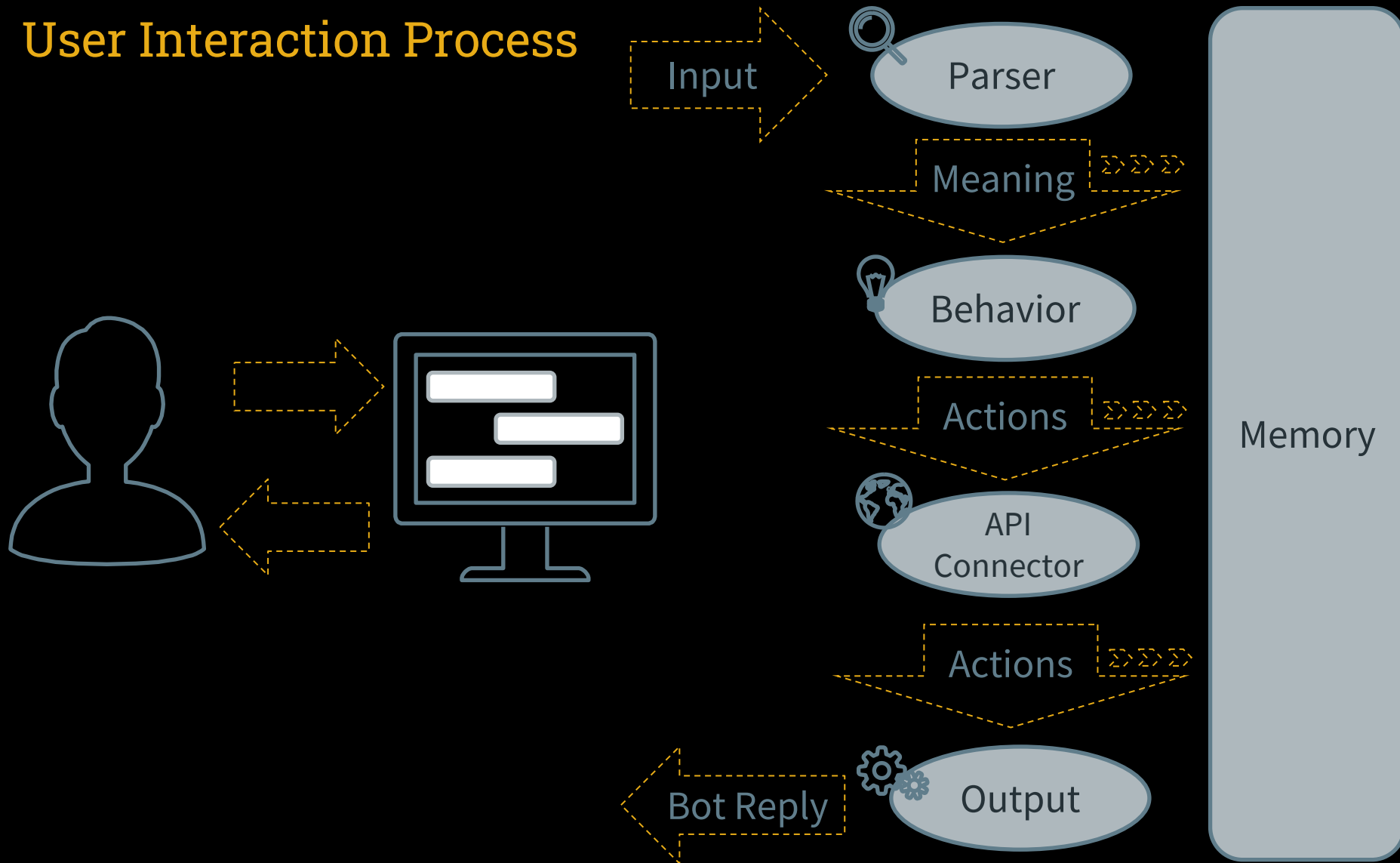
Example in Action

Let's Build a ChatGPT powered Bot

User Interaction Process

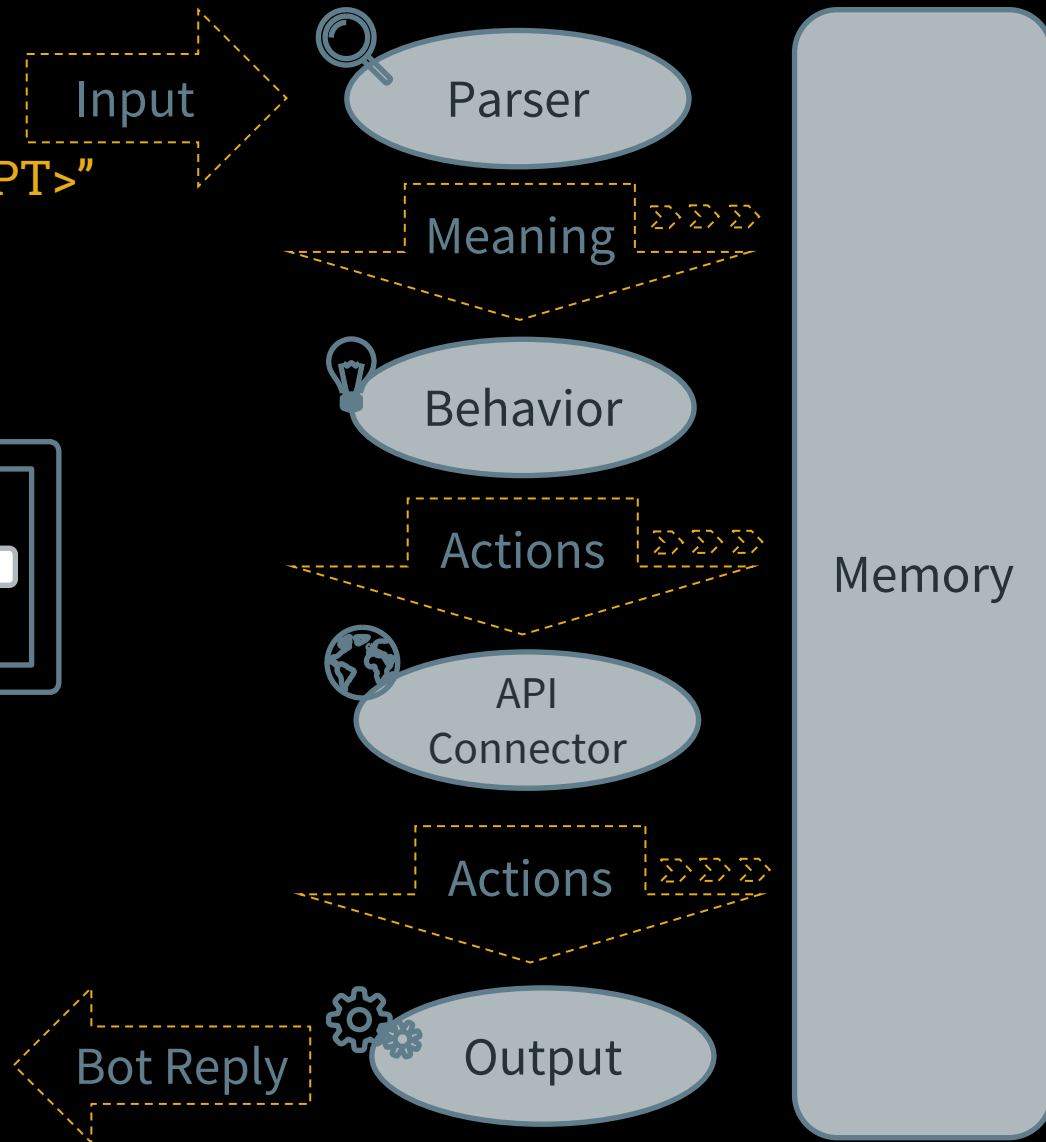
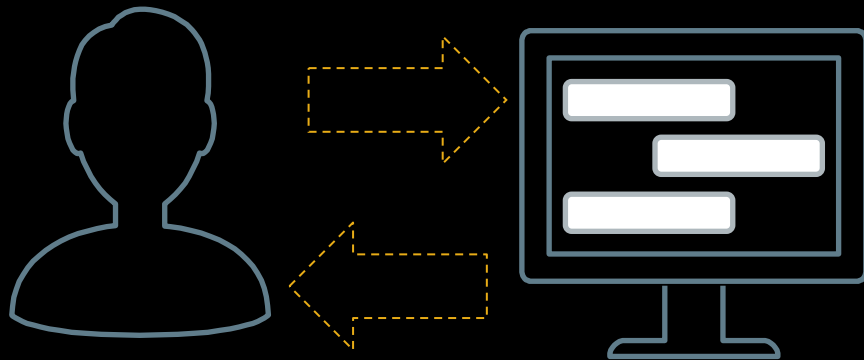


User Interaction Process



User Interaction Process

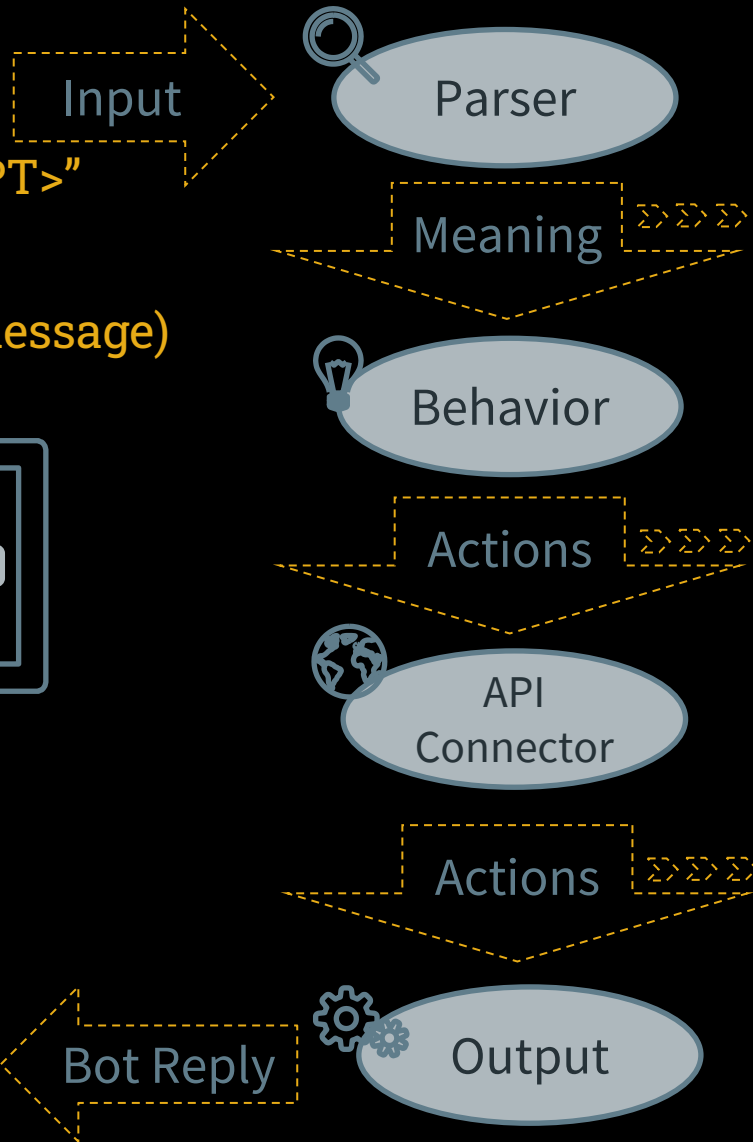
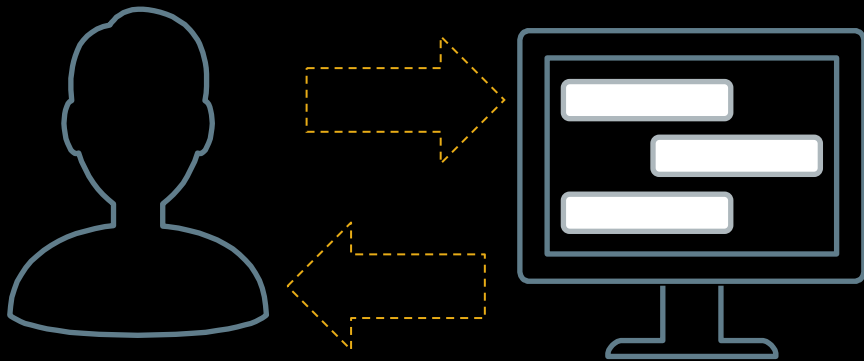
"<User input for ChatGPT>"



User Interaction Process

"<User input for ChatGPT>"

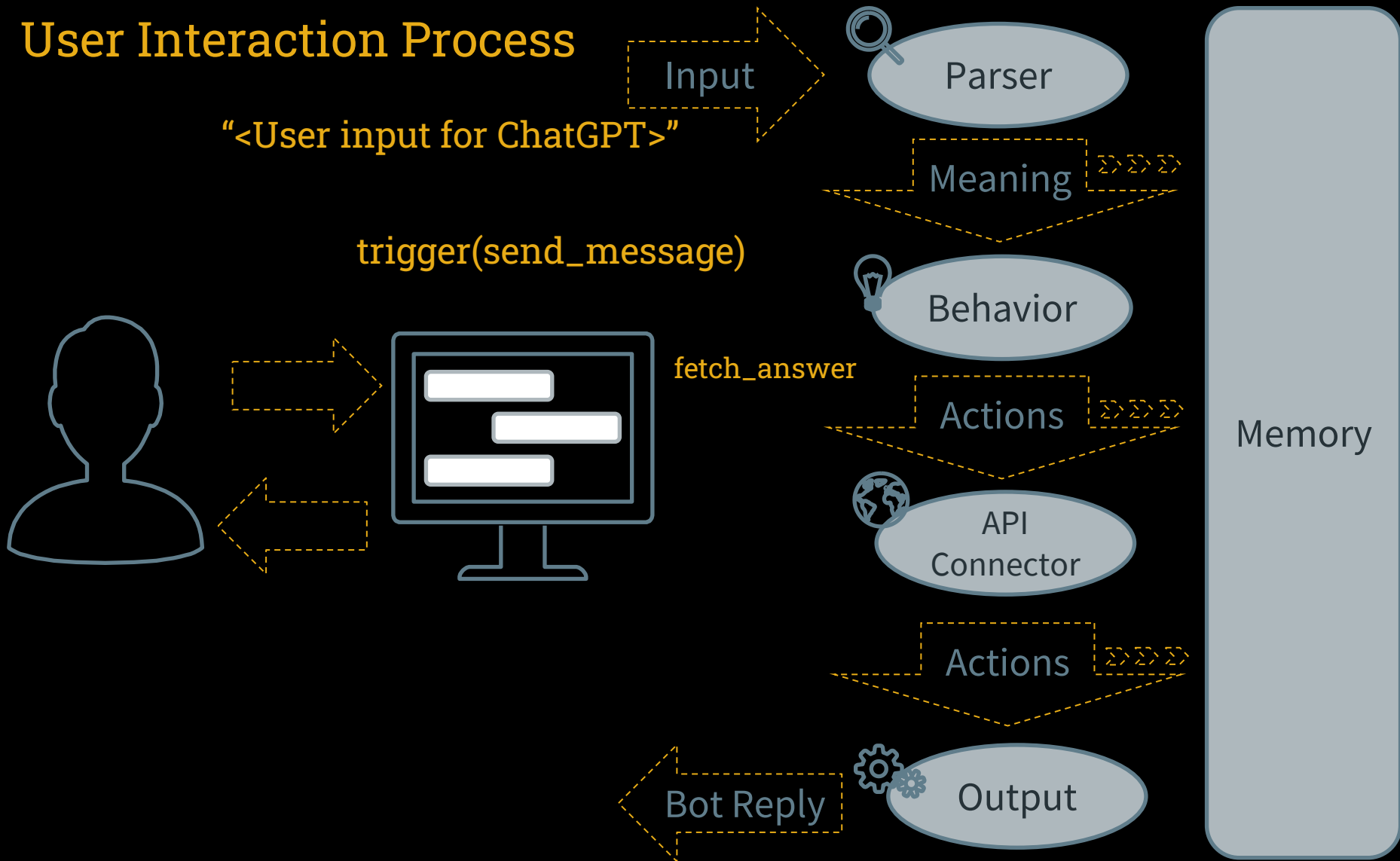
trigger(send_message)



Memory

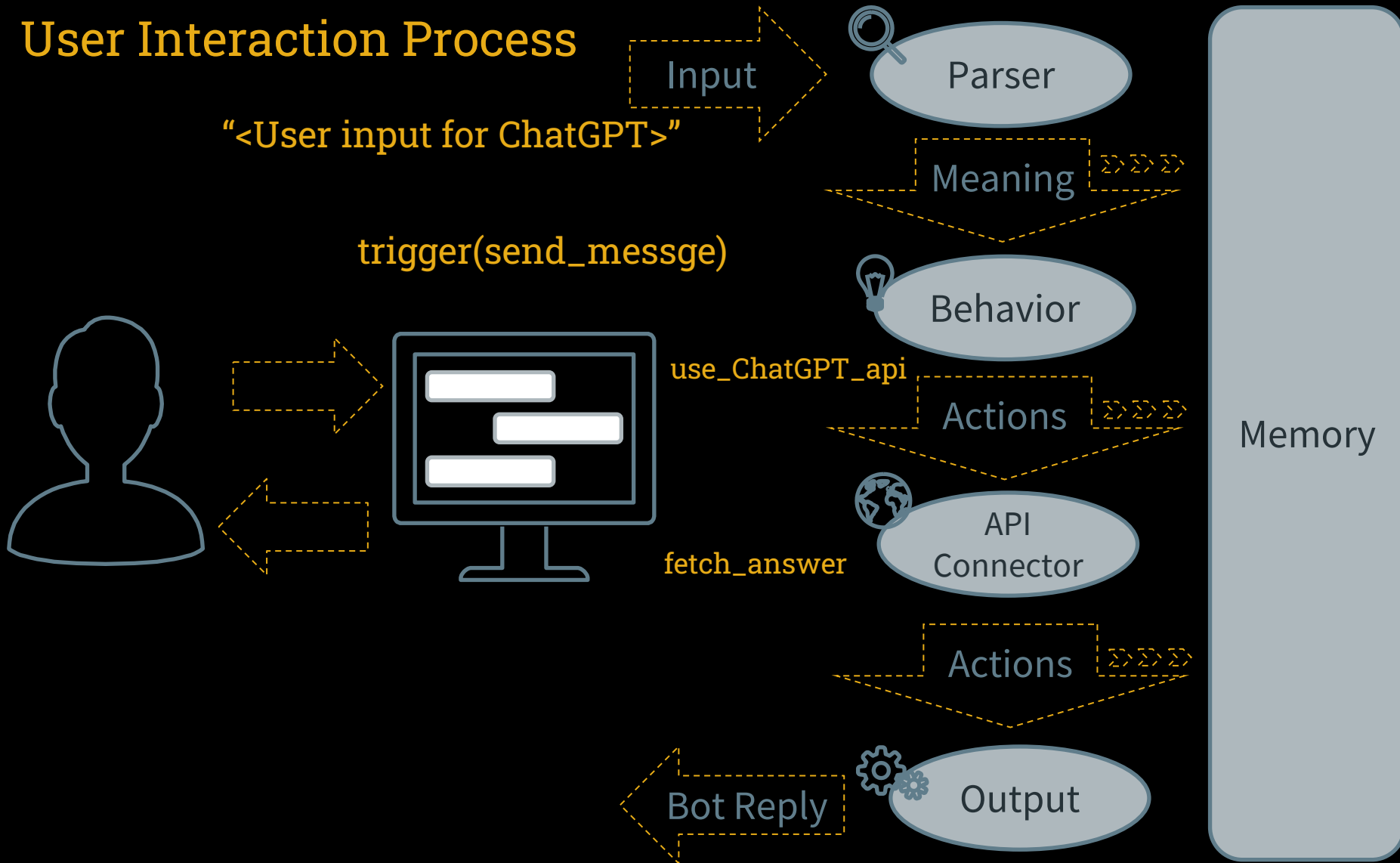
User Interaction Process

"<User input for ChatGPT>"



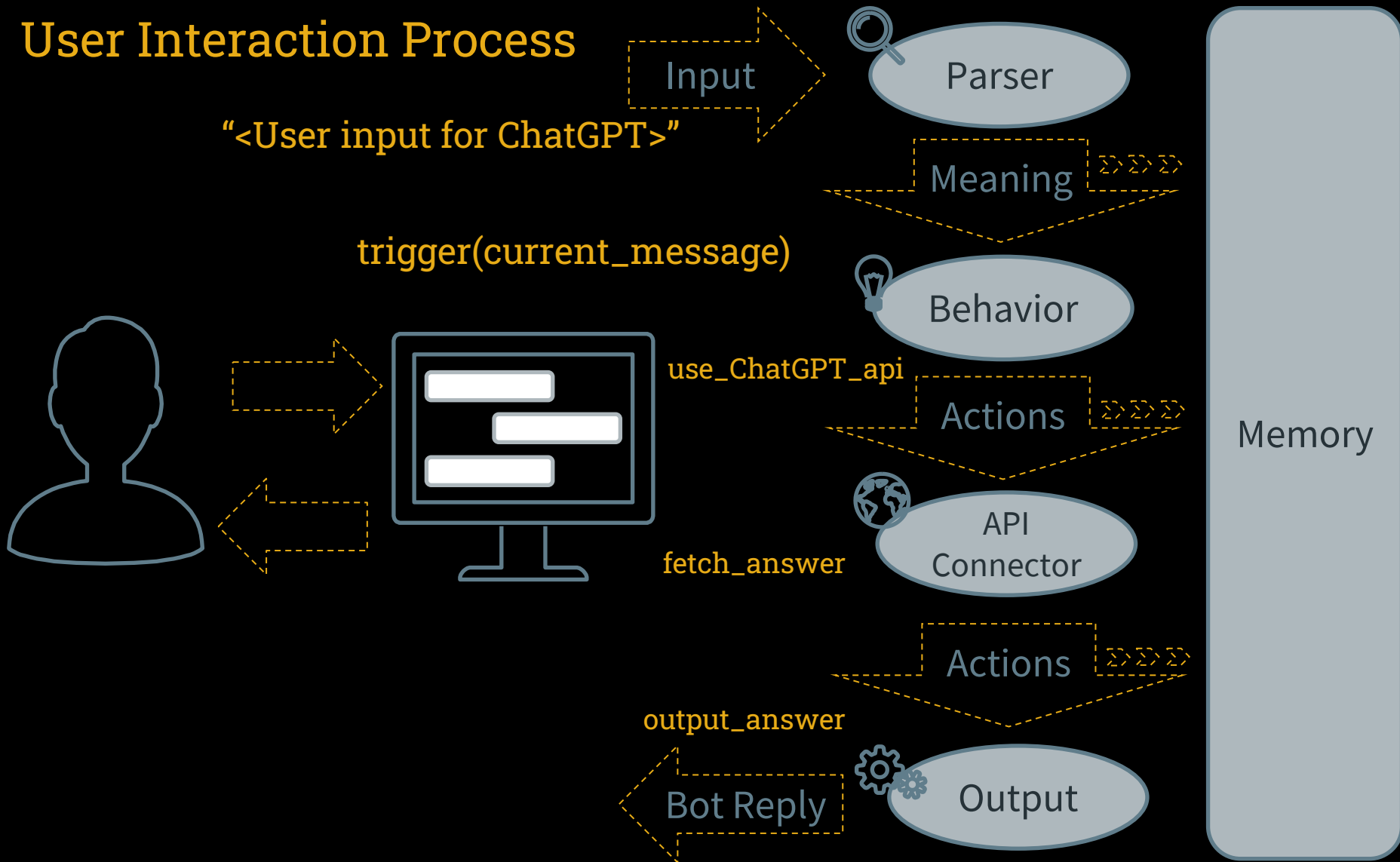
User Interaction Process

"<User input for ChatGPT>"



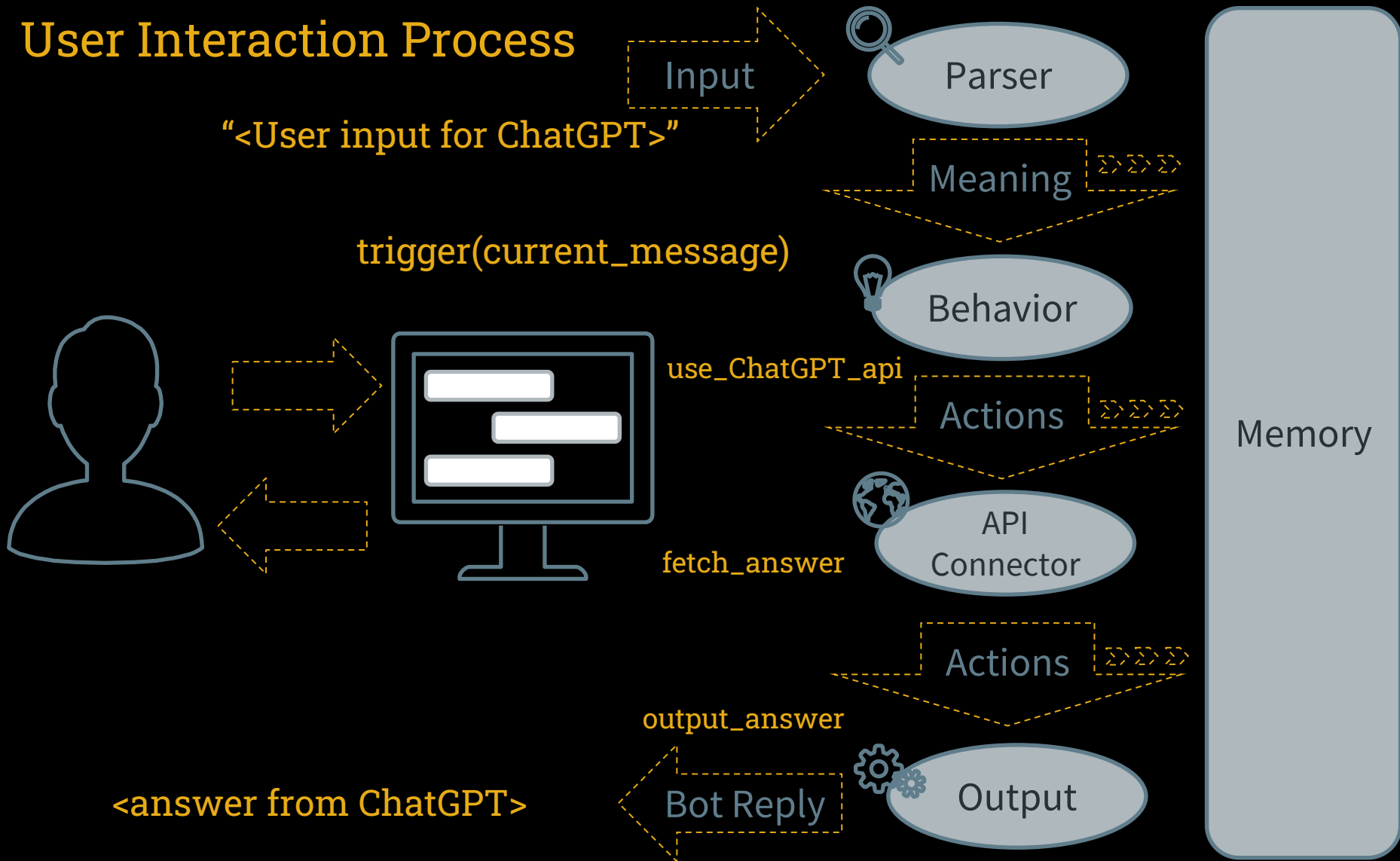
User Interaction Process

"<User input for ChatGPT>"



User Interaction Process

"<User input for ChatGPT>"



4.

Another Example in Action

Let's integrate a different API
without using ChatGPT

What's the weather?

*What's the **weather**?*



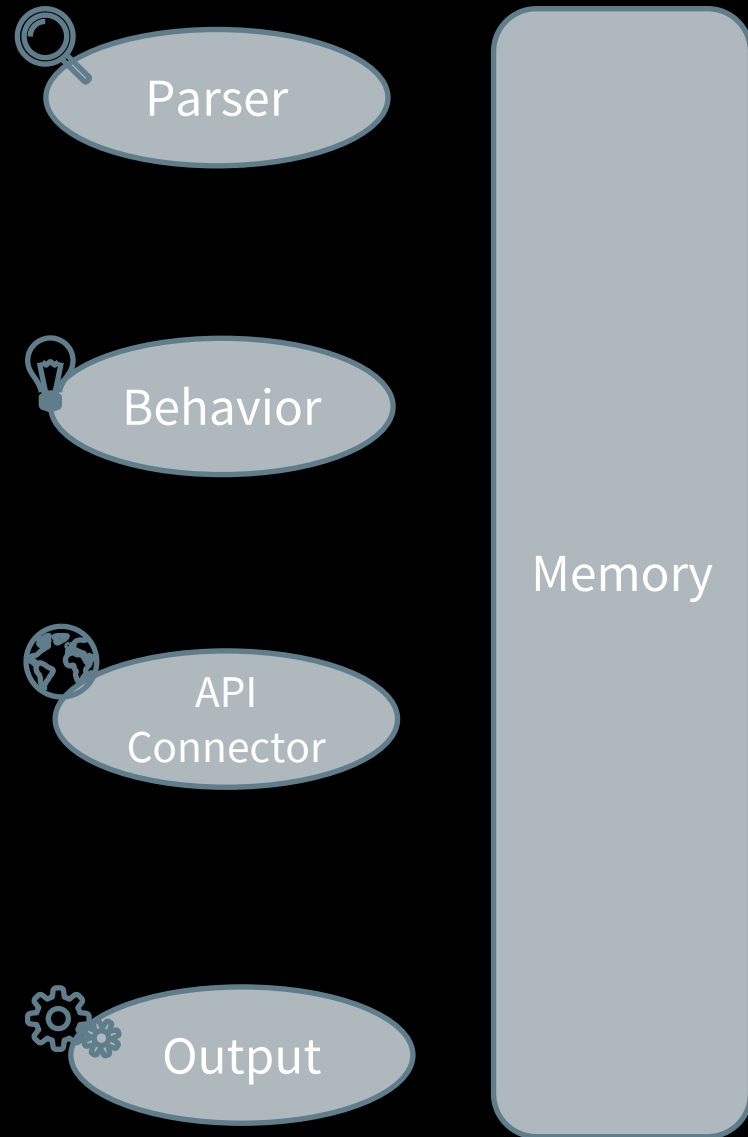
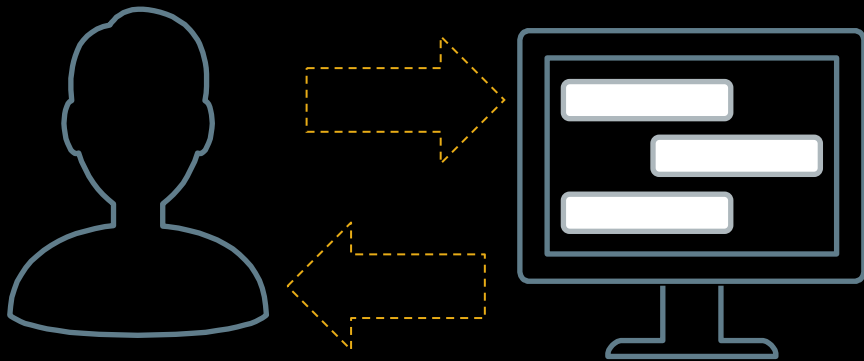
*Which city would you like to know
the weather of?*

New York

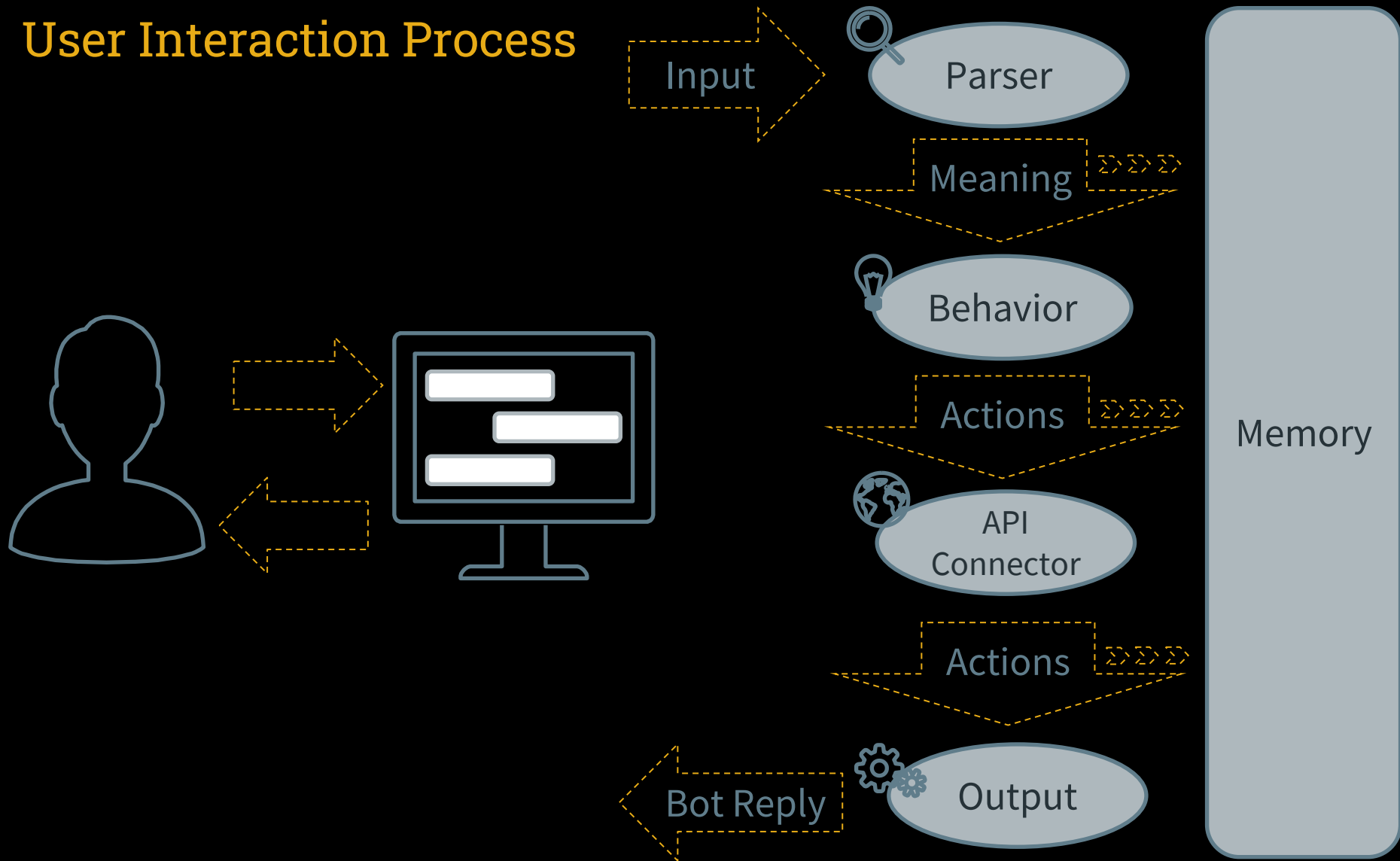


The current weather situation in New York is light rain at 18°C.

User Interaction Process

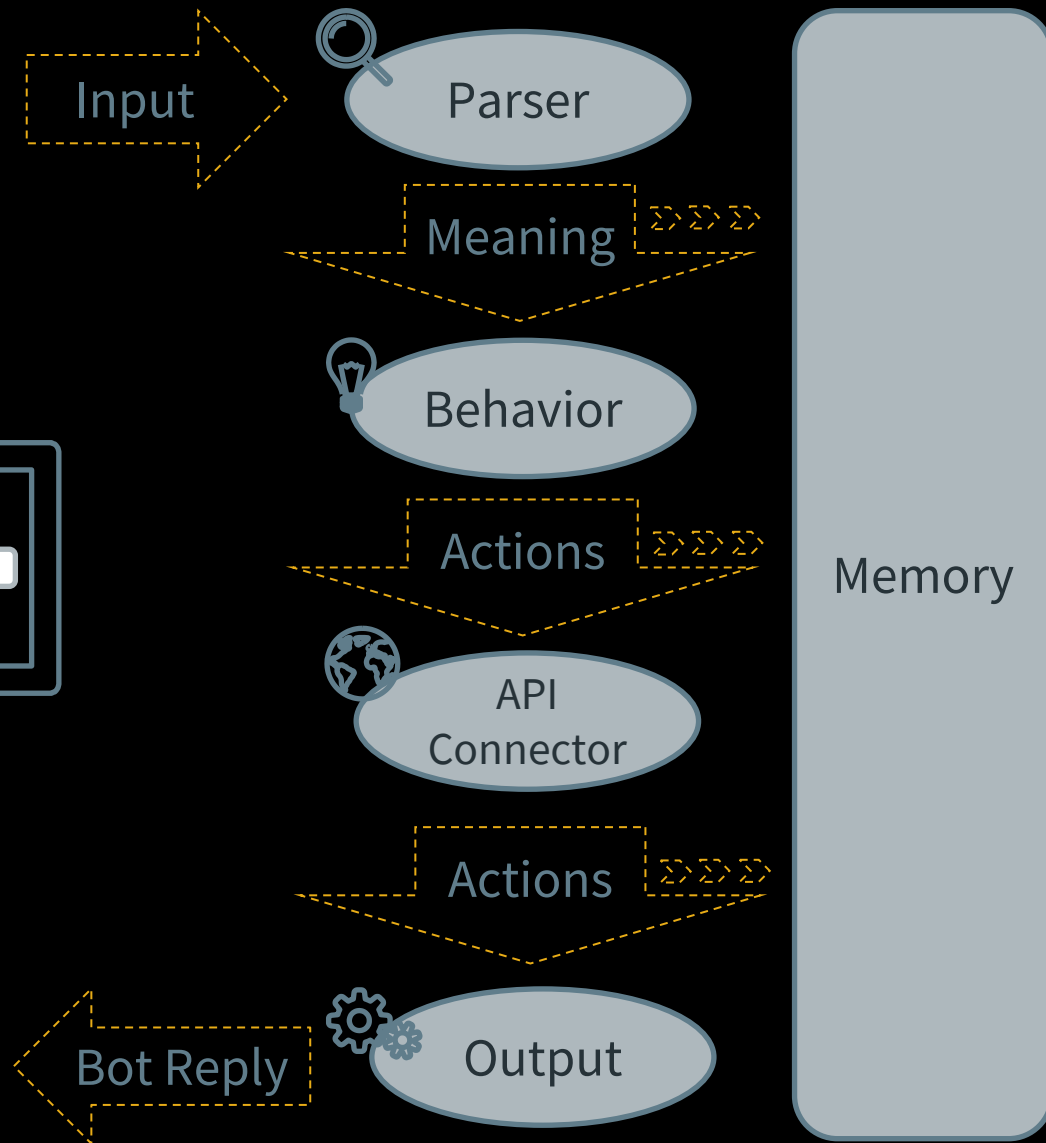
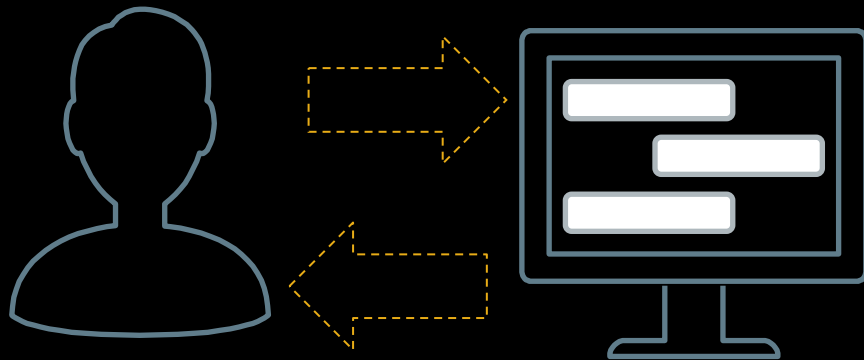


User Interaction Process



User Interaction Process

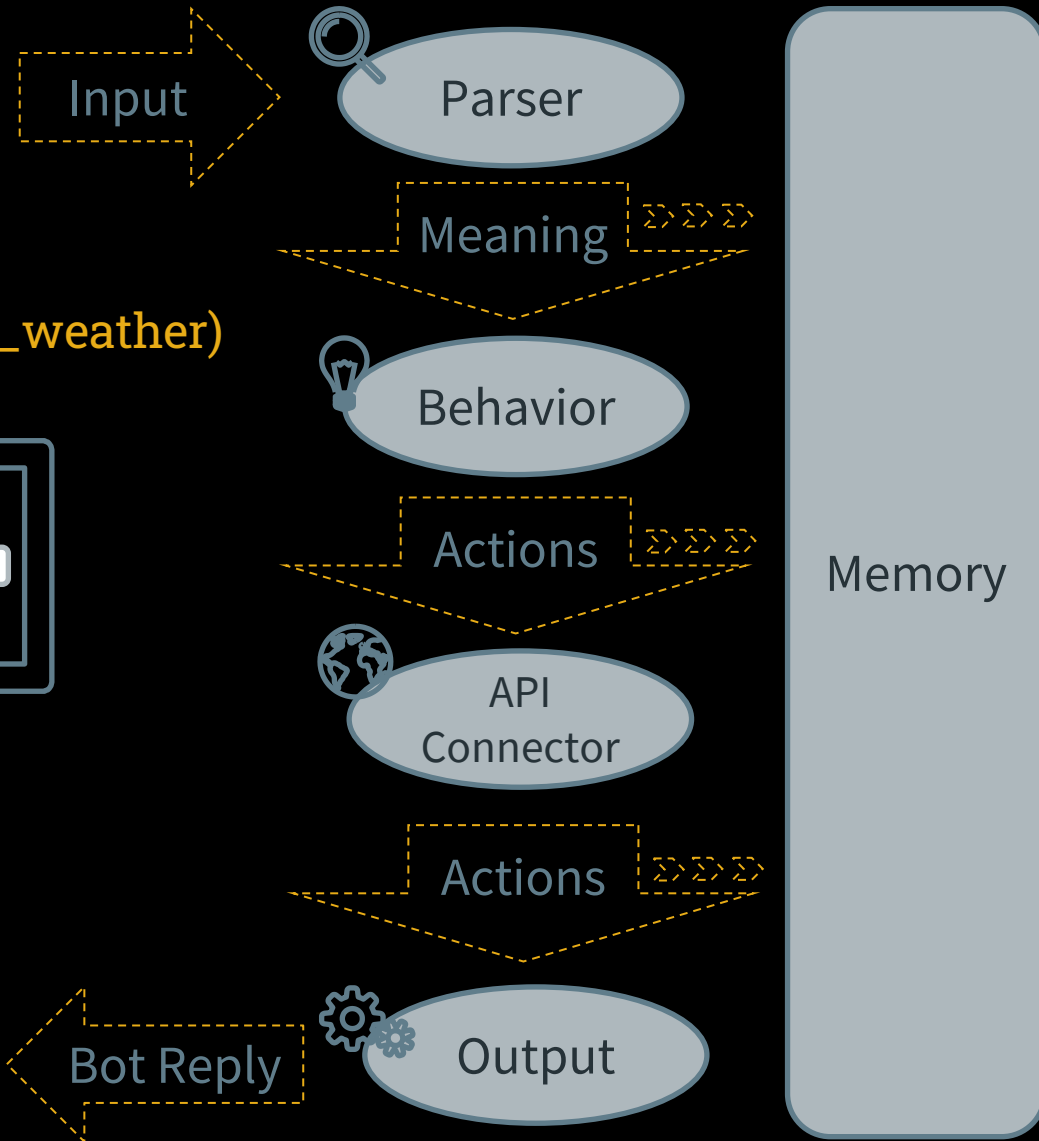
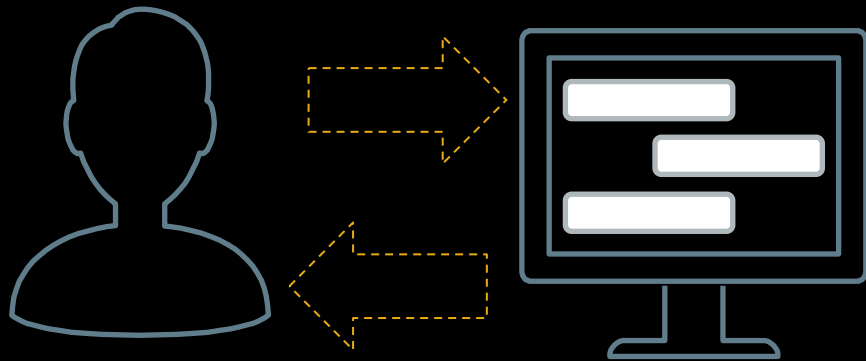
"What's the weather"



User Interaction Process

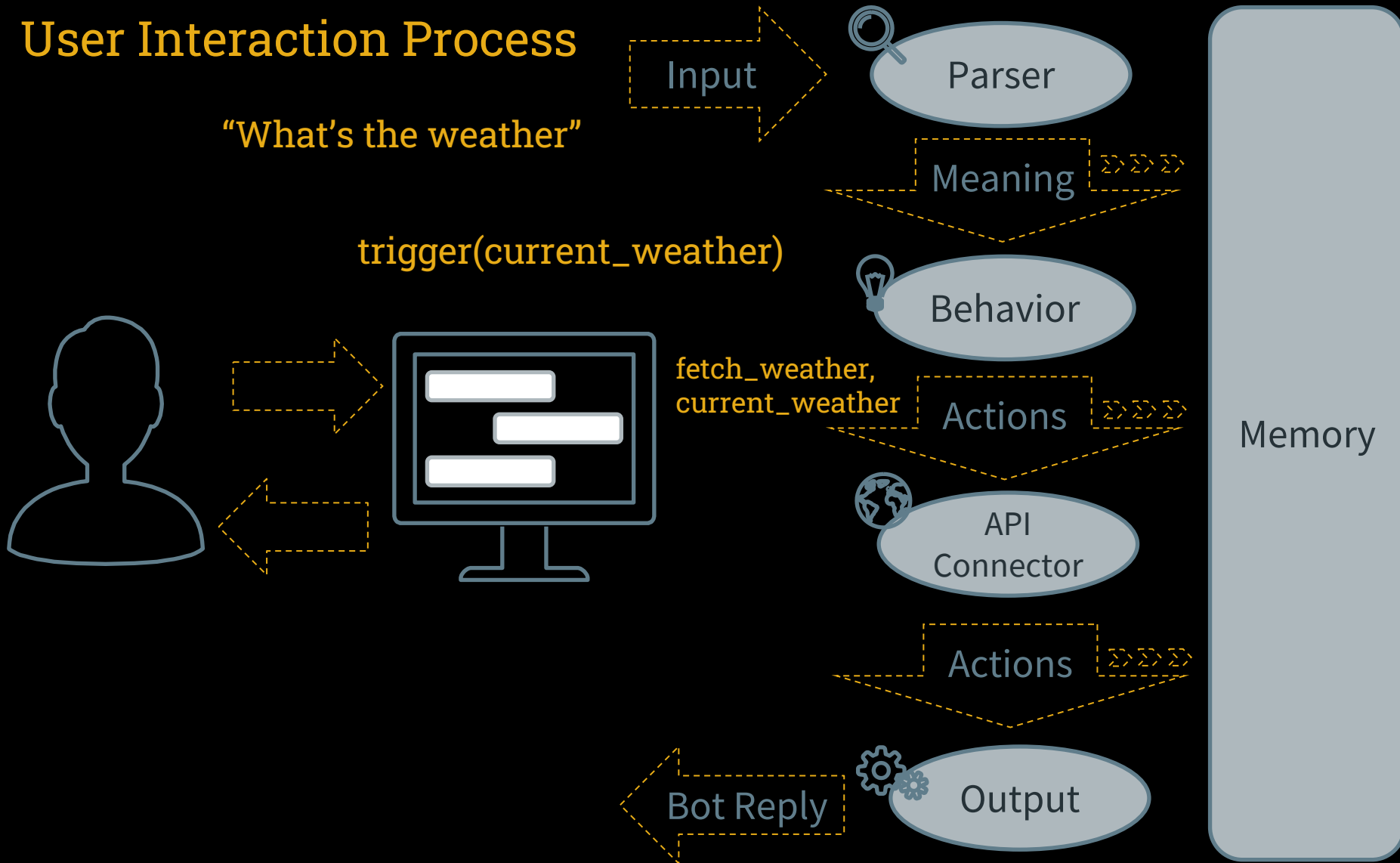
"What's the weather"

trigger(current_weather)



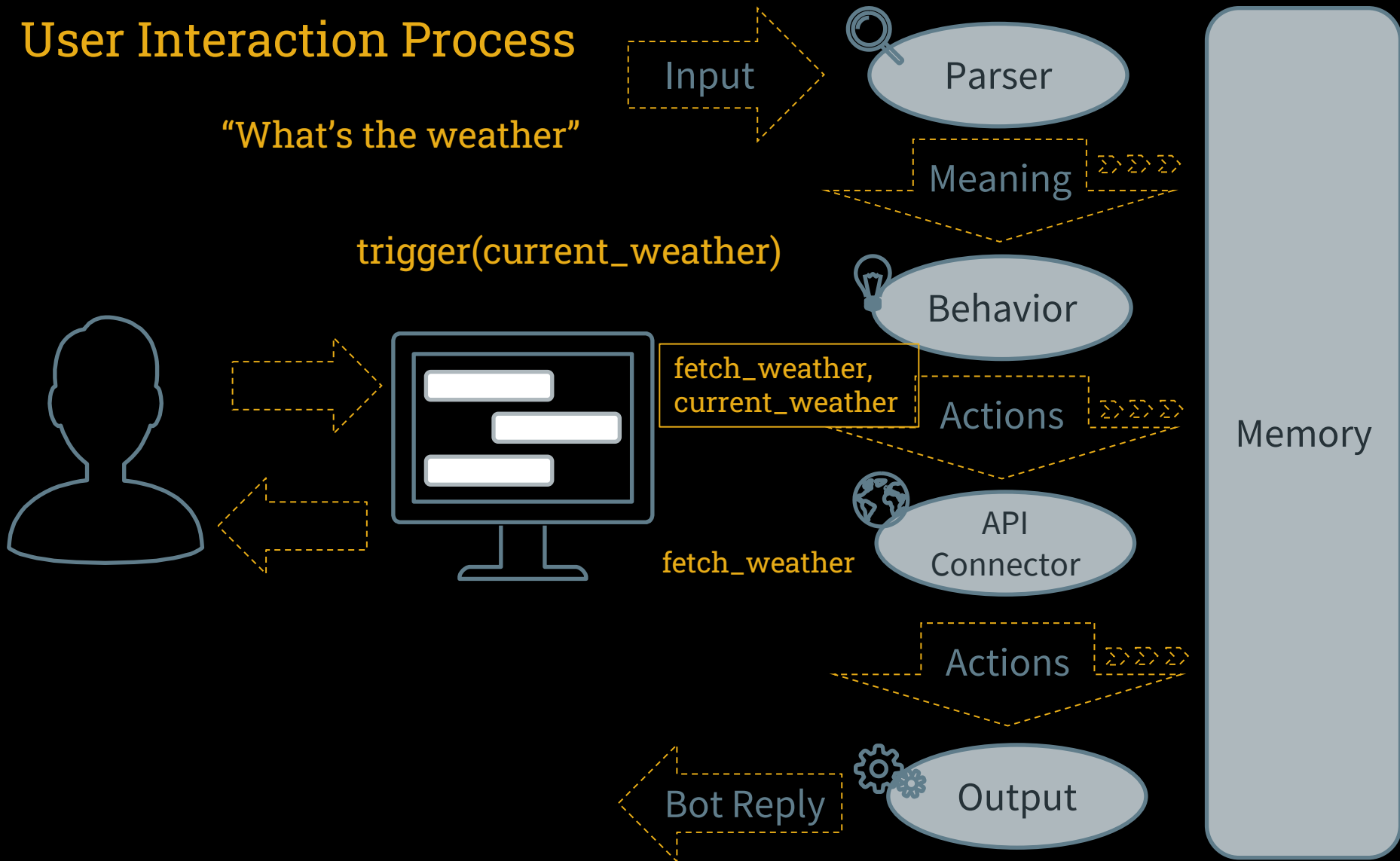
User Interaction Process

"What's the weather"



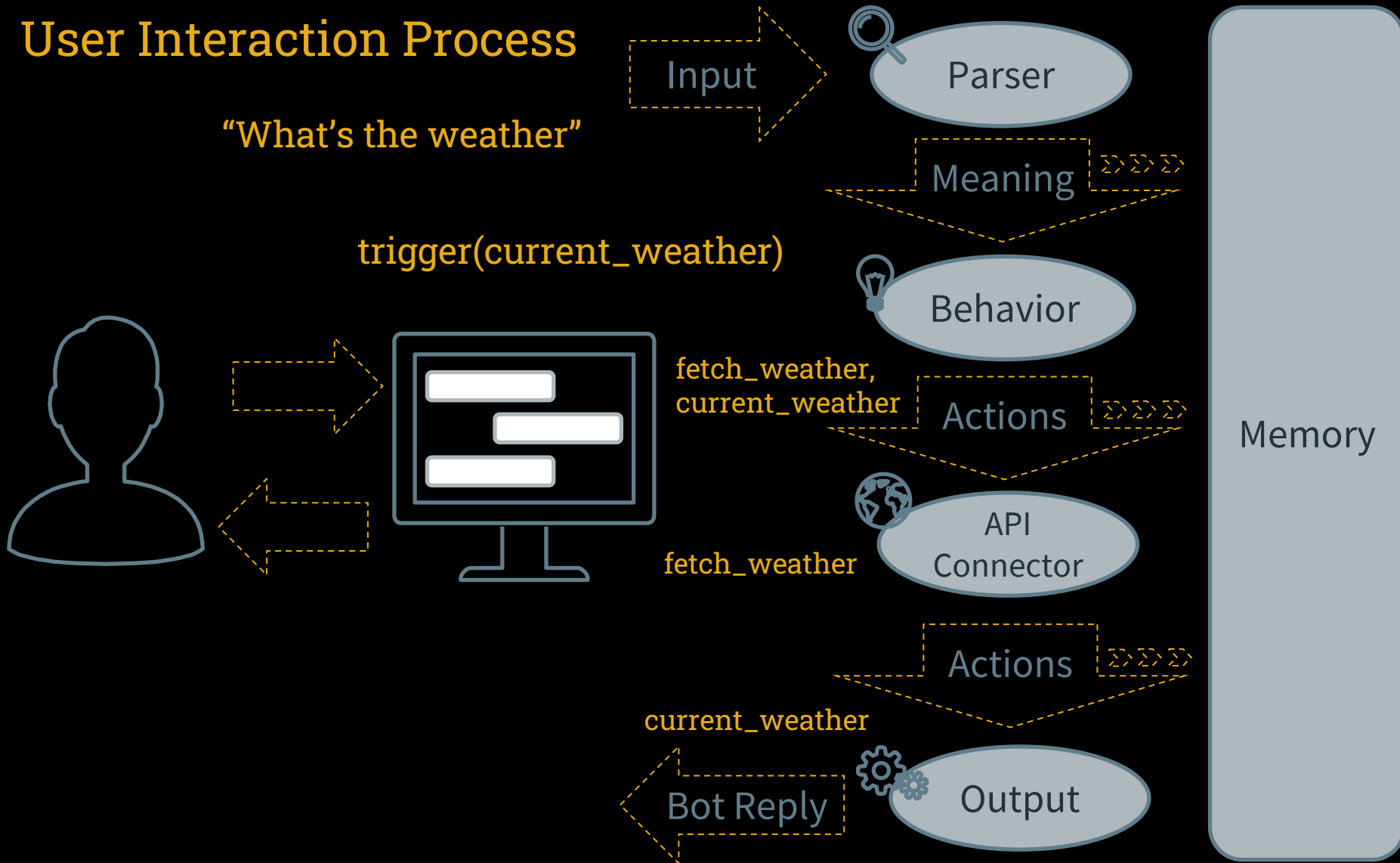
User Interaction Process

"What's the weather"



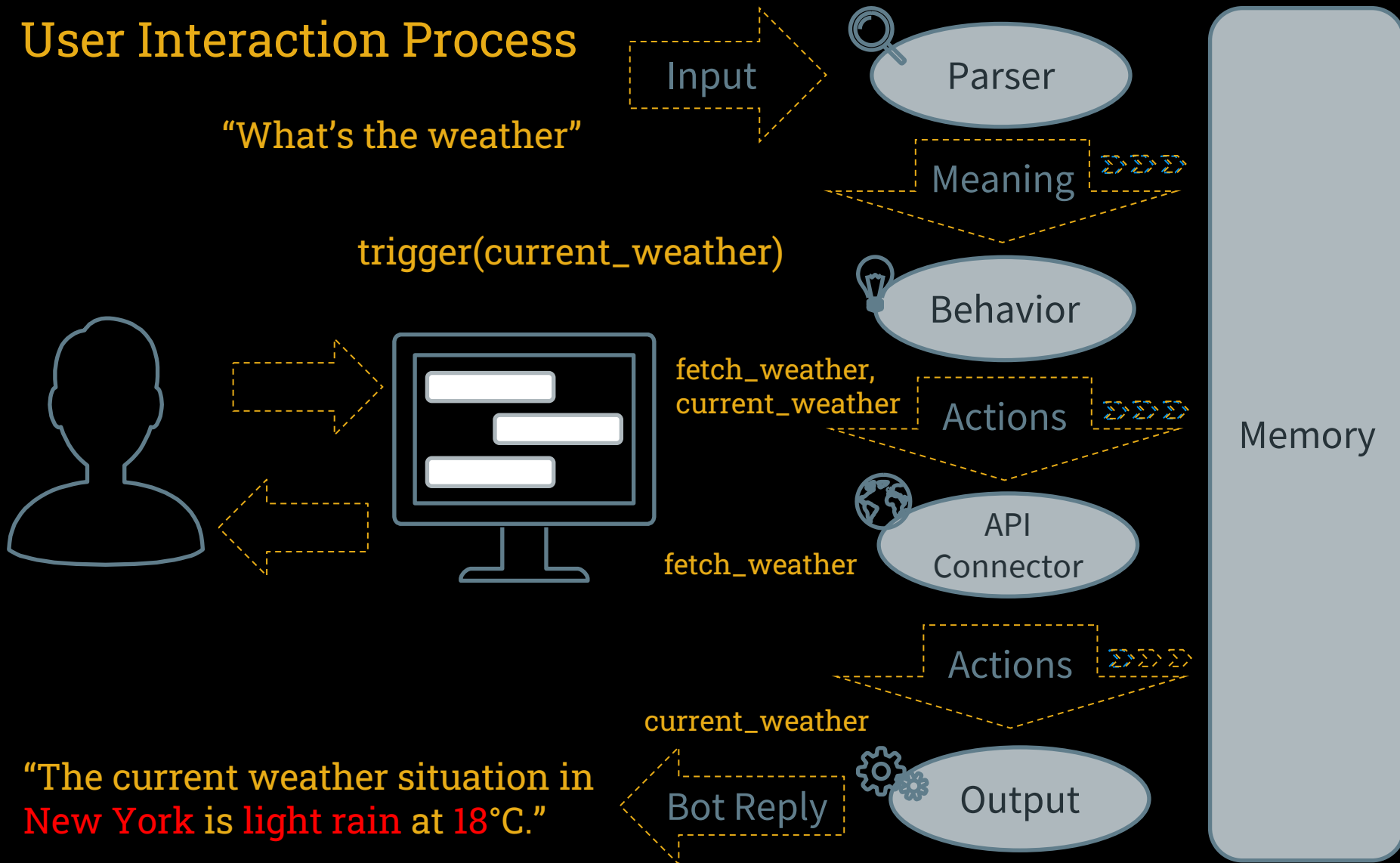
User Interaction Process

"What's the weather"



User Interaction Process

"What's the weather"



4.

JSON to Bot

How to configure Bots in EDDI

Dictionary

```
{
  "words": [
    {
      "word": "weather",
      "exp": "trigger(current_weather) "
    }
  ],
  "phrases": [
    {
      "phrase": "whats the weather",
      "exp": "trigger(current_weather) "
    }
  ]
}
```

Dictionary

```
{
  "words": [
    {
      "word": "weather",
      "expressions": "trigger(current_weather)"
    }
  ],
  "phrases": [
    {
      "phrase": "whats the weather",
      "expressions": "trigger(current_weather)"
    }
  ]
}
```


Behavior Rule (1)

```
{
  "behaviorGroups": [
    {
      "name": "Questions about weather",
      "behaviorRules": [
        {
          "name": "Ask for City",
          "actions": [
            "ask_for_city"
          ],
          "conditions": [
            {
              "type": "inputmatcher",
              "configs": {
                "expressions": "trigger(current_weather)"
              }
            }
          ]
        }
      ]
    }
  ]
}
```

Behavior Rule (1)

```
{
  "behaviorGroups": [
    {
      "name": "Questions about weather",
      "behaviorRules": [
        {
          "name": "Ask for City",
          "actions": [
            "ask_for_city"
          ],
          "conditions": [
            {
              "type": "inputmatcher",
              "configs": {
                "expressions": "trigger(current_weather)"
              }
            }
          ]
        }
      ]
    }
  ]
}
```

Behavior Rule (2)

```
{
  "behaviorGroups": [
    {
      "name": "Questions about weather",
      "behaviorRules": [
        {
          "name": "Current Weather in City",
          "actions": [
            "fetch_current_weather",
            "current_weather_in_city"
          ],
          "conditions": [
            {
              "type": "inputmatcher",
              "configs": {
                "expressions": "trigger(current_weather)",
                "occurrence": "lastStep"
              }
            }
          ]
        }
      ]
    }
  ]
}
```

Behavior Rule (2)

```
{
  "behaviorGroups": [
    {
      "name": "Questions about weather",
      "behaviorRules": [
        {
          "name": "Current Weather in City",
          "actions": [
            "fetch_current_weather",
            "current_weather_in_city"
          ],
          "conditions": [
            {
              "type": "inputmatcher",
              "configs": {
                "expressions": "trigger(current_weather)",
                "occurrence": "lastStep"
              }
            }
          ]
        }
      ]
    }
  ]
}
```

API Connector

```
{
  "targetServer": "api.openweathermap.org/data/2.5/weather",
  "httpCalls": [
    {
      "name": "currentWeather",
      "saveResponse": true,
      "responseObjectNames": "currentWeather",
      "actions": [
        "fetch_current_weather"
      ],
      "request": {
        "method": "get",
        "queryParams": {
          "APPID": "<APP-TOKEN-ID>",
          "units": "metric",
          "q": "[[${memory.current.input}]]"
        }
      }
    }
  ]
}
```

API Connector

```
{
  "targetServer": "api.openweathermap.org/data/2.5/weather",
  "httpCalls": [
    {
      "name": "currentWeather",
      "saveResponse": true,
      "responseObjectNames": "currentWeather",
      "actions": [
        "fetch_current_weather"
      ],
      "request": {
        "method": "get",
        "queryParams": {
          "APPID": "<APP-TOKEN-ID>",
          "units": "metric",
          "q": "[[${memory.current.input}]]"
        }
      }
    }
  ]
}
```

Templating Engine:



API Connector

```
{
  "targetServer": "api.openweathermap.org/data/2.5/weather",
  "httpCalls": [
    {
      "name": "currentWeather",
      "saveResponse": true,
      "responseObjectNames": "currentWeather",
      "actions": [
        "fetch_current_weather"
      ],
      "request": {
        "method": "get",
        "queryParams": {
          "APPID": "<APP-TOKEN-ID>",
          "units": "metric",
          "q": "[[${memory.current.input}]]"
        }
      }
    }
  ]
}
```

Templating Engine:



Output Templating

```
{
  "outputSet": [
    {
      "action": "current_weather_in_city",
      "outputs": [
        {
          "type": "text",
          "valueAlternatives": [
            "The current weather situation of [[${memory.current.input}]] is [[${memory.current.httpCalls.currentWeather.weather[0].description}]] at [[${memory.current.httpCalls.currentWeather.main.temp}]] °C"
          ]
        }
      ]
    }
  ]
}
```

Templating Engine:



Output Templating

```
{
  "outputSet": [
    {
      "action": "current_weather_in_city",
      "outputs": [
        {
          "type": "text",
          "valueAlternatives": [
            "The current weather situation of [[${memory.current.input}]] is [[${memory.current.httpCalls.currentWeather.weather[0].description}]] at [[${memory.current.httpCalls.currentWeather.main.temp}]] °C"
          ]
        }
      ]
    }
  ]
}
```

Templating Engine:



*What's the **weather** in **New York**?*



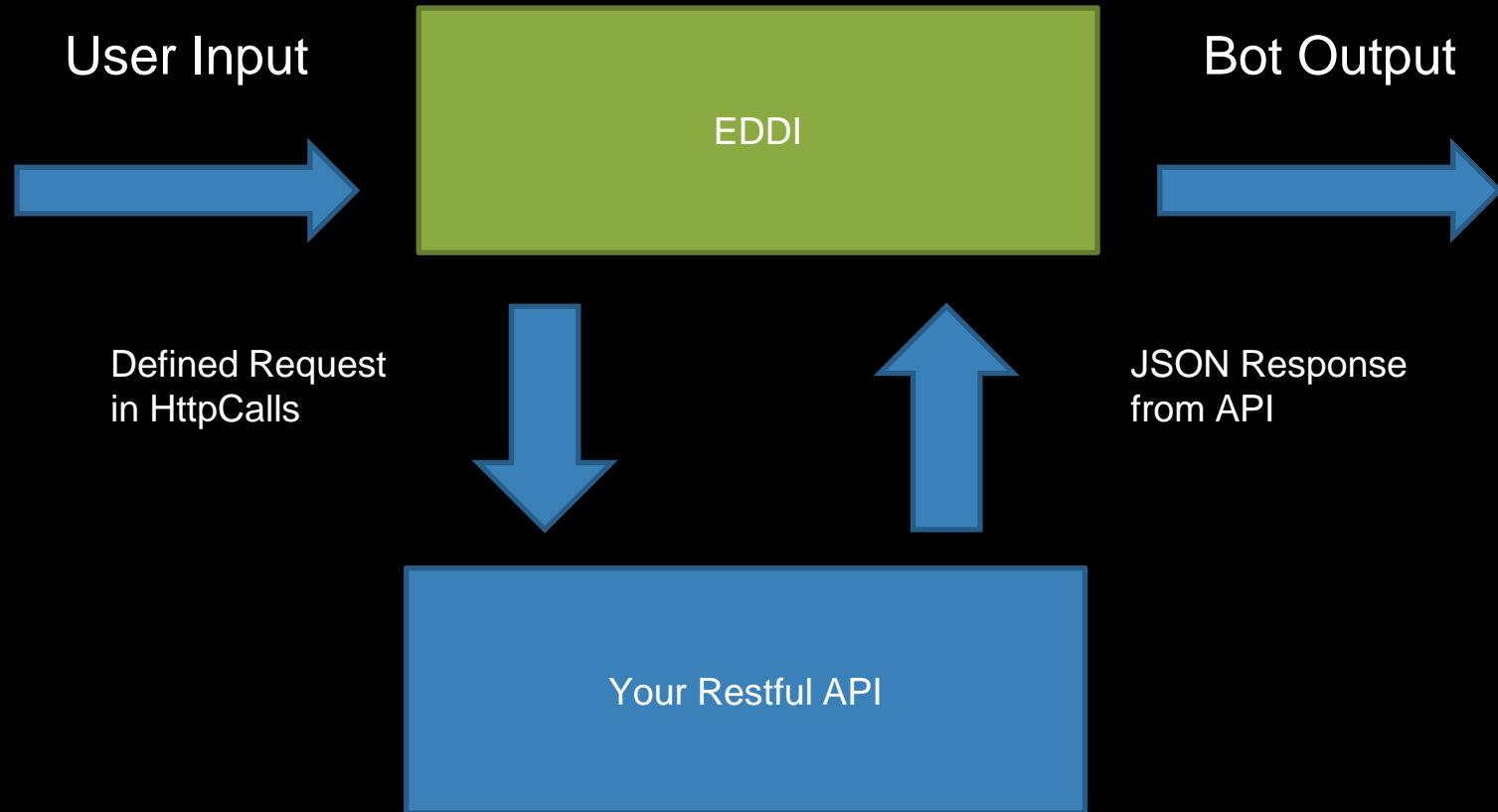
*The current weather situation in
Vienna is light rain at 18°C.*

5.

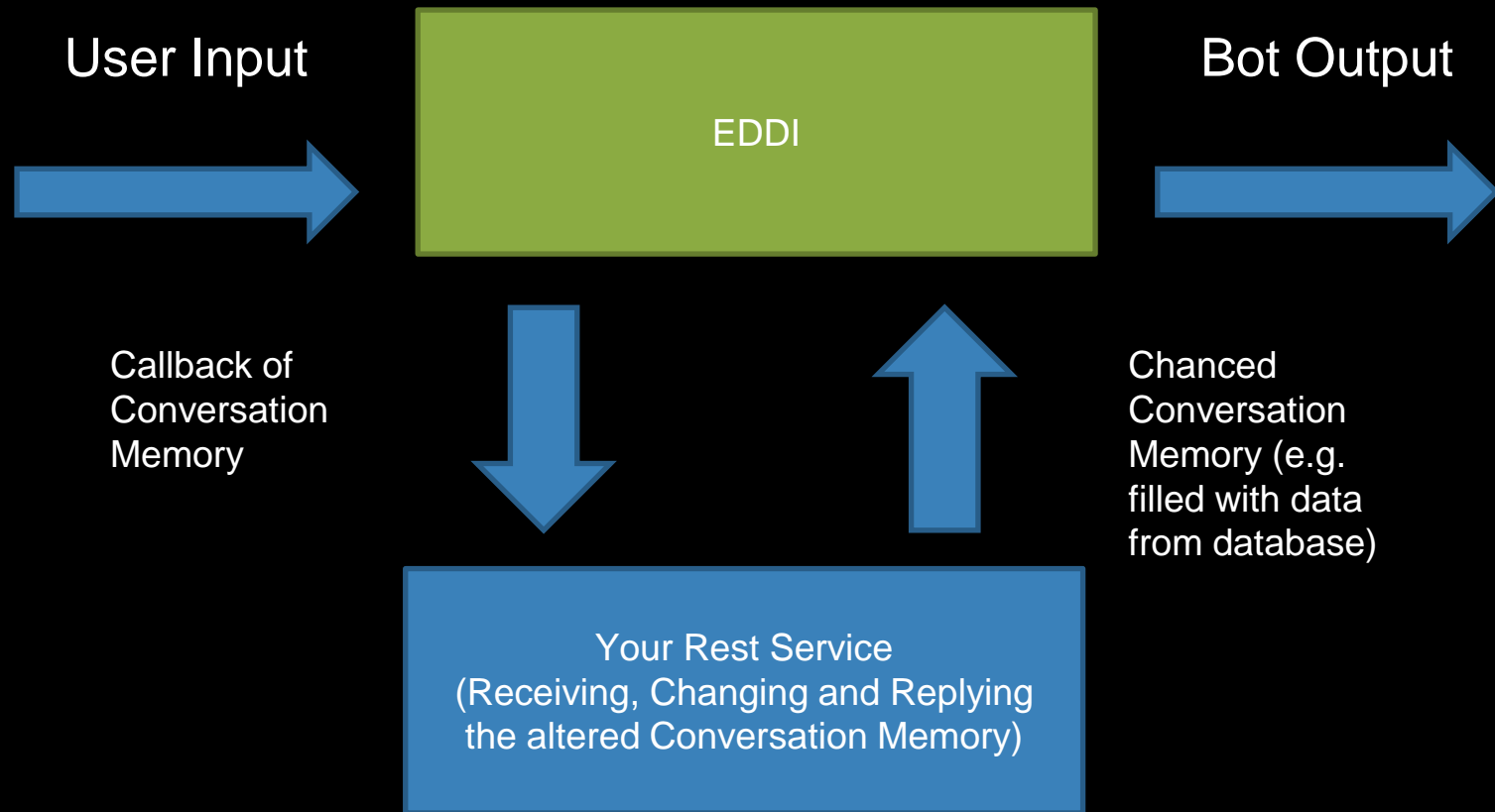
Possibilities to Code with & around EDDI

Different Entry Points into EDDI

Restful 1: Making EDDI use of Restful API



Restful 2: Send Conversation Memory to other Service



Java: Write a Plugin

© Extend NLP to extend the recognition of user inputs

- ai.labs.eddi.modules.nlp.extensions.dictionaries.IDictionary
- ai.labs.eddi.modules.nlp.extensions.corrections.ICorrection
- ai.labs.eddi.modules.nlp.extensions.normalizers.INormalizer

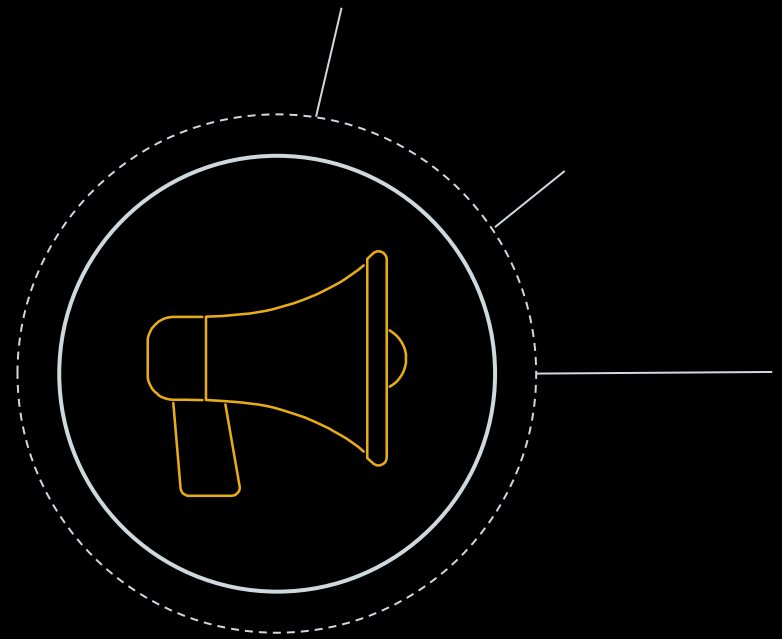
© Make new Conditions for Rules

- ai.labs.eddi.modules.behavior.impl.conditions.IBehaviorCondition

© Create new LifecycleTask (like NLP, Rules, Connector, etc)

- ai.labs.eddi.engine.lifecycle.ILifecycleTask

E.D.D.I



*“Spin up your own instance of EDDI
and create a ChatGPT powered Bot within a few minutes”*

Project Links:

Docs: **docs.labs.ai**

Website: **eddi.labs.ai**

GitHub: **github.com/labsai/eddi**

RH Marketplace:

marketplace.redhat.com/en-us/products/labsai