

MS-4014: AI エージェントを構築し、 Microsoft 365 Copilotを拡張するための ~~基盤を構築する~~ 全体像を理解する

全4モジュール

本資料の内容は以下のMicrosoft Learnコンテンツに準拠しています。



AI エージェントを構築し、Microsoft 365 Copilotを拡張するための基盤を構築する

モジュール4
カスタムエンジンエージェント開発

MS-4014: AI エージェントを構築し、
Microsoft 365 Copilotを拡張するための
基盤を構築する
全体像を理解する

モジュール3
Microsoft 365 Copilotの
拡張・カスタマイズ

モジュール1
エージェントの基礎の解説
モジュール2
MicrosoftのAIエージェント関連テクノロジー解説

MS-4014: エージェント開発のための
多数のテクノロジの概要を理解し
適切なものを選ぶようにする

Microsoft 365 Copilotの カスタマイズや エージェント開発に関する コースのご紹介

concept learning

MS-4014 Build a foundation to
extend Microsoft 365 Copilot

今回解説するMS-4014では
エージェント開発のための基礎知
識、全体像を俯瞰的に解説。
演習なし、座学のみ。

using No-code experience

MS-4019 Create a no-code agent for Microsoft 365 SharePoint
(Transform your everyday business processes with agents)
[Create agents in **Copilot Chat** / **SharePoint**]

using Visual Studio Code

MS-4001 ~~Build collaborative apps for Microsoft Teams~~
retired 2 Nov, 2024, replacement: none

MS-4010 Extend Microsoft 365 Copilot with declarative agents by using VSCode
[Teams Toolkit extension]

MS-4015 ~~Build custom agents for Microsoft Teams~~
[Teams AI Library] (deleted from ESI catalog)

using Microsoft Copilot Studio

MS-4009 ~~Extend Microsoft 365 Copilot with Copilot Studio actions and connectors~~
retired 31 Mar, 2025. replacement: MS-4022

MS-4022 Extend Microsoft 365 Copilot in Copilot Studio

PL-7008 Create agents with Copilot Studio

using Azure AI Foundry / Azure OpenAI

AI-3026 Develop AI Agents in Azure [Azure AI Agent Service]

AZ-2005 Develop generative AI Apps with Azure OpenAI and Semantic Kernel



Microsoft 365 Copilotの カスタマイズや エージェント開発に関する コースのご紹介

concept learning

MS-4014

Build a foundation to
extend Microsoft 365 Copilot

それぞれのツールを使った、より
具体的な開発方法については、対
応するコースがあります（演習あ
り）。ぜひ今回のご受講と合わせ
てこれらの後続コースの受講もご
検討ください。
これら後続コースについて詳しく
は**講義終了時に**解説いたします。

using No-code experience

MS-4019

Create a no-code agent for Microsoft 365 SharePoint
(Transform your everyday business processes with agents)
[Create agents in **Copilot Chat** / **SharePoint**]

using Visual Studio Code

MS-4001

~~Build collaborative apps for Microsoft Teams~~
retired 2 Nov, 2024, replacement: none

MS-4010

Extend Microsoft 365 Copilot with declarative agents by using VSCode
[Teams Toolkit extension]

MS-4015

~~Build custom agents for Microsoft Teams~~
[Teams AI Library] (deleted from ESI catalog)

using Microsoft Copilot Studio

MS-4009

~~Extend Microsoft 365 Copilot with Copilot Studio actions and connectors~~
retired 31 Mar, 2025. replacement: MS-4022

MS-4022

Extend Microsoft 365 Copilot in Copilot Studio

PL-7008

Create agents with Copilot Studio

using Azure AI Foundry / Azure OpenAI

AI-3026

Develop AI Agents in Azure [Azure AI Agent Service]

AZ-2005

Develop generative AI Apps with Azure OpenAI and Semantic Kernel



MS-4014 モジュール構成

- モジュール1 AIエージェントの概要
- モジュール2 マイクロソフトのAIエージェント関連テクノロジー
- モジュール3 Microsoft 365 Copilotのカスタマイズ方法
- モジュール4 カスタムエンジンエージェントの開発

モジュール1 AIエージェントの概要



AI エージェントの開発の概要

AI エージェントとは何かを調べて、コア コンポーネントを理解し、AI を活用した変革を推進するさまざまなインパクトのあるユース ケースを見つけ出します。



モジュール1 目次

- AIエージェントとは？
- 従来のチャットボットとAIエージェントは何が違うのか？
- なぜ今AIエージェントが着目されているのか？
- エージェントはビジネスでどのように役に立つのか？
- お客様事例
- AIエージェントの内部構成
- まとめ

モジュール1 目次

- AIエージェントとは？
- 従来のチャットボットとAIエージェントは何が違うのか？
- なぜ今AIエージェントが着目されているのか？
- エージェントはビジネスでどのように役に立つのか？
- お客様事例
- AIエージェントの内部構成
- まとめ

AIエージェントとは？

- 「エージェント」 (Agent) = 代理人、仲介者。
- 目的達成のために自律的に行動するAIソフトウェア
- 単に質問への回答を行うだけではなく、問題解決に向けて能動的に動く
- ユーザーの指示を理解し、複数のタスクを実行できる

モジュール1 目次

- AIエージェントとは？
- 従来のチャットボットとAIエージェントは何が違うのか？
- なぜ今AIエージェントが着目されているのか？
- エージェントはビジネスでどのように役に立つのか？
- お客様事例
- AIエージェントの内部構成
- まとめ

従来のチャットボットとAI エージェント は何が違うのか？

- 従来のチャットボットはあらかじめプログラミングされたことしかできなかった
 - 決まったパターンでの質問や指示にしか応答できなかった
- AIエージェントは文脈を理解し、柔軟にタスクを遂行する
 - 生成AIにより、ユーザーの指示を柔軟に理解して作業を実行
 - 複雑な指示や複数ステップの処理にも対応できる

モジュール1 目次

- AIエージェントとは？
- 従来のチャットボットとAIエージェントは何が違うのか？
- なぜ今AIエージェントが着目されているのか？
- エージェントはビジネスでどのように役に立つのか？
- お客様事例
- AIエージェントの内部構成
- まとめ

なぜ今AIエージェントが着目されているのか？

- 生成AIの進化により、高度な推論と自律性が実現
 - OpenAI o1などの高度な推論能力を持つAIが登場
- エージェントを実装するための標準化が進んだ
 - MCP（Model Context Protocol）：エージェントが外部ツールを呼び出す仕組み。Anthropicが提唱。多くのAIツールで対応が行われた。Microsoftの製品でも対応が進行しており、例えばMicrosoft 365 CopilotではMCPサーバーと通信するカスタムコネクタを利用できる。
 - A2A（Agent to Agent Protocol）：エージェント間のコラボレーションを行うためのプロトコル。Googleが提唱。50社以上が支持を表明。Microsoftの製品でもこのプロトコルへの対応を行う予定。
- Azure AI Foundry Agent ServiceやSemantic Kernelなど、エージェントの開発・運用を行うためのサービスやSDKが充実してきた

■ Teams AI Libraryを使用したエージェントの開発についての解説動画。MCPやA2Aに対応するエージェントを比較的簡単に開発できるようになった。

YouTube JP

teams ai library

× 🔍 🎤

EXPLORER

UNTITLED (WORKSPACE)

- a2ademo
 - appPackage
 - color.png
 - manifest.json
 - outline.png
 - src
 - TS handler.ts
 - TS index.ts
 - package.json
 - .vscode
 - settings.json

TS index.ts

TS handler.ts

```
1 a2ademo > src > TS index.ts > ...
2 import { handleA2AMessage } from './handler';
3
4
5 // A2A Agent Card
6 const standupAgentCard: AgentCard = {
7   name: "Standup Agent",
8   description: "An agent that manages standups and helps teams keep on track with important priorities for the next discussion.",
9   url: "http://localhost:3000/a2a",
10  provider: { ...
11  },
12  version: "0.0.1",
13  capabilities: {},
14  skills: [
15    {
16      id: "add_parking_lot",
17      name: "Add to Parking Lot",
18      description: "Add a new item to the parking lot for discussion.",
19      tags: ["parking-lot", "add"],
20      examples: [ ...
21      ],
22    },
23    {
24      id: "get_parking_lot",
25      name: "Get Parking Lot Items",
26      description:
27        "Retrieve all current parking lot items for a given conversation id",
28      tags: ["parking-lot", "list"],
29      examples: [ ...
30      ],
31    },
32  ],
33 };
34
35 const app = new App({
36   name: "Standup Agent",
37   url: "http://localhost:3000/a2a",
38   provider: { ...
39   },
40   version: "0.0.1",
41   capabilities: {},
42   skills: [
43     {
44       id: "add_parking_lot",
45       name: "Add to Parking Lot",
46       description: "Add a new item to the parking lot for discussion.",
47       tags: ["parking-lot", "add"],
48       examples: [ ...
49       ],
50     },
51     {
52       id: "get_parking_lot",
53       name: "Get Parking Lot Items",
54       description:
55         "Retrieve all current parking lot items for a given conversation id",
56       tags: ["parking-lot", "list"],
57       examples: [ ...
58       ],
59     },
60   ],
61 });
62
63 new A2APlugin({
64   agentCard: standupAgentCard,
65   app: app,
66 });
67
68 // Teams Message Handler
69 const handleA2AMessage = async (message: A2AMessage) => {
70   // ...
71   return {
72     type: "A2A",
73     data: {
74       id: "add_parking_lot",
75       name: "Add to Parking Lot",
76       description: "Add a new item to the parking lot for discussion.",
77       tags: ["parking-lot", "add"],
78       examples: [ ...
79       ],
80     },
81   };
82 };
83
84 export { handleA2AMessage };
85
86 const app = new App({
87   name: "Standup Agent",
88   url: "http://localhost:3000/a2a",
89   provider: { ...
90   },
91   version: "0.0.1",
92   capabilities: {},
93   skills: [
94     {
95       id: "add_parking_lot",
96       name: "Add to Parking Lot",
97       description: "Add a new item to the parking lot for discussion.",
98       tags: ["parking-lot", "add"],
99       examples: [ ...
100      ],
101    },
102     {
103       id: "get_parking_lot",
104       name: "Get Parking Lot Items",
105       description:
106         "Retrieve all current parking lot items for a given conversation id",
107       tags: ["parking-lot", "list"],
108       examples: [ ...
109       ],
110     },
111   ],
112 });
113
114 new A2APlugin({
115   agentCard: standupAgentCard,
116   app: app,
117 });
118
119 // Teams Message Handler
120 const handleA2AMessage = async (message: A2AMessage) => {
121   // ...
122   return {
123     type: "A2A",
124     data: {
125       id: "add_parking_lot",
126       name: "Add to Parking Lot",
127       description: "Add a new item to the parking lot for discussion.",
128       tags: ["parking-lot", "add"],
129       examples: [ ...
130       ],
131     },
132   };
133 };
134
135 export { handleA2AMessage };
136
137 const app = new App({
138   name: "Standup Agent",
139   url: "http://localhost:3000/a2a",
140   provider: { ...
141   },
142   version: "0.0.1",
143   capabilities: {},
144   skills: [
145     {
146       id: "add_parking_lot",
147       name: "Add to Parking Lot",
148       description: "Add a new item to the parking lot for discussion.",
149       tags: ["parking-lot", "add"],
150       examples: [ ...
151       ],
152     },
153     {
154       id: "get_parking_lot",
155       name: "Get Parking Lot Items",
156       description:
157         "Retrieve all current parking lot items for a given conversation id",
158       tags: ["parking-lot", "list"],
159       examples: [ ...
160       ],
161     },
162   ],
163 });
164
165 new A2APlugin({
166   agentCard: standupAgentCard,
167   app: app,
168 });
169
170 // Teams Message Handler
171 const handleA2AMessage = async (message: A2AMessage) => {
172   // ...
173   return {
174     type: "A2A",
175     data: {
176       id: "add_parking_lot",
177       name: "Add to Parking Lot",
178       description: "Add a new item to the parking lot for discussion.",
179       tags: ["parking-lot", "add"],
180       examples: [ ...
181       ],
182     },
183   };
184 };
185
186 export { handleA2AMessage };
187
188 const app = new App({
189   name: "Standup Agent",
190   url: "http://localhost:3000/a2a",
191   provider: { ...
192   },
193   version: "0.0.1",
194   capabilities: {},
195   skills: [
196     {
197       id: "add_parking_lot",
198       name: "Add to Parking Lot",
199       description: "Add a new item to the parking lot for discussion.",
200       tags: ["parking-lot", "add"],
201       examples: [ ...
202       ],
203     },
204     {
205       id: "get_parking_lot",
206       name: "Get Parking Lot Items",
207       description:
208         "Retrieve all current parking lot items for a given conversation id",
209       tags: ["parking-lot", "list"],
210       examples: [ ...
211       ],
212     },
213   ],
214 });
215
216 new A2APlugin({
217   agentCard: standupAgentCard,
218   app: app,
219 });
220
221 // Teams Message Handler
222 const handleA2AMessage = async (message: A2AMessage) => {
223   // ...
224   return {
225     type: "A2A",
226     data: {
227       id: "add_parking_lot",
228       name: "Add to Parking Lot",
229       description: "Add a new item to the parking lot for discussion.",
230       tags: ["parking-lot", "add"],
231       examples: [ ...
232       ],
233     },
234   };
235 };
236
237 export { handleA2AMessage };
238
239 const app = new App({
240   name: "Standup Agent",
241   url: "http://localhost:3000/a2a",
242   provider: { ...
243   },
244   version: "0.0.1",
245   capabilities: {},
246   skills: [
247     {
248       id: "add_parking_lot",
249       name: "Add to Parking Lot",
250       description: "Add a new item to the parking lot for discussion.",
251       tags: ["parking-lot", "add"],
252       examples: [ ...
253       ],
254     },
255     {
256       id: "get_parking_lot",
257       name: "Get Parking Lot Items",
258       description:
259         "Retrieve all current parking lot items for a given conversation id",
260       tags: ["parking-lot", "list"],
261       examples: [ ...
262       ],
263     },
264   ],
265 });
266
267 new A2APlugin({
268   agentCard: standupAgentCard,
269   app: app,
270 });
271
272 // Teams Message Handler
273 const handleA2AMessage = async (message: A2AMessage) => {
274   // ...
275   return {
276     type: "A2A",
277     data: {
278       id: "add_parking_lot",
279       name: "Add to Parking Lot",
280       description: "Add a new item to the parking lot for discussion.",
281       tags: ["parking-lot", "add"],
282       examples: [ ...
283       ],
284     },
285   };
286 };
287
288 export { handleA2AMessage };
289
290 const app = new App({
291   name: "Standup Agent",
292   url: "http://localhost:3000/a2a",
293   provider: { ...
294   },
295   version: "0.0.1",
296   capabilities: {},
297   skills: [
298     {
299       id: "add_parking_lot",
300       name: "Add to Parking Lot",
301       description: "Add a new item to the parking lot for discussion.",
302       tags: ["parking-lot", "add"],
303       examples: [ ...
304       ],
305     },
306     {
307       id: "get_parking_lot",
308       name: "Get Parking Lot Items",
309       description:
310         "Retrieve all current parking lot items for a given conversation id",
311       tags: ["parking-lot", "list"],
312       examples: [ ...
313       ],
314     },
315   ],
316 });
317
318 new A2APlugin({
319   agentCard: standupAgentCard,
320   app: app,
321 });
322
323 // Teams Message Handler
324 const handleA2AMessage = async (message: A2AMessage) => {
325   // ...
326   return {
327     type: "A2A",
328     data: {
329       id: "add_parking_lot",
330       name: "Add to Parking Lot",
331       description: "Add a new item to the parking lot for discussion.",
332       tags: ["parking-lot", "add"],
333       examples: [ ...
334       ],
335     },
336   };
337 };
338
339 export { handleA2AMessage };
340
341 const app = new App({
342   name: "Standup Agent",
343   url: "http://localhost:3000/a2a",
344   provider: { ...
345   },
346   version: "0.0.1",
347   capabilities: {},
348   skills: [
349     {
350       id: "add_parking_lot",
351       name: "Add to Parking Lot",
352       description: "Add a new item to the parking lot for discussion.",
353       tags: ["parking-lot", "add"],
354       examples: [ ...
355       ],
356     },
357     {
358       id: "get_parking_lot",
359       name: "Get Parking Lot Items",
360       description:
361         "Retrieve all current parking lot items for a given conversation id",
362       tags: ["parking-lot", "list"],
363       examples: [ ...
364       ],
365     },
366   ],
367 });
368
369 new A2APlugin({
370   agentCard: standupAgentCard,
371   app: app,
372 });
373
374 // Teams Message Handler
375 const handleA2AMessage = async (message: A2AMessage) => {
376   // ...
377   return {
378     type: "A2A",
379     data: {
380       id: "add_parking_lot",
381       name: "Add to Parking Lot",
382       description: "Add a new item to the parking lot for discussion.",
383       tags: ["parking-lot", "add"],
384       examples: [ ...
385       ],
386     },
387   };
388 };
389
390 export { handleA2AMessage };
391
392 const app = new App({
393   name: "Standup Agent",
394   url: "http://localhost:3000/a2a",
395   provider: { ...
396   },
397   version: "0.0.1",
398   capabilities: {},
399   skills: [
400     {
401       id: "add_parking_lot",
402       name: "Add to Parking Lot",
403       description: "Add a new item to the parking lot for discussion.",
404       tags: ["parking-lot", "add"],
405       examples: [ ...
406       ],
407     },
408     {
409       id: "get_parking_lot",
410       name: "Get Parking Lot Items",
411       description:
412         "Retrieve all current parking lot items for a given conversation id",
413       tags: ["parking-lot", "list"],
414       examples: [ ...
415       ],
416     },
417   ],
418 });
419
420 new A2APlugin({
421   agentCard: standupAgentCard,
422   app: app,
423 });
424
425 // Teams Message Handler
426 const handleA2AMessage = async (message: A2AMessage) => {
427   // ...
428   return {
429     type: "A2A",
430     data: {
431       id: "add_parking_lot",
432       name: "Add to Parking Lot",
433       description: "Add a new item to the parking lot for discussion.",
434       tags: ["parking-lot", "add"],
435       examples: [ ...
436       ],
437     },
438   };
439 };
440
441 export { handleA2AMessage };
442
443 const app = new App({
444   name: "Standup Agent",
445   url: "http://localhost:3000/a2a",
446   provider: { ...
447   },
448   version: "0.0.1",
449   capabilities: {},
450   skills: [
451     {
452       id: "add_parking_lot",
453       name: "Add to Parking Lot",
454       description: "Add a new item to the parking lot for discussion.",
455       tags: ["parking-lot", "add"],
456       examples: [ ...
457       ],
458     },
459     {
460       id: "get_parking_lot",
461       name: "Get Parking Lot Items",
462       description:
463         "Retrieve all current parking lot items for a given conversation id",
464       tags: ["parking-lot", "list"],
465       examples: [ ...
466       ],
467     },
468   ],
469 });
470
471 new A2APlugin({
472   agentCard: standupAgentCard,
473   app: app,
474 });
475
476 // Teams Message Handler
477 const handleA2AMessage = async (message: A2AMessage) => {
478   // ...
479   return {
480     type: "A2A",
481     data: {
482       id: "add_parking_lot",
483       name: "Add to Parking Lot",
484       description: "Add a new item to the parking lot for discussion.",
485       tags: ["parking-lot", "add"],
486       examples: [ ...
487       ],
488     },
489   };
490 };
491
492 export { handleA2AMessage };
493
494 const app = new App({
495   name: "Standup Agent",
496   url: "http://localhost:3000/a2a",
497   provider: { ...
498   },
499   version: "0.0.1",
500   capabilities: {},
501   skills: [
502     {
503       id: "add_parking_lot",
504       name: "Add to Parking Lot",
505       description: "Add a new item to the parking lot for discussion.",
506       tags: ["parking-lot", "add"],
507       examples: [ ...
508       ],
509     },
510     {
511       id: "get_parking_lot",
512       name: "Get Parking Lot Items",
513       description:
514         "Retrieve all current parking lot items for a given conversation id",
515       tags: ["parking-lot", "list"],
516       examples: [ ...
517       ],
518     },
519   ],
520 });
521
522 new A2APlugin({
523   agentCard: standupAgentCard,
524   app: app,
525 });
526
527 // Teams Message Handler
528 const handleA2AMessage = async (message: A2AMessage) => {
529   // ...
530   return {
531     type: "A2A",
532     data: {
533       id: "add_parking_lot",
534       name: "Add to Parking Lot",
535       description: "Add a new item to the parking lot for discussion.",
536       tags: ["parking-lot", "add"],
537       examples: [ ...
538       ],
539     },
540   };
541 };
542
543 export { handleA2AMessage };
544
545 const app = new App({
546   name: "Standup Agent",
547   url: "http://localhost:3000/a2a",
548   provider: { ...
549   },
550   version: "0.0.1",
551   capabilities: {},
552   skills: [
553     {
554       id: "add_parking_lot",
555       name: "Add to Parking Lot",
556       description: "Add a new item to the parking lot for discussion.",
557       tags: ["parking-lot", "add"],
558       examples: [ ...
559       ],
560     },
561     {
562       id: "get_parking_lot",
563       name: "Get Parking Lot Items",
564       description:
565         "Retrieve all current parking lot items for a given conversation id",
566       tags: ["parking-lot", "list"],
567       examples: [ ...
568       ],
569     },
570   ],
571 });
572
573 new A2APlugin({
574   agentCard: standupAgentCard,
575   app: app,
576 });
577
578 // Teams Message Handler
579 const handleA2AMessage = async (message: A2AMessage) => {
580   // ...
581   return {
582     type: "A2A",
583     data: {
584       id: "add_parking_lot",
585       name: "Add to Parking Lot",
586       description: "Add a new item to the parking lot for discussion.",
587       tags: ["parking-lot", "add"],
588       examples: [ ...
589       ],
590     },
591   };
592 };
593
594 export { handleA2AMessage };
595
596 const app = new App({
597   name: "Standup Agent",
598   url: "http://localhost:3000/a2a",
599   provider: { ...
600   },
601   version: "0.0.1",
602   capabilities: {},
603   skills: [
604     {
605       id: "add_parking_lot",
606       name: "Add to Parking Lot",
607       description: "Add a new item to the parking lot for discussion.",
608       tags: ["parking-lot", "add"],
609       examples: [ ...
610       ],
611     },
612     {
613       id: "get_parking_lot",
614       name: "Get Parking Lot Items",
615       description:
616         "Retrieve all current parking lot items for a given conversation id",
617       tags: ["parking-lot", "list"],
618       examples: [ ...
619       ],
620     },
621   ],
622 });
623
624 new A2APlugin({
625   agentCard: standupAgentCard,
626   app: app,
627 });
628
629 // Teams Message Handler
630 const handleA2AMessage = async (message: A2AMessage) => {
631   // ...
632   return {
633     type: "A2A",
634     data: {
635       id: "add_parking_lot",
636       name: "Add to Parking Lot",
637       description: "Add a new item to the parking lot for discussion.",
638       tags: ["parking-lot", "add"],
639       examples: [ ...
640       ],
641     },
642   };
643 };
644
645 export { handleA2AMessage };
646
647 const app = new App({
648   name: "Standup Agent",
649   url: "http://localhost:3000/a2a",
650   provider: { ...
651   },
652   version: "0.0.1",
653   capabilities: {},
654   skills: [
655     {
656       id: "add_parking_lot",
657       name: "Add to Parking Lot",
658       description: "Add a new item to the parking lot for discussion.",
659       tags: ["parking-lot", "add"],
660       examples: [ ...
661       ],
662     },
663     {
664       id: "get_parking_lot",
665       name: "Get Parking Lot Items",
666       description:
667         "Retrieve all current parking lot items for a given conversation id",
668       tags: ["parking-lot", "list"],
669       examples: [ ...
670       ],
671     },
672   ],
673 });
674
675 new A2APlugin({
676   agentCard: standupAgentCard,
677   app: app,
678 });
679
680 // Teams Message Handler
681 const handleA2AMessage = async (message: A2AMessage) => {
682   // ...
683   return {
684     type: "A2A",
685     data: {
686       id: "add_parking_lot",
687       name: "Add to Parking Lot",
688       description: "Add a new item to the parking lot for discussion.",
689       tags: ["parking-lot", "add"],
690       examples: [ ...
691       ],
692     },
693   };
694 };
695
696 export { handleA2AMessage };
697
698 const app = new App({
699   name: "Standup Agent",
700   url: "http://localhost:3000/a2a",
701   provider: { ...
702   },
703   version: "0.0.1",
704   capabilities: {},
705   skills: [
706     {
707       id: "add_parking_lot",
708       name: "Add to Parking Lot",
709       description: "Add a new item to the parking lot for discussion.",
710       tags: ["parking-lot", "add"],
711       examples: [ ...
712       ],
713     },
714     {
715       id: "get_parking_lot",
716       name: "Get Parking Lot Items",
717       description:
718         "Retrieve all current parking lot items for a given conversation id",
719       tags: ["parking-lot", "list"],
720       examples: [ ...
721       ],
722     },
723   ],
724 });
725
726 new A2APlugin({
727   agentCard: standupAgentCard,
728   app: app,
729 });
730
731 // Teams Message Handler
732 const handleA2AMessage = async (message: A2AMessage) => {
733   // ...
734   return {
735     type: "A2A",
736     data: {
737       id: "add_parking_lot",
738       name: "Add to Parking Lot",
739       description: "Add a new item to the parking lot for discussion.",
740       tags: ["parking-lot", "add"],
741       examples: [ ...
742       ],
743     },
744   };
745 };
746
747 export { handleA2AMessage };
748
749 const app = new App({
750   name: "Standup Agent",
751   url: "http://localhost:3000/a2a",
752   provider: { ...
753   },
754   version: "0.0.1",
755   capabilities: {},
756   skills: [
757     {
758       id: "add_parking_lot",
759       name: "Add to Parking Lot",
760       description: "Add a new item to the parking lot for discussion.",
761       tags: ["parking-lot", "add"],
762       examples: [ ...
763       ],
764     },
765     {
766       id: "get_parking_lot",
767       name: "Get Parking Lot Items",
768       description:
769         "Retrieve all current parking lot items for a given conversation id",
770       tags: ["parking-lot", "list"],
771       examples: [ ...
772       ],
773     },
774   ],
775 });
776
777 new A2APlugin({
778   agentCard: standupAgentCard,
779   app: app,
780 });
781
782 // Teams Message Handler
783 const handleA2AMessage = async (message: A2AMessage) => {
784   // ...
785   return {
786     type: "A2A",
787     data: {
788       id: "add_parking_lot",
789       name: "Add to Parking Lot",
790       description: "Add a new item to the parking lot for discussion.",
791       tags: ["parking-lot", "add"],
792       examples: [ ...
793       ],
794     },
795   };
796 };
797
798 export { handleA2AMessage };
799
800 const app = new App({
801   name: "Standup Agent",
802   url: "http://localhost:3000/a2a",
803   provider: { ...
804   },
805   version: "0.0.1",
806   capabilities: {},
807   skills: [
808     {
809       id: "add_parking_lot",
810       name: "Add to Parking Lot",
811       description: "Add a new item to the parking lot for discussion.",
812       tags: ["parking-lot", "add"],
813       examples: [ ...
814       ],
815     },
816     {
817       id: "get_parking_lot",
818       name: "Get Parking Lot Items",
819       description:
820         "Retrieve all current parking lot items for a given conversation id",
821       tags: ["parking-lot", "list"],
822       examples: [ ...
823       ],
824     },
825   ],
826 });
827
828 new A2APlugin({
829   agentCard: standupAgentCard,
830   app: app,
831 });
832
833 // Teams Message Handler
834 const handleA2AMessage = async (message: A2AMessage) => {
835   // ...
836   return {
837     type: "A2A",
838     data: {
839       id: "add_parking_lot",
840       name: "Add to Parking Lot",
841       description: "Add a new item to the parking lot for discussion.",
842       tags: ["parking-lot", "add"],
843       examples: [ ...
844       ],
845     },
846   };
847 };
848
849 export { handleA2AMessage };
850
851 const app = new App({
852   name: "Standup Agent",
853   url: "http://localhost:3000/a2a",
854   provider: { ...
855   },
856   version: "0.0.1",
857   capabilities: {},
858   skills: [
859     {
860       id: "add_parking_lot",
861       name: "Add to Parking Lot",
862       description: "Add a new item to the parking lot for discussion.",
863       tags: ["parking-lot", "add"],
864       examples: [ ...
865       ],
866     },
867     {
868       id: "get_parking_lot",
869       name: "Get Parking Lot Items",
870       description:
871         "Retrieve all current parking lot items for a given conversation id",
872       tags: ["parking-lot", "list"],
873       examples: [ ...
874       ],
875     },
876   ],
877 });
878
879 new A2APlugin({
880   agentCard: standupAgentCard,
881   app: app,
882 });
883
884 // Teams Message Handler
885 const handleA2AMessage = async (message: A2AMessage) => {
886   // ...
887   return {
888     type: "A2A",
889     data: {
890       id: "add_parking_lot",
891       name: "Add to Parking Lot",
892       description: "Add a new item to the parking lot for discussion.",
893       tags: ["parking-lot", "add"],
894       examples: [ ...
895       ],
896     },
897   };
898 };
899
900 export { handleA2AMessage };
901
902 const app = new App({
903   name: "Standup Agent",
904   url: "http://localhost:3000/a2a",
905   provider: { ...
906   },
907   version: "0.0.1",
908   capabilities: {},
909   skills: [
910     {
911       id: "add_parking_lot",
912       name: "Add to Parking Lot",
913       description: "Add a new item to the parking lot for discussion.",
914       tags: ["parking-lot", "add"],
915       examples: [ ...
916       ],
917     },
918     {
919       id: "get_parking_lot",
920       name: "Get Parking Lot Items",
921       description:
922         "Retrieve all current parking lot items for a given conversation id",
923       tags: ["parking-lot", "list"],
924       examples: [ ...
925       ],
926     },
927   ],
928 });
929
930 new A2APlugin({
931   agentCard: standupAgentCard,
932   app: app,
933 });
934
935 // Teams Message Handler
936 const handleA2AMessage = async (message: A2AMessage) => {
937   // ...
938   return {
939     type: "A2A",
940     data: {
941       id: "add_parking_lot",
942       name: "Add to Parking Lot",
943       description: "Add a new item to the parking lot for discussion.",
944       tags: ["parking-lot", "add"],
945       examples: [ ...
946       ],
947     },
948   };
949 };
950
951 export { handleA2AMessage };
952
953 const app = new App({
954   name: "Standup Agent",
955   url: "http://localhost:3000/a2a",
956   provider: { ...
957   },
958   version: "0.0.1",
959   capabilities: {},
960   skills: [
961     {
962       id: "add_parking_lot",
963       name: "Add to Parking Lot",
964       description: "Add a new item to the parking lot for discussion.",
965       tags: ["parking-lot", "add"],
966       examples: [ ...
967       ],
968     },
969     {
970       id: "get_parking_lot",
971       name: "Get Parking Lot Items",
972       description:
973         "Retrieve all current parking lot items for a given conversation id",
974       tags: ["parking-lot", "list"],
975       examples: [ ...
976       ],
977     },
978   ],
979 });
980
981 new A2APlugin({
982   agentCard: standupAgentCard,
983   app: app,
984 });
985
986 // Teams Message Handler
987 const handleA2AMessage = async (message: A2AMessage) => {
988   // ...
989   return {
990     type: "A2A",
991     data: {
992       id: "add_parking_lot",
993       name: "Add to Parking Lot",
994       description: "Add a new item to the parking lot for discussion.",
995       tags: ["parking-lot", "add"],
996       examples: [ ...
997       ],
998     },
999   };
1000 };
1001
1002 export { handleA2AMessage };
1003
1004 const app = new App({
1005   name: "Standup Agent",
1006   url: "http://localhost:3000/a2a",
1007   provider: { ...
1008   },
1009   version: "0.0.1",
1010   capabilities: {},
1011   skills: [
1012     {
1013       id: "add_parking_lot",
1014       name: "Add to Parking Lot",
1015       description: "Add a new item to the parking lot for discussion.",
1016       tags: ["parking-lot", "add"],
1017       examples: [ ...
1018       ],
1019     },
1020     {
1021       id: "get_parking_lot",
1022       name: "Get Parking Lot Items",
1023       description:
1024         "Retrieve all current parking lot items for a given conversation id",
1025       tags: ["parking-lot", "list"],
1026       examples: [ ...
1027       ],
1028     },
1029   ],
1030 });
1031
1032 new A2APlugin({
1033   agentCard: standupAgentCard,
1034   app: app,
1035 });
1036
1037 // Teams Message Handler
1038 const handleA2AMessage = async (message: A2AMessage) => {
1039   // ...
1040   return {
1041     type: "A2A",
1042     data: {
1043       id: "add_parking_lot",
1044       name: "Add to Parking Lot",
1045       description: "Add a new item to the parking lot for discussion.",
1046       tags: ["parking-lot", "add"],
1047       examples: [ ...
1048       ],
1049     },
1050   };
1051 };
1052
1053 export { handleA2AMessage };
1054
1055 const app = new App({
1056   name: "Standup Agent",
1057   url: "http://localhost:3000/a2a",
1058   provider: { ...
1059   },
1060   version: "0.0.1",
1061   capabilities: {},
1062   skills: [
1063     {
1064       id: "add_parking_lot",
1065       name: "Add to Parking Lot",
1066       description: "Add a new item to the parking lot for discussion.",
1067       tags: ["parking-lot", "add"],
1068       examples: [ ...
1069       ],
1070     },
1071     {
1072       id: "get_parking_lot",
1073       name: "Get Parking Lot Items",
1074       description:
1075         "Retrieve all current parking lot items for a given conversation id",
1076       tags: ["parking-lot", "list"],
1077       examples: [ ...
1078       ],
1079     },
1080   ],
1081 });
1082
1083 new A2APlugin({
1084   agentCard: standupAgentCard,
1085   app: app,
1086 });
1087
1088 // Teams Message Handler
1089 const handleA2AMessage = async (message: A2AMessage) => {
1090   // ...
1091   return {
1092     type: "A2A",
1093     data: {
1094       id: "add_parking_lot",
1095       name: "Add to Parking Lot",
1096       description: "Add a new item to the parking lot for discussion.",
1097       tags: ["parking-lot", "add"],
1098       examples: [ ...
1099       ],
1100     },
1101   };
1102 };
1103
1104 export { handleA2AMessage };
1105
1106 const app = new App({
1107   name: "Standup Agent",
1108   url: "http://localhost:3000/a2a",
1109   provider: { ...
1110   },
1111   version: "0.0.1",
1112   capabilities: {},
1113   skills: [
1114     {
1115       id: "add_parking_lot",
1116       name: "Add to Parking Lot",
1117       description: "Add a new item to the parking lot for discussion.",
1118       tags: ["parking-lot", "add"],
1119       examples: [ ...
1120       ],
1121     },
1122     {
1123       id: "get_parking_lot",
1124       name: "Get Parking Lot Items",
1125       description:
1126         "Retrieve all current parking lot items for a given conversation id",
1127       tags: ["parking-lot", "list"],
1128       examples: [ ...
1129       ],
1130     },
1131   ],
1132 });
1133
1134 new A2APlugin({
1135   agentCard: standupAgentCard,
1136   app: app,
1137 });
1138
1139 // Teams Message Handler
1140 const handleA2AMessage = async (message: A2AMessage) => {
1141   // ...
1142   return {
1143     type: "A2A",
1144     data: {
1145       id: "add_parking_lot",
1146       name: "Add to Parking Lot",
1147       description: "Add a new item to the parking lot for discussion.",
1148       tags: ["parking-lot", "add"],
1149       examples: [ ...
1150       ],
1151     },
1152   };
1153 };
1154
1155 export { handleA2AMessage };
1156
1157 const app = new App({
1158   name: "Standup Agent",
1159   url: "http://localhost:3000/a2a",
1160   provider: { ...
1161   },
1162   version: "0.0.1",
1163   capabilities: {},
1164   skills: [
1165     {
1166       id: "add_parking_lot",
1167       name: "Add to Parking Lot",
1168       description: "Add a new item to the parking lot for discussion.",
1169       tags: ["parking-lot", "add"],
1170       examples: [ ...
1171       ],
1172     },
1173     {
1174       id: "get_parking_lot",
1175       name: "Get Parking Lot Items",
1176       description:
1177         "Retrieve all current parking lot items for a given conversation id",
1178       tags: ["parking-lot", "list"],
1179       examples: [ ...
1180       ],
1181     },
1182   ],
1183 });
1184
1185 new A2APlugin({
1186   agentCard: standupAgentCard,
1187   app: app,
1188 });
1189
1190 // Teams Message Handler
1191 const handleA2AMessage = async (message: A2AMessage) => {
1192   // ...
1193   return {
1194     type: "A2A",
1195     data: {
1196       id: "add_parking_lot",
1197       name: "Add to Parking Lot",
1198       description: "Add a new item to the parking lot for discussion.",
1199       tags: ["parking-lot", "add"],
1200       examples: [ ...
1201       ],
1202     },
1203   };
1204 };
1205
1206 export { handleA2AMessage };
1207
1208 const app = new App({
1209   name: "Standup Agent",
1210   url: "http://localhost:3000/a2a",
1211   provider: { ...
1212   },
1213  
```

モジュール1 目次

- AIエージェントとは？
- 従来のチャットボットとAIエージェントは何が違うのか？
- なぜ今AIエージェントが着目されているのか？
- エージェントはビジネスでどのように役に立つのか？
- お客様事例
- AIエージェントの内部構成
- まとめ

エージェントはビジネスでどう役に立つのか？

- 業務の自動化、情報収集、意思決定支援などに活用可能
- カスタマーサポートや営業支援など幅広い分野で導入が進んでいる
- 人の作業を補完し、生産性を大きく向上させる

モジュール1 目次

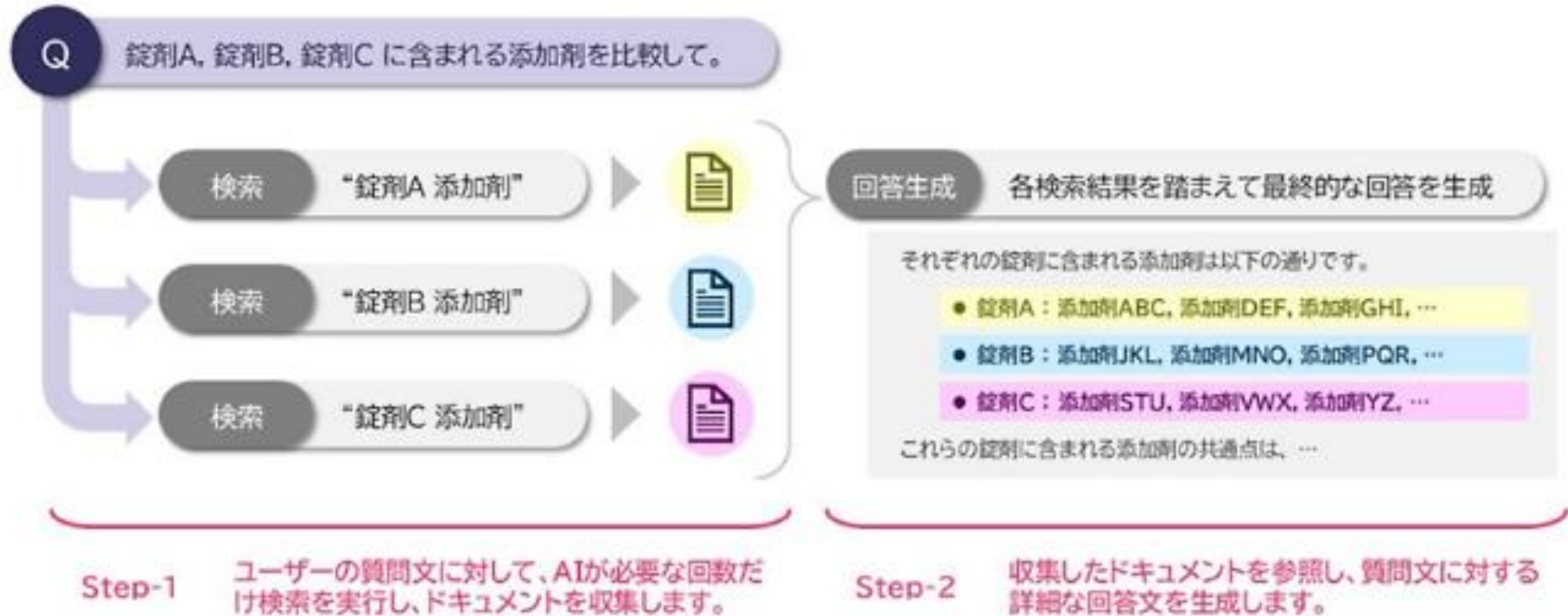
- AIエージェントとは？
- 従来のチャットボットとAIエージェントは何が違うのか？
- なぜ今AIエージェントが着目されているのか？
- エージェントはビジネスでどのように役に立つのか？
- お客様事例
- AIエージェントの内部構成
- まとめ

お客様事例: 沢井製薬様

- 「ベテラン社員のようなAIエージェントを作りたい」「研究者の相談相手として使用したい」という社内の要望を受けて、**Azure OpenAI Service**を使用してエージェントを実装
- AIエージェントが自律的に質問を解釈し、RAG（取得拡張生成）を使用して社内データから回答を探して回答
- ユーザーの入力が不足している場合は、『何の素材に対する試験結果ですか?』など質問を聞き返し、フォローアップする工夫も行っている
- 2024年8月から、「相談相手に生成AIを活用する仕組み」を製剤研究部全体に展開。現在、同部の全研究者50人が使える状態となっており、部門全体で利用頻度は100回/月に及ぶ

自律制御型RAG (Agent)

ユーザーの質問に対して、AI が“何をすれば最適な答えが得られるか”を判断。必要な回数だけ検索を繰り返し、回答を生成します。質問に対して1回のみ検索を実行する従来手法とは異なり、比較や列挙といった複雑な質問にも柔軟に対応できます。

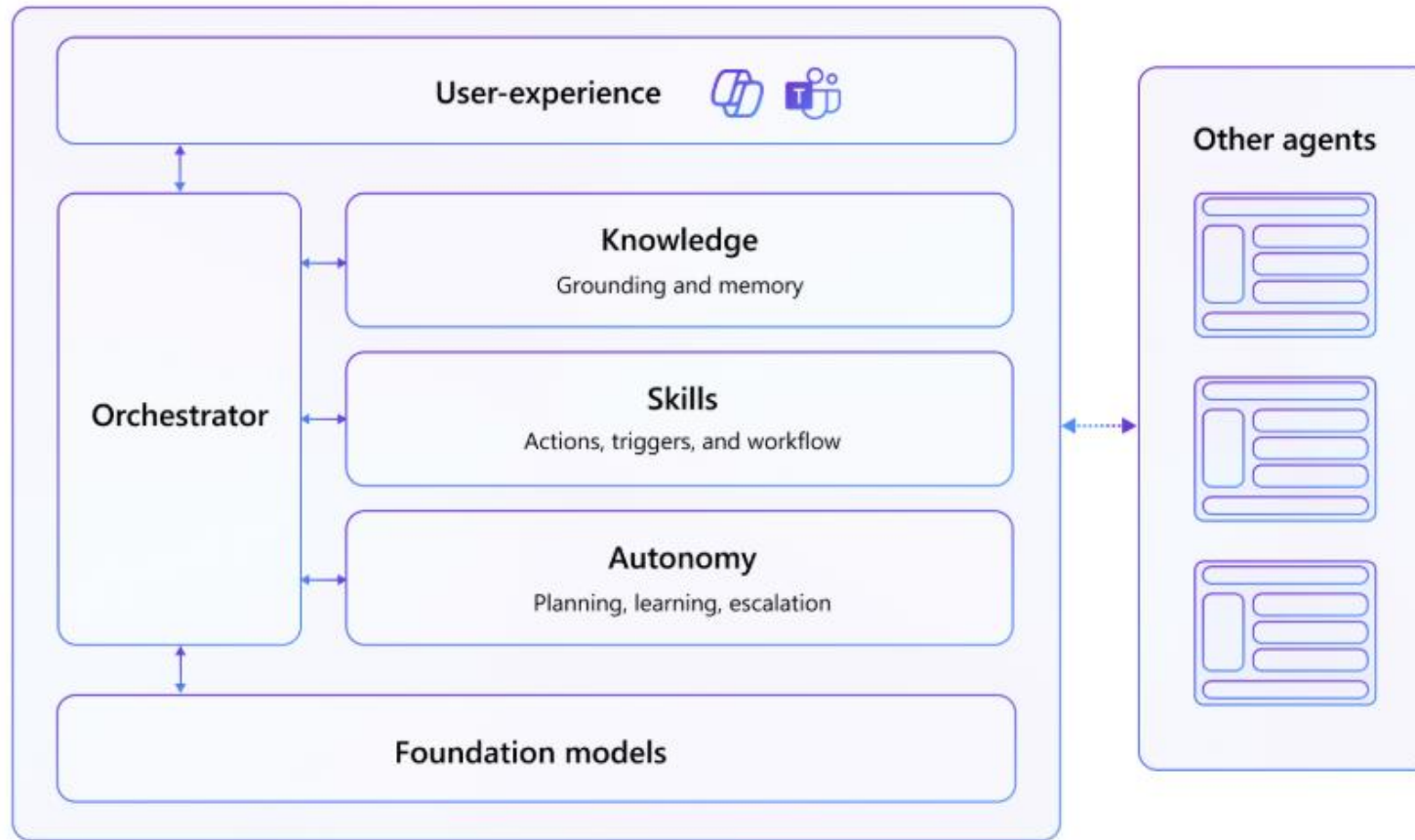


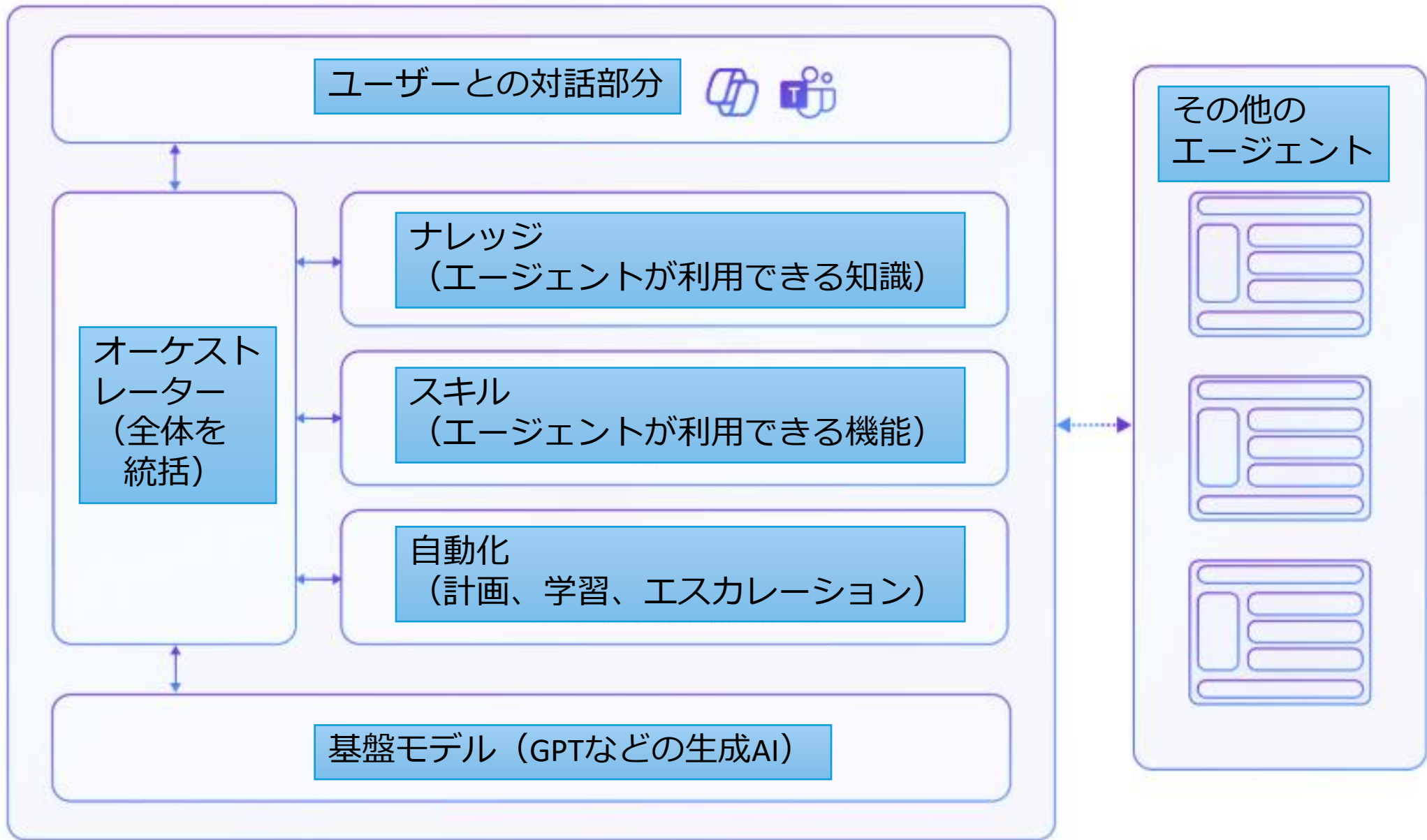
自立制御型 RAG (Agent) の概要図。ベーシック提供

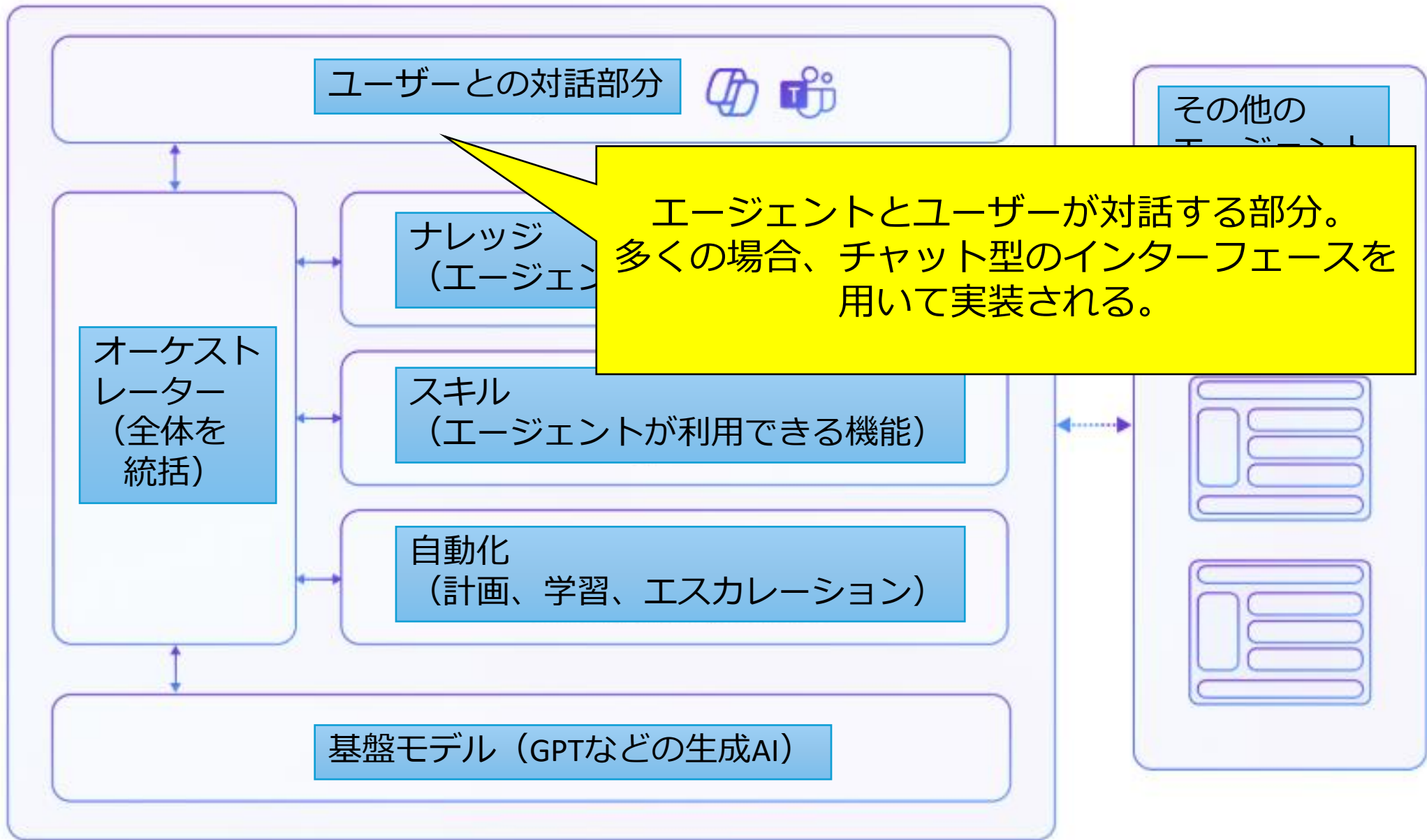
モジュール1 目次

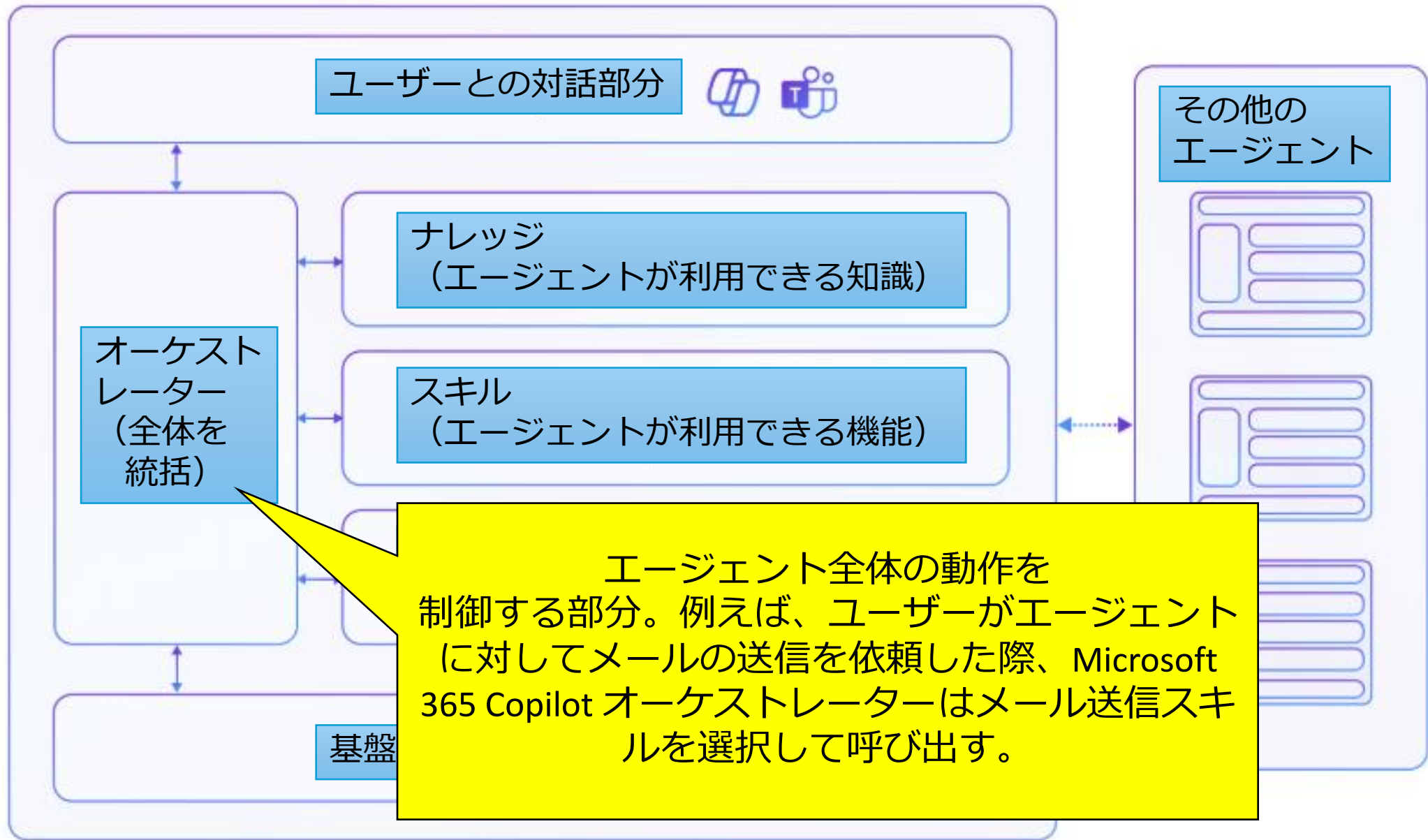
- AIエージェントとは？
- 従来のチャットボットとAIエージェントは何が違うのか？
- なぜ今AIエージェントが着目されているのか？
- エージェントはビジネスでどのように役に立つのか？
- お客様事例
- AIエージェントの内部構成
- まとめ

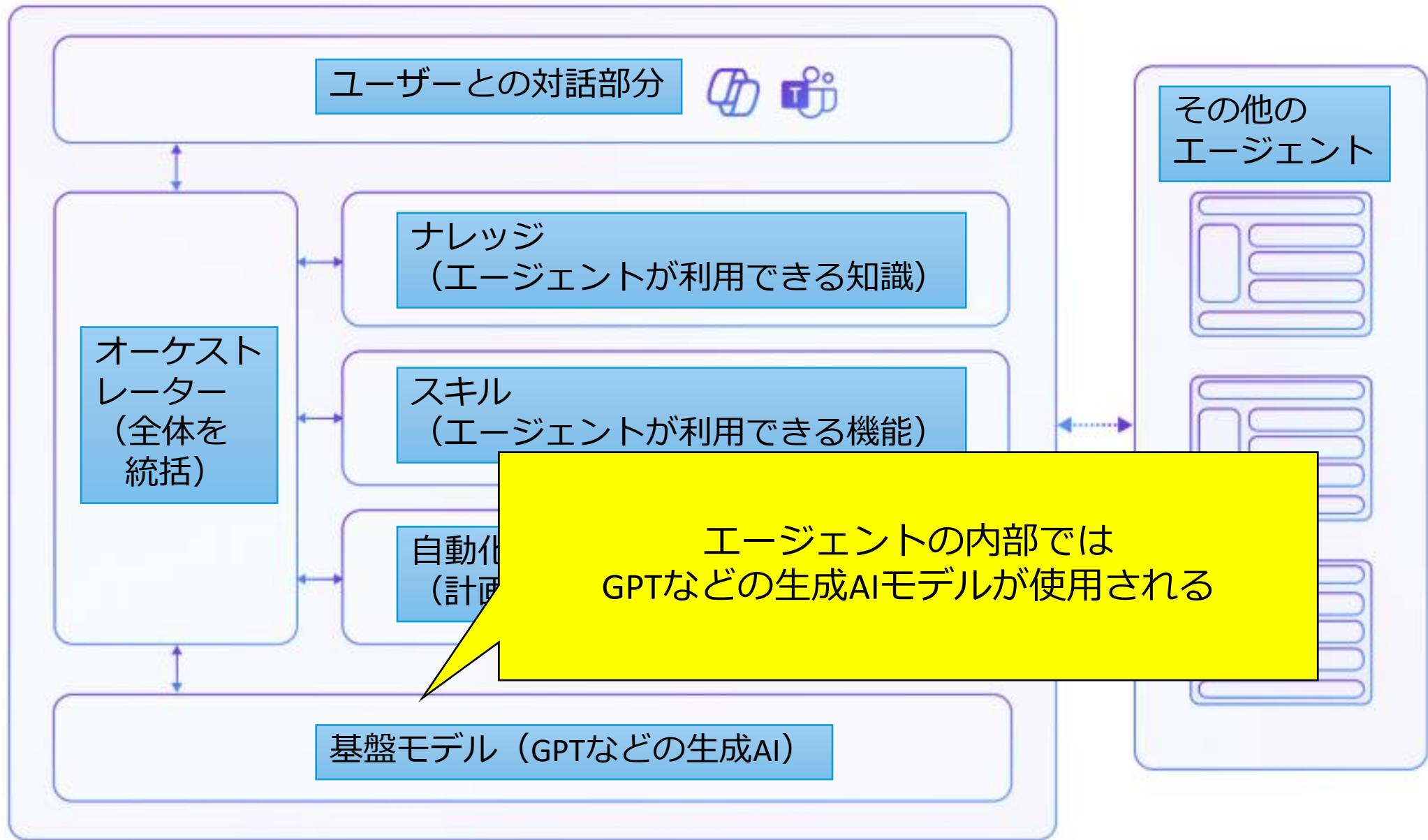
AIエージェントの内部構成

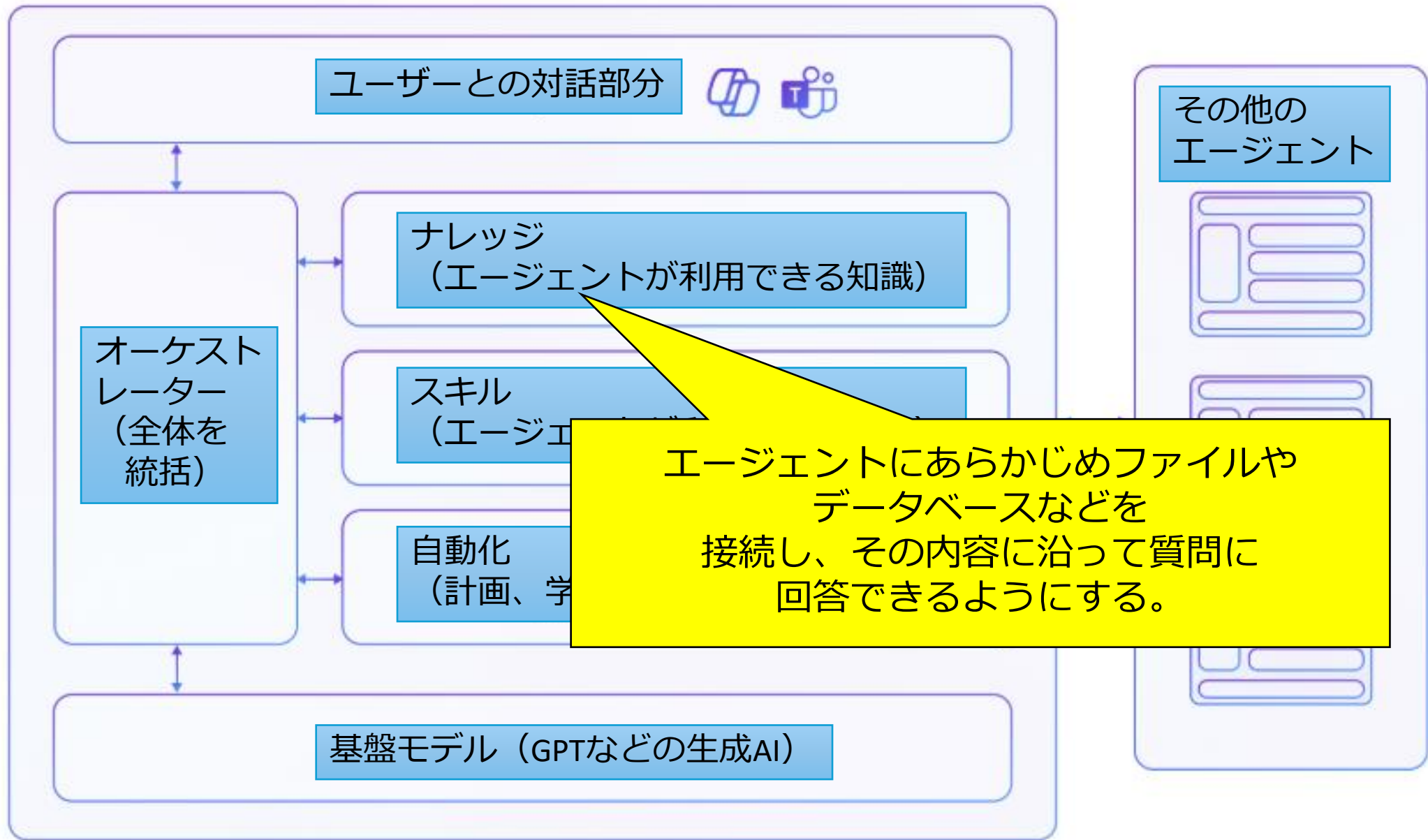


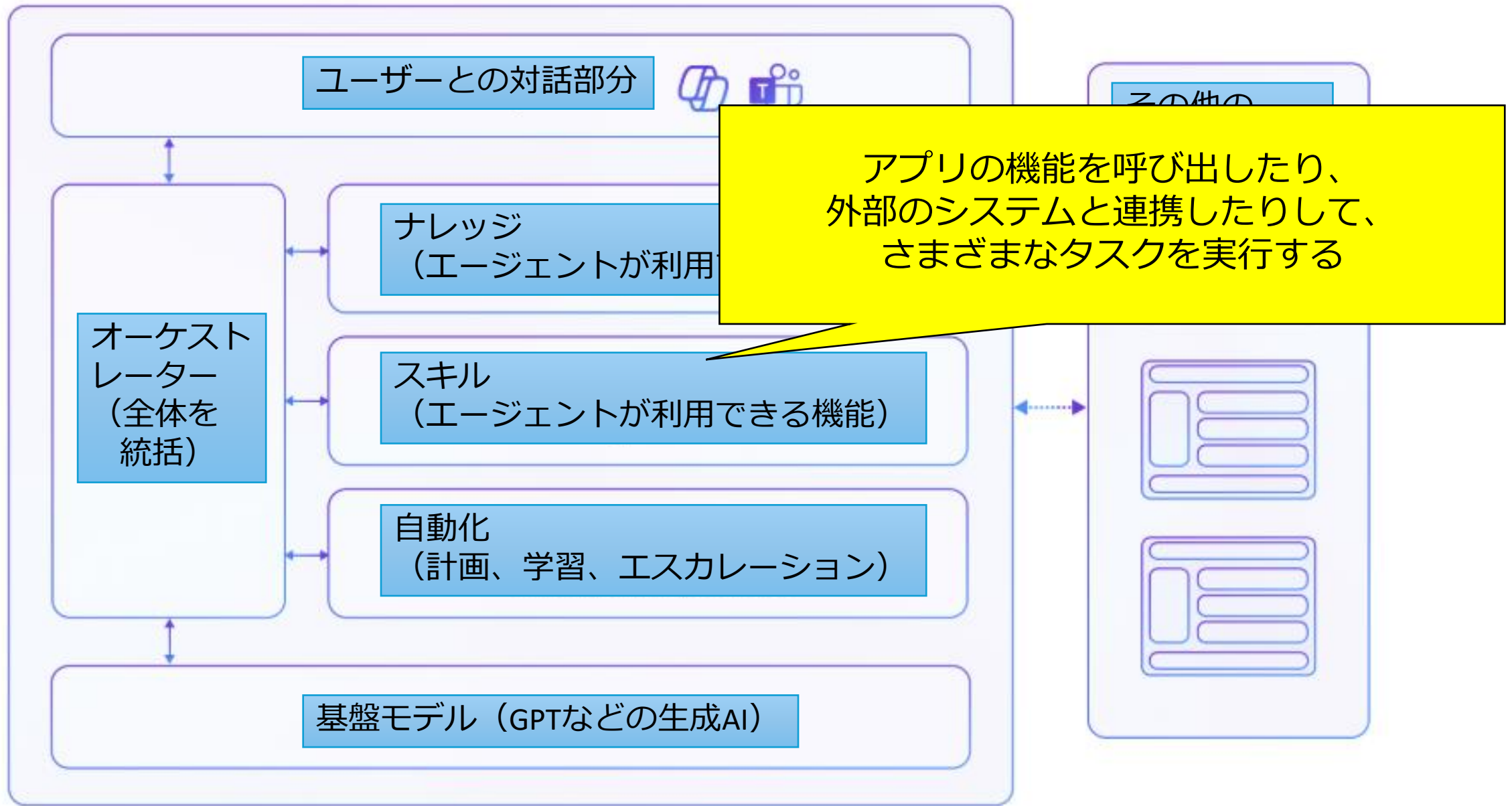


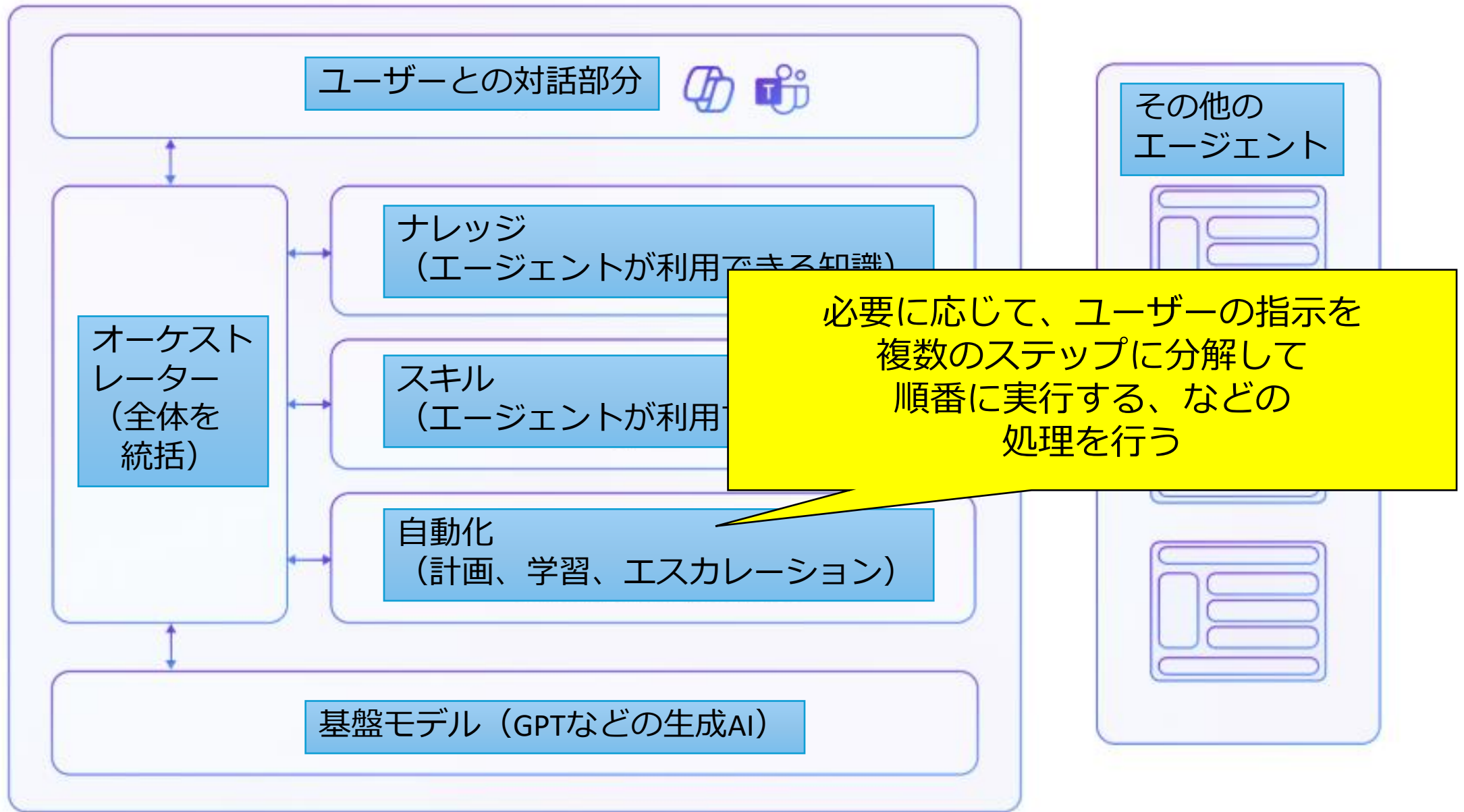


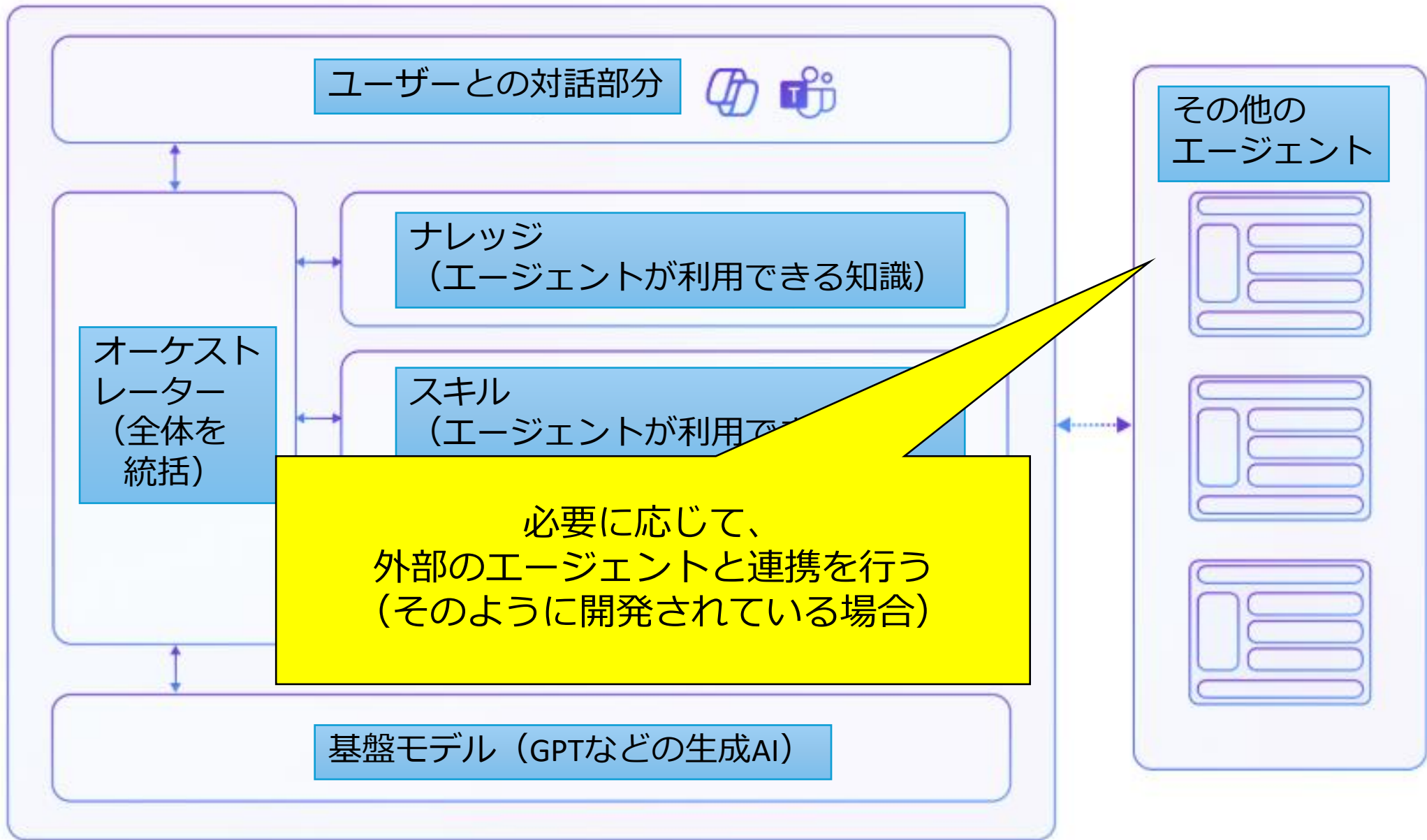












モジュール1 目次

- AIエージェントとは？
- 従来のチャットボットとAIエージェントは何が違うのか？
- なぜ今AIエージェントが着目されているのか？
- エージェントはビジネスでどのように役に立つのか？
- お客様事例
- AIエージェントの内部構成
- まとめ

モジュール1 AIエージェントの概要

まとめ

AIエージェントとは？	目的達成のために自律的に行動するAIソフトウェア。単に質問への回答を行うだけではなく、問題解決に向けて能動的に動くもの。
従来のチャットボットとAI エージェントとの違い	従来のチャットボットはあらかじめプログラミングされたことしかできなかったが、AIエージェントは生成AIを利用してユーザーの指示を柔軟に理解して作業を実行できる。
なぜ今AIエージェントが注目されているのか？	生成AIモデルの進歩、エージェントを実装するためのMCP・A2Aなどの技術の登場、AzureのAIサービスの進化などにより、AIエージェントを開発・運用できる下地が整った。実際のビジネスでのエージェント開発・運用の事例もある。
AIエージェントはビジネスでどう役に立つのか？	業務の自動化、情報収集、意思決定支援などの場面でエージェントが役立つ。たとえばベテラン社員に近い回答を行うことができる社内用AIエージェントなどが実際に利用されている。
AIエージェントの構成	オーケストレーター、生成AIモデル、接続されたナレッジやスキルなどから構成される。