

BroadAI

Multi-Agent AI System (MAS)
Development Framework



65%

Executives report that AI is enabling organizations to stay ahead of the compete

63%

Executives view AI as critical to their business success

65% 68% 3%

Executives report that AI is
enabling organizations to
stay ahead of the compete

**Executives report moderate
to extreme skills gap**

Executives view AI as
critical to their business
success

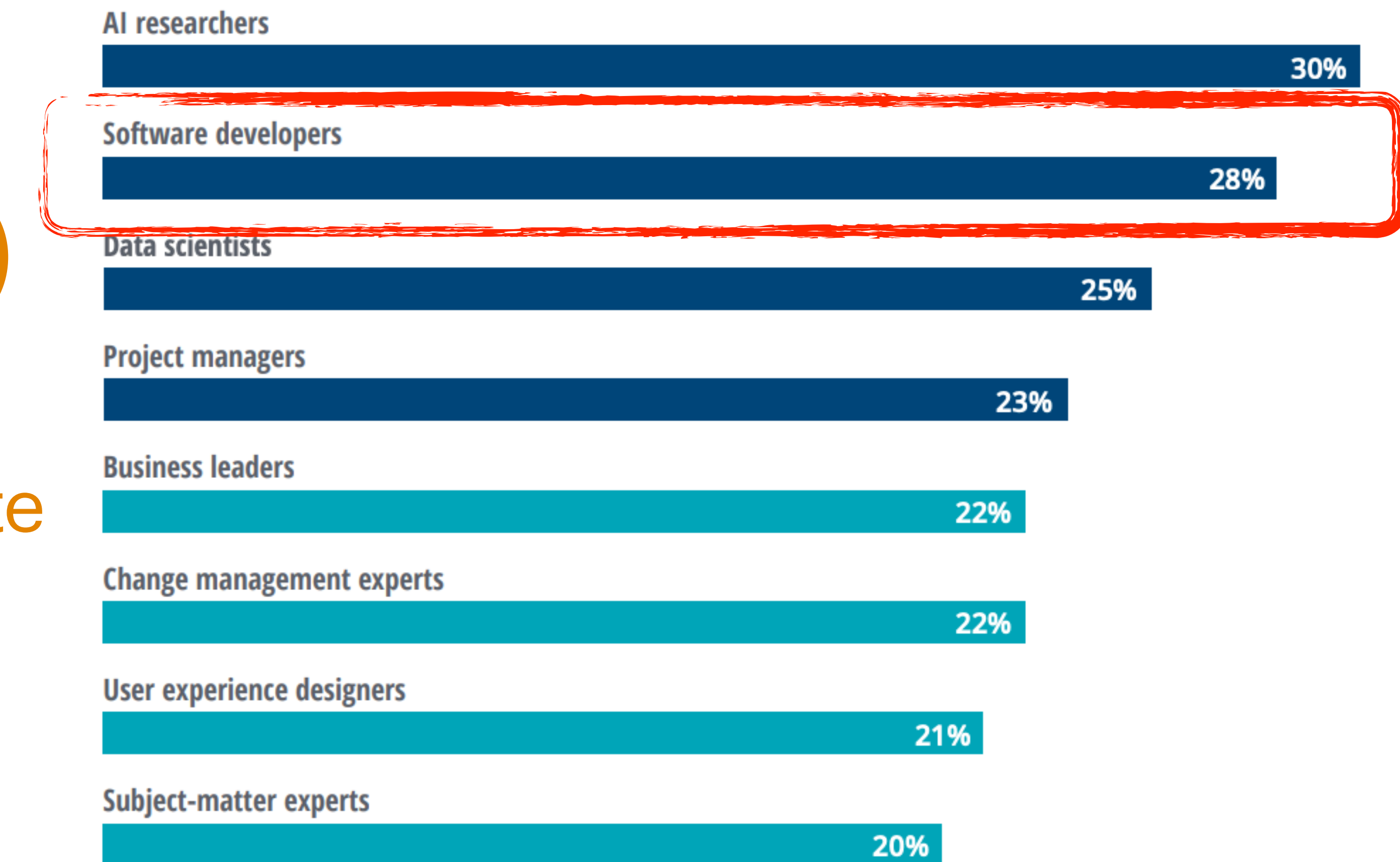
68%

Executives report moderate
to extreme skills gap

As companies strive to fill their AI skills gap, “AI builders” are the most sought-after professionals

Respondents rating each a top-two needed skill to fill their company's AI skills gap

■ AI builders ■ AI translators



68%

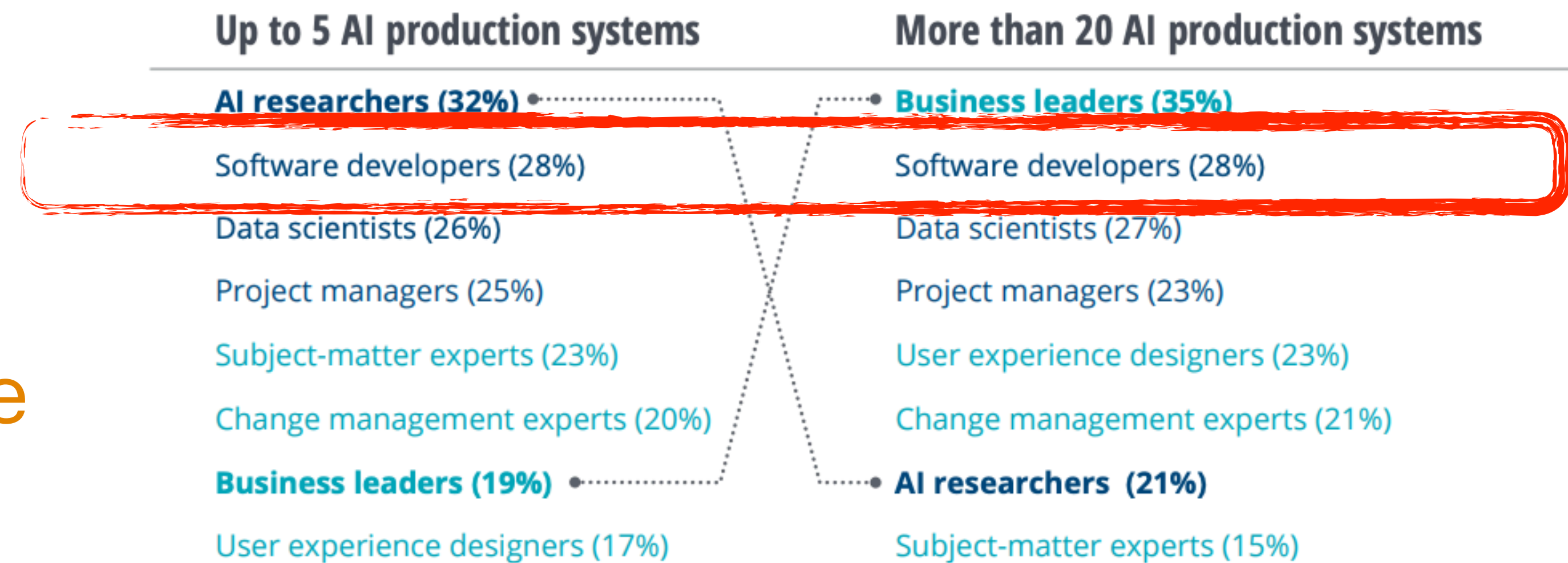
Executives report that AI is enabling organizations to stay ahead of the competition

Executives report moderate to extreme skills gap

As companies gain experience building AI systems, skill needs shift from a focus on “AI researchers” to a desire for “business leaders”

Respondents rating each a top-two needed skill to fill their company’s AI skills gap

■ AI builders ■ AI translators



Only if you are building highly customized or domain-specific AI solutions!

Many companies embarking on AI initiatives may feel they need to hire **AI superstars.**

Really?

How can skilled software developers build AI solutions?

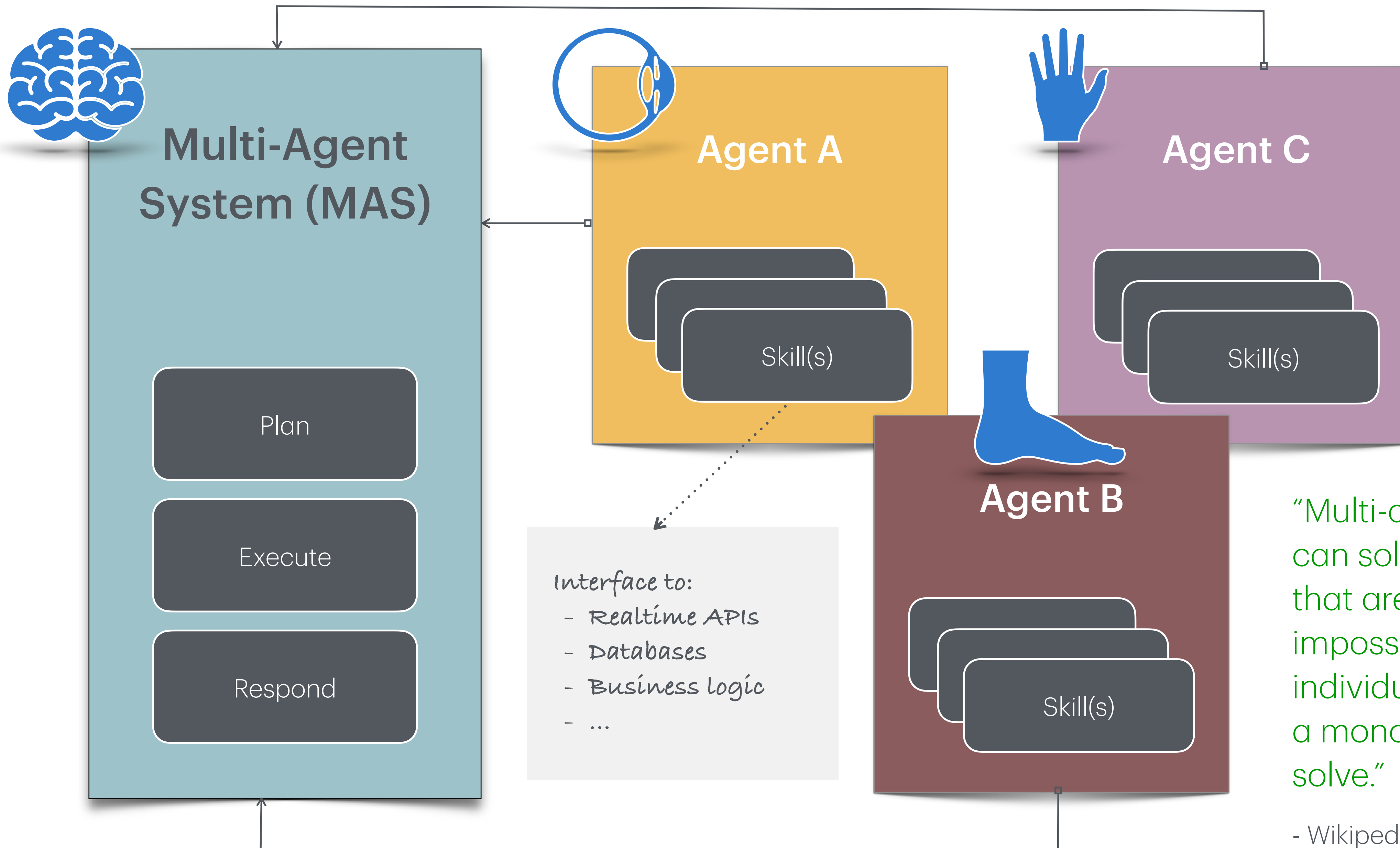
- > Machine Learning APIs
- > Cloud-based AI Services
- > **AI Frameworks & SDKs**

AI Agents are an autonomous system that can possess specific skills and execute tasks utilizing these skills.

BroadAI

Multi-Agent AI System (MAS) Development **Framework**

MAS is a system that can demonstrate working of a complex environment by utilizing multiple expert agents that collectively help achieve a given goal.

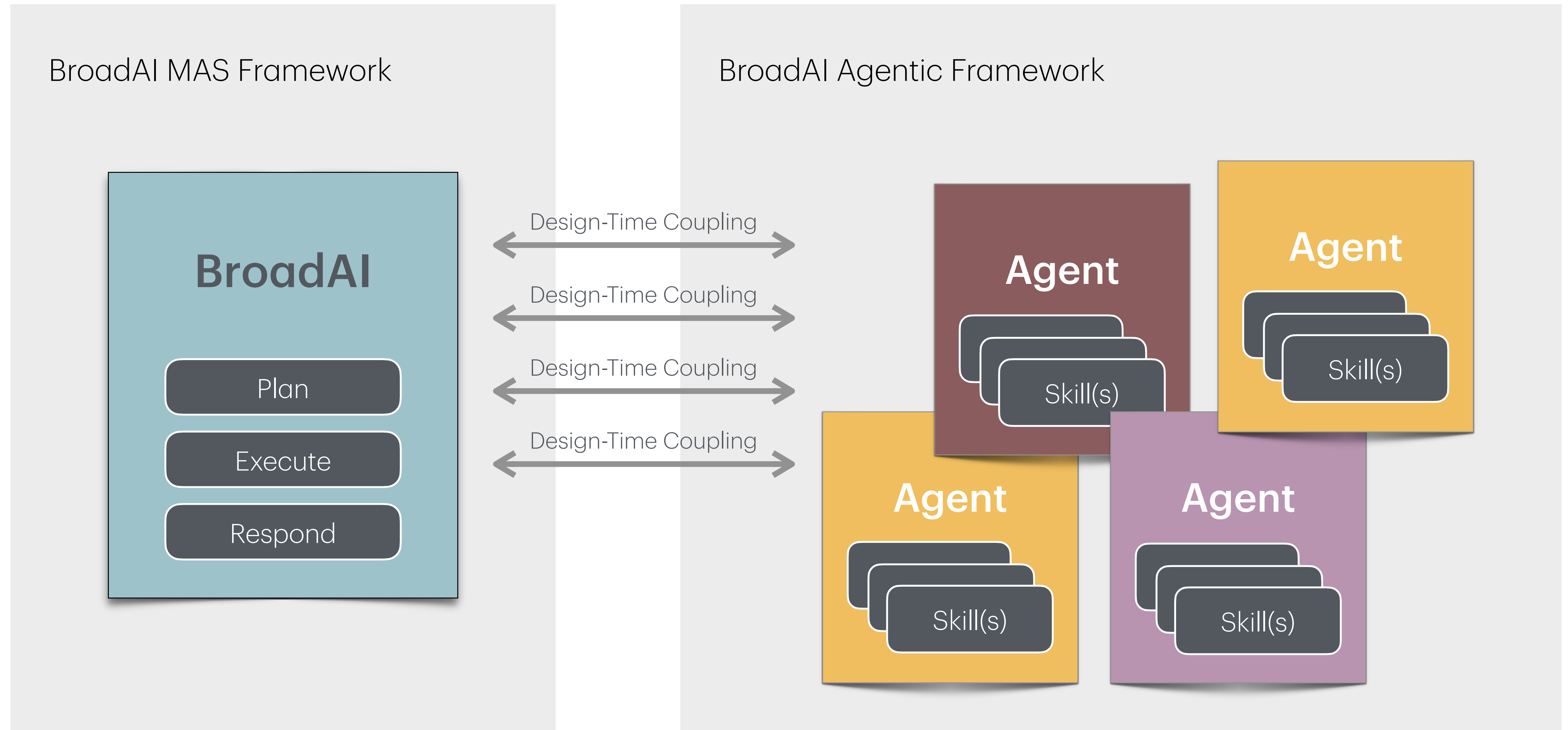


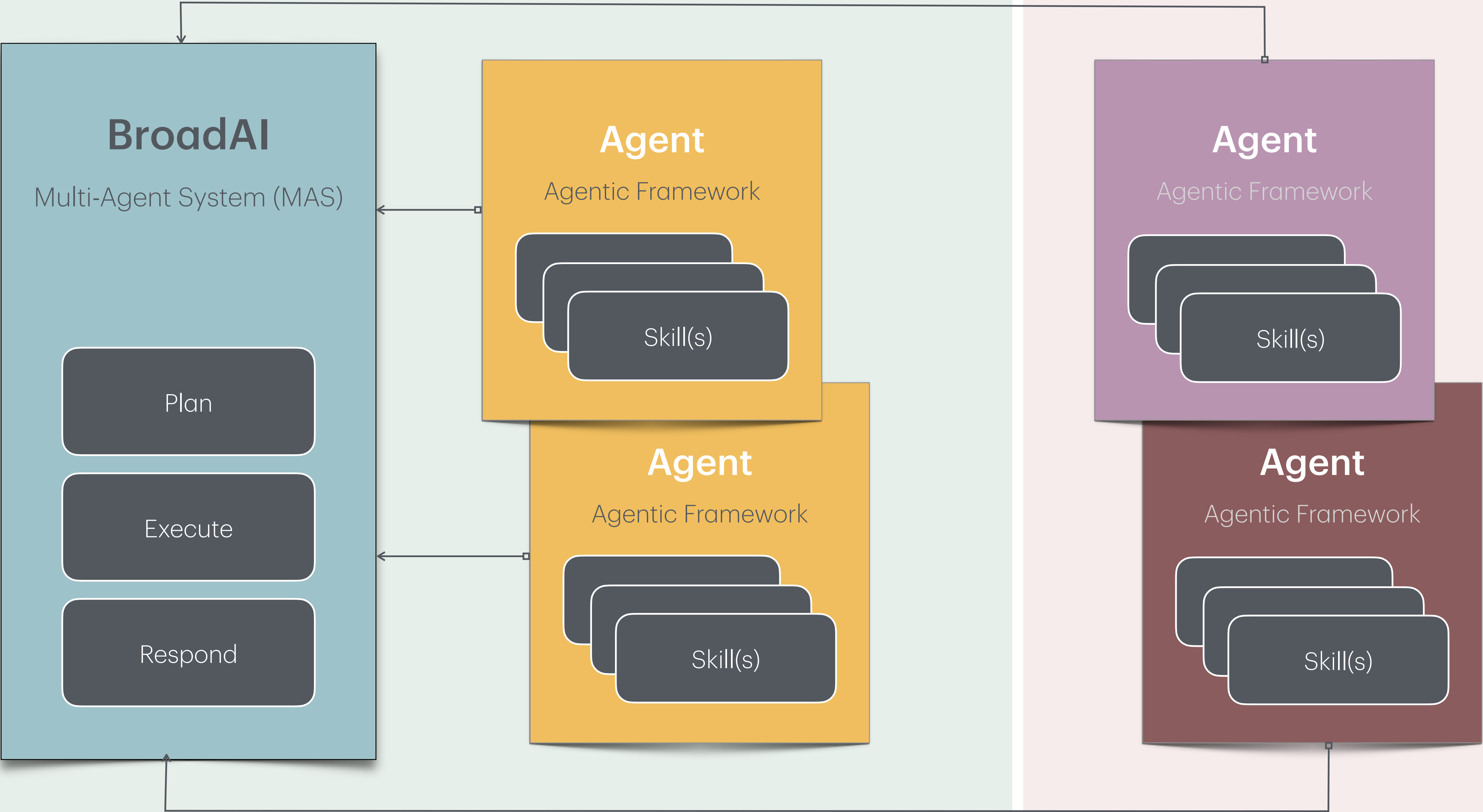
“Multi-agent systems can solve problems that are difficult or impossible for an individual agent or a monolithic system to solve.”

- Wikipedia

BroadAI

Multi-Agent AI System (MAS) Development Framework

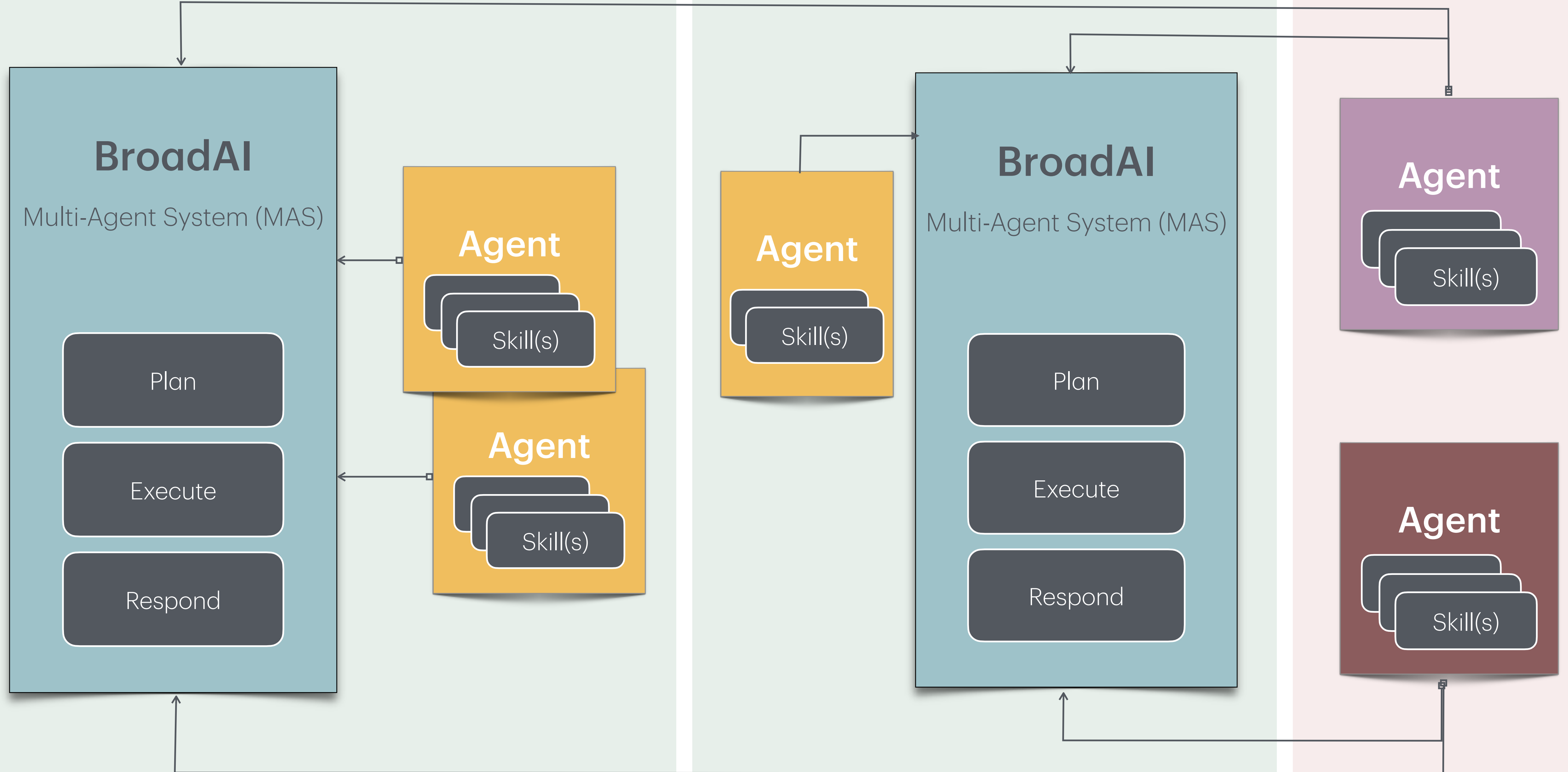




Private Space

Private Space

Public Space



How to Build with

BroadAI “MAS” Framework

BroadAI MAS Framework

to develop applications enhanced by AI capabilities

1 Register your application to receive an **appid**

2 Terminal
`npm install broadai broadai-agents --save`

3 Import **broadai** module in your application and create an instance (object)

Editor

```
import BroadAI from 'broadai';  
const ai = new BroadAI([/*agents*/], {/**/});
```

Where would we get these
agents from?



OPTION 1:

Use BroadAI Agent Store

OPTION 2:

Build a new BroadAI Agent
(and publish on BroadAI Agents Store)

BroadAI Agent Store

already-made agents to plug into your application

4 Visit <https://broad-ai.github.io/agent-store.html> and identify one or more agent(s) for your use-cases

Let's say we picked **Researcher** agent for our use case

5 Import the agent and notify BroadAI to use it at runtime

Editor

```
import BroadAI from 'broadai';  
import rs from 'broadai-agents/Researcher';  
const ai = new BroadAI([rs.agent], { });
```

* Note: You can import any number of agents and pass them in this array

6 Register the agent with **BroadAIConfig** and **agentConfig**

Editor

```
import BroadAI from 'broadai';  
import rs from 'broadai-agents/Researcher';  
const ai = new BroadAI([rs.agent], {/**/});  
rs.register(ai.config, {})
```

BroadAI MAS Framework

to develop applications enhanced by AI capabilities

Editor

```
import BroadAI from 'broadai';
import rs from 'broadai-agents/Researcher';

const ai = new BroadAI([rs.agent], {/**/});
rs.register(ai.config, {});
```

BroadAIConfiguration?

Four-part configuration:

- **appid** <- registered in 1st step
- **personality** <- to influence tone and authority
- **llm** <- (most important) BroadAI will use LLM of your choice. Comes in 2 flavors, **llm**, and **llmapi** for advanced / custom LLM configurations
- **history** <- so that BroadAI can emulate stateful conversation exchanges

Editor

```
import BroadAI from 'broadai';
import rs from 'broadai-agents/Researcher';

const ai = new BroadAI([rs.agent], {
  appid: "dswZsbf",
  personality: {
    name: "Jack", role: "assistant",
    pov: "first-person"
  },
  llm: {
    provider: "openai",
    model: "gpt-3.5-turbo-0125",
    apikey: env['OPENAI_KEY']
  },
  history: {
    enabled: false, max_exchanges: 5
  }
});
rs.register(ai.config, {/**/});
```

BroadAI MAS Framework

to develop applications enhanced by AI capabilities

Editor

```
import BroadAI from 'broadai';
import rs from 'broadai-agents/Researcher';
const ai = new BroadAI([rs.agent], {
  appid: "dswZsbf",
  personality: { name: "Jack",
    role: "assistant", pov: "first-person" },
  llm: {
    provider: "openai",
    model: "gpt-3.5-turbo-0125",
    apikey: env['OPENAI_KEY']
  },
  history: {
    enabled: false, max_exchanges: 5
  }
});
rs.register(ai.config, {/**/});
```

7

Finally, simply call **go()** method to invoke Generative AI capability on behalf of your application

Editor

```
ai.go(question, true, []).then((resp)=>{
  console.log(resp);
});
```

go() works like this:

- Based on the **question**, BroadAI generates an objective and a **plan** to achieve that objective
- If the 2nd parameter is **true**, BroadAI **audits** the **plan** to ensure it can accurately and optimally achieve the objective.
- The updated **plan** is a series of steps comprising of identified agents and skills that can provide incremental results to the overall objective.
- This **plan** is executed by BroadAI and results are collected..
- Collected results are used to ground the last step which also takes any past **conversations** into account to generate the final **response** to user's original **question**.

BroadAI MAS Framework

to develop applications enhanced by AI capabilities

Question: Find fast food joints around Columbus airport

Terminal

Plan ...

... with agents and skills

```
{
  plan: {
    status: 'complete',
    plan: [
      {
        sequence: 0,
        objective: 'Search for fast food joints around Columbus airport',
        agent: 'Researcher',
        skill: {
          name: 'findBusinesses',
          parameters: { term: 'fast food', location: 'CMH' }
        },
        result: "Raising Cane's Chicken Fingers (3.4 stars) | 1320 N Hamilton Rd Gahanna, OH 43230 Ph: (614) 475-1300[alias: raising-canes-chicken-fingers-gahanna]\n" +
          'Chick-fil-A (3.2 stars) | 3940 Morse Rd Columbus, OH 43219 Ph: (614) 414-7222[alias: chick-fil-a-columbus-22]\n' +
          'Chick-fil-A (3.3 stars) | 6240 E Broad St Columbus, OH 43213 Ph: (614) 861-2300[alias: chick-fil-a-columbus-5]\n' +
          'MrBeast Burger (3.3 stars) | Columbus, OH 43219 Ph: (877) 232-7898[alias: mrbeast-burger-columbus-4]\n' +
          'Sonic Drive-In (1.9 stars) | 2846 Stelzer Rd Columbus, OH 43219 Ph: (614) 532-5738[alias: sonic-drive-in-columbus-17]\n' +
          'The Mad Greek (4.2 stars) | 4210 E Broad St Columbus, OH 43213 Ph: (614) 338-0000[alias: the-mad-greek-columbus]'
      }
    ],
    reason: "The plan involves using the Researcher agent to find fast food joints around Columbus airport, fulfilling the user request. Updated the location parameter to the IATA code 'CMH' for Columbus airport to ensure accurate search results."
  },
}
```

Explainability

Terminal

Response ...

... with formatting tags and response

```
response: {
  status: 'complete',
  response: [
    { html_tag: 'h2', text: 'Fast Food Joints Around Columbus Airport' },
    { html_tag: 'p', text: 'Based on the search results, here are some fast food joints around Columbus airport.' },
    { html_tag: 'h3', text: '1. Raising Cane's Chicken Fingers (3.4 stars)' },
    { html_tag: 'p', text: 'Location: 1320 N Hamilton Rd Gahanna, OH 43230' },
    { html_tag: 'p', text: 'Phone: (614) 475-1300' },
    { html_tag: 'h3', text: '2. Chick-fil-A (3.2 stars)' },
    { html_tag: 'p', text: 'Location: 3940 Morse Rd Columbus, OH 43219' },
    { html_tag: 'p', text: 'Phone: (614) 414-7222' },
    { html_tag: 'h3', text: '3. The Mad Greek (4.2 stars)' },
    { html_tag: 'p', text: 'Location: 4210 E Broad St Columbus, OH 43213' },
    { html_tag: 'p', text: 'Phone: (614) 338-0000' }
  ],
  reason: 'The response utilizes the provided plan results to list the fast food joints around Columbus airport.',
  conversation: [
    '> Q: Find fast food joints around Columbus airport',
    '> A: \n' +
      '## Fast Food Joints Around Columbus Airport: \n' +
      'Based on the search results, here are some fast food joints around Columbus airport: \n' +
      '### 1. Raising Cane's Chicken Fingers (3.4 stars): \n' +
      'Location: 1320 N Hamilton Rd Gahanna, OH 43230 Phone: (614) 475-1300 \n' +
      '### 2. Chick-fil-A (3.2 stars): \n' +
      'Location: 3940 Morse Rd Columbus, OH 43219 Phone: (614) 414-7222 \n' +
      '### 3. The Mad Greek (4.2 stars): \n' +
      'Location: 4210 E Broad St Columbus, OH 43213 Phone: (614) 338-0000 '
  ]
}
```

Explainability

Conversation history for stateful application use-cases

How to Build with

BroadAI “Agentic” Framework

BroadAI MAS Framework

to develop applications enhanced by AI capabilities

1 Register your application to receive an **appid**

2 Terminal
`npm install broadai broadai-agents --save`

3 Import **broadai** module in your application and create an instance (object)

Editor

```
import BroadAI from 'broadai';  
const ai = new BroadAI([/*agents*/], { });
```

Where would we get these
agents from?



OPTION 1:
Use BroadAI
Agent Store

OPTION 2:
Build a new BroadAI Agent (and
publish on BroadAI Agents Store)

BroadAI Agentic Framework

To develop and publish BroadAI Agents

1 Start with an empty npm project to create a new agent

2 Terminal
`npm install broadai --save`

3 Import **broadai/agentic** module in your project and create placeholders for Agentic instance and BroadAI configuration (*more on this to come!*)

Editor

```
const Agentic = require('broadai/agentic');  
let agentic;  
let config;
```

BroadAI Agentic Framework

To develop and publish BroadAI Agents

4 Create a module with **register** method and **agent** property

Editor

```
const Agentic = require('broadai/agent');
let agentic;
let config;
module.exports = {
```

```
  // - register()
```

```
  register: (broadAIConfig, agentConfig)=>{
    agentic = new Agentic(broadAIConfig);
    config = agentConfig;
  },
```

```
  // - agent
```

```
  agent: {
```

From BroadAI MAS application:

```
const ai = new BroadAI([rs.agent], {});
rs.register(ai.config, {});
```

5

Define the **agent** which includes it's **skills** and **actions** associated with those skills

Editor

```
...
// - agent
agent: {
  // name your agent to the likes of a persona, e.g. StockBroker, Analyst,
  // Linguist, etc.
  "agent-name": "--??--",
  // describe what kind of overall capabilities this agent will provide.
  // Better description will help BroadAI MAS utilize it at the right place.
  "capability": "--??--",
  // list of all the skills this agent possesses.
  "skills": [
    {
      // name the skill, e.g. translateText, composeEmail, etc.
      "skill-name": "skillName",
      // describe what the skill will help achieve.
      "objective": "--??--",
      // MANDATORY: Must match the skill-name
      "action": skillName,
      // parameters the action needs to execute
      "parameters": {
        // MANDATORY: Must match param used in the function later
        "param": "--data-type-- | --description--"
        // RECOMMENDATION: 'param' value must describe the data-type
        // expected followed by brief description and possibly an example. Better
        // description and examples will help BroadAI MAS determine appropriate parameter
        // values so that the function can execute completely.
      }
    },
    /* add additional skills using the same structure above */
  ],
}
...

```

BroadAI Agentic Framework

To develop and publish BroadAI Agents

Editor

```
...
// - agent
agent:{
  // name your agent to the likes of a persona, e.g. StockBroker,
  // Analyst, Linguist, etc.
  "agent-name": "--??--",
  // describe what kind of overall capabilities this agent will
  // provide. Better description will help BroadAI MAS utilize it at the
  // right place.
  "capability": "--??--",
  // list of all the skills this agent possesses.
  "skills": [
    {
      // name the skill like a function, e.g. translateText,
      // composeEmail, etc.
      "skill-name": "skillName",
      // describe what the skill will help achieve. Better
      // description will help BroadAI MAS utilize it at the right place.
      "objective": "--??--",
      // MANDATORY: Must match the skill-name
      "action": skillName,
      // parameters the action needs to execute
      "parameters": {
        // MANDATORY: Parameter name used in the function definition
        // above
        "param": "--data-type-- | --description and possibly an
        // example--"
        // RECOMMENDATION: 'param' value must describe the data-type
        // expected followed by brief description and possibly an example. Better
        // description and examples will help BroadAI MAS determine appropriate
        // parameter values so that the action can execute completely.
      }
    },
    /* add additional skills using the same structure above */
  ]
}
...
```

6

Define the **action**, which is essentially a NodeJS function

Editor

```
const skillName = (param) => {
  // RECOMMENDATION: returns Promise
  return new Promise((resolve, reject) => {

    // ... You may use Agentic accelerator methods
    // ... > to call external API resource, or
    // ... > to call LLM using BroadAI's configuration
    // ... depending on your use case.
    // ... Alternatively, you are free to
    // ... write the entire logic on your own.

    resolve();
  });
};
```

7

Export as **npm package** and:

- use it as a **private agent**, or
- send it to BroadAI to publish as a **public agent** so that developers can plug-and-play it into their applications

Demonstration

Thank You!

BroadAI

Multi-Agent AI System (MAS) Development Framework

<https://broad-ai.github.io/>

