

Lab 076 : Microsoft Agent Framework ? De SK à MAF

Niveau : L200 Parcours : Tous les parcours Durée : ~75 min ? Coût : Gratuit ? Utilise des données de migration simulées

Ce que vous apprendrez

Comment Semantic Kernel et AutoGen sont unifiés dans le Microsoft Agent Framework (MAF) Ce que sont les Agent Skills (packages de compétences portables .md) et comment ils permettent la réutilisation Comment les Graph Workflows (DAG avec points de contrôle) remplacent les pipelines linéaires Analyser une matrice de migration comparant 15 fonctionnalités entre SK, AutoGen et MAF Identifier les niveaux d'effort de migration et les fonctionnalités exclusives à MAF

Introduction

Le Microsoft Agent Framework (MAF) unifie Semantic Kernel et AutoGen en une seule plateforme cohérente pour la création d'agents IA. Publié en tant que Release Candidate en février 2026, MAF réunit le meilleur des deux mondes :

Le système de plugins, les planificateurs et les connecteurs d'entreprise de Semantic Kernel Les conversations multi-agents, l'exécution de code et les modèles de discussion de groupe d'AutoGen

Installation

[diagram or code sample]

Concepts clés

Concept | Description | Agent Skills | Packages de compétences portables .md qui définissent les capacités, entrées, sorties et dépendances d'un agent ? partageables entre équipes et projets | Graph Workflows | Orchestration basée sur des DAG avec points de contrôle, réessais et embranchements ? remplaçant les pipelines linéaires | DevUI | Interface de développement intégrée pour déboguer les conversations d'agents, inspecter l'exécution des compétences et visualiser les workflows | Unified API | Surface d'API unique qui remplace le Kernel de SK et l'AssistantAgent d'AutoGen par une classe commune Agent |

Le scénario

Vous êtes un ingénieur de plateforme dans une entreprise qui a créé des agents avec Semantic Kernel et AutoGen. La direction a décidé de migrer vers MAF. Vous disposez d'une matrice de migration (migration_matrix.csv) qui associe 15 fonctionnalités entre les trois frameworks ? en suivant la disponibilité, l'effort de migration et les fonctionnalités exclusives à MAF.

Votre mission : analyser la matrice, identifier les gains rapides, signaler les défis et construire un plan de migration.

info "Données simulées" Ce lab utilise un fichier CSV de matrice de migration simulé. Les données reflètent la correspondance réelle des fonctionnalités entre Semantic Kernel, AutoGen et MAF, telle que documentée dans les guides de migration.

Prérequis

Exigence | Pourquoi | Python 3.10+ | Exécuter les scripts d'analyse | Bibliothèque pandas | Manipulation des données |

[diagram or code sample]

tip "Démarrage rapide avec GitHub Codespaces" Open in GitHub Codespaces

Toutes les dépendances sont pré-installées dans le devcontainer.

? Fichiers de support

note "Téléchargez ces fichiers avant de commencer le lab" Enregistrez tous les fichiers dans un dossier lab-076/ de votre répertoire de travail.

Fichier | Description | Téléchargement | [broken_migration.py](#) | Exercice de correction de bugs (3 bugs + auto-tests) | ? Télécharger | [migration_matrix.csv](#) | Comparaison de 15 fonctionnalités : SK vs AutoGen vs MAF | ? Télécharger |

Étape 1 : Comprendre la matrice de migration

La matrice de migration associe 15 fonctionnalités entre trois frameworks. Chaque ligne représente une capacité :

Colonne | Description | feature | La capacité comparée (ex. : plugins, multi_agent_chat) |
sk_support | Si Semantic Kernel prend en charge cette fonctionnalité : yes, partial ou no |
autogen_support | Si AutoGen prend en charge cette fonctionnalité : yes, partial ou no |
maf_support | Si MAF prend en charge cette fonctionnalité : yes, partial ou no |
migration_effort | Effort de migration de SK/AutoGen vers MAF : low, medium ou high |
category | Catégorie de la fonctionnalité : core, orchestration, tooling ou integration |

Étape 2 : Charger et explorer la matrice

[diagram or code sample]

Sortie attendue :

[diagram or code sample]

Étape 3 : Identifier les gains rapides (effort de migration faible)

Les fonctionnalités avec un effort de migration low sont vos gains rapides ? commencez par celles-ci :

[diagram or code sample]

tip "Stratégie de migration" Commencez par les fonctionnalités à faible effort pour renforcer la confiance de l'équipe et démontrer l'API unifiée de MAF. Celles-ci ont généralement des équivalents directs dans SK ou AutoGen, ce qui rend la migration simple.

Étape 4 : Trouver les fonctionnalités exclusives à MAF

Quelles fonctionnalités existent dans MAF mais pas dans Semantic Kernel ?

[diagram or code sample]

[diagram or code sample]

Étape 5 : Analyser les migrations à effort élevé

Ces fonctionnalités nécessitent le plus de planification :

[diagram or code sample]

warning "Risque de migration" Les fonctionnalités à effort élevé impliquent souvent des changements architecturaux ? par exemple, remplacer l'orchestration personnalisée par des Graph Workflows ou convertir des plugins propriétaires en Agent Skills. Prévoyez 2 à 4 semaines par fonctionnalité à effort élevé.

Étape 6 : Construire le plan de migration

[diagram or code sample]

? Exercice de correction de bugs

Le fichier lab-076/broken_migration.py contient 3 bugs qui produisent une analyse de migration incorrecte. Pouvez-vous les trouver et les corriger tous ?

Exécutez les auto-tests pour voir lesquels échouent :

[diagram or code sample]

Vous devriez voir 3 tests échoués. Chaque test correspond à un bug :

Test | Ce qu'il vérifie | Indice | Test 1 | Nombre de fonctionnalités à faible effort |
Devrait filtrer migration_effort == "low", pas "high" | Test 2 | Nombre de fonctionnalités MAF uniquement vs SK | Devrait vérifier sk_support == "no", pas "yes" | Test 3 | Nombre total de fonctionnalités | Devrait utiliser len(df), pas len(df.columns) |

Corrigez les 3 bugs, puis relancez. Quand vous voyez All passed!, c'est terminé !

? Vérification des connaissances

question "Q1 (Choix multiple) : Qu'est-ce que le Microsoft Agent Framework (MAF) ?"

- A) Une nouvelle version d'AutoGen avec un nom différent - B) Un framework unifié combinant Semantic Kernel et AutoGen en une seule plateforme - C) Un service cloud uniquement pour exécuter des agents sur Azure - D) Un remplacement de LangChain
??? success "?" Révéler la réponse" Correct : B) Un framework unifié combinant Semantic Kernel et AutoGen en une seule plateforme

MAF fusionne les forces des deux frameworks : le système de plugins d'entreprise et les planificateurs de SK avec les conversations multi-agents et l'exécution de code d'AutoGen. Il fournit une classe unique Agent, des Agent Skills portables et des Graph Workflows basés sur des DAG.

question "Q2 (Choix multiple) : Que sont les Agent Skills dans MAF ?"

- A) Des fonctions Python décorées avec @skill - B) Des packages de compétences portables .md qui définissent les capacités, entrées, sorties et dépendances - C) Des poids de modèle pré-entraînés pour des tâches spécifiques - D) Des Azure Functions que les agents peuvent appeler

??? success "?" Révéler la réponse" Correct : B) Des packages de compétences portables .md qui définissent les capacités, entrées, sorties et dépendances

Les Agent Skills sont des packages basés sur Markdown qui décrivent ce qu'un agent peut faire, les entrées dont il a besoin, les sorties qu'il produit et les dépendances requises. Ils sont partageables entre équipes et projets, permettant un marketplace de capacités d'agents réutilisables.

question "Q3 (Exécutez le lab) : Combien de fonctionnalités ont un effort de migration « low » ?"

Exécutez l'analyse de l'étape 3 sur ? migration_matrix.csv et comptez les fonctionnalités à faible effort.

??? success "?" Révéler la réponse" 7 fonctionnalités

Sur les 15 fonctionnalités de la matrice de migration, 7 ont migration_effort = "low". Ce sont les gains rapides de la phase 1 de la migration ? généralement des fonctionnalités avec des équivalents directs entre SK/AutoGen et MAF.

question "Q4 (Exécutez le lab) : Combien de fonctionnalités sont disponibles dans MAF mais PAS dans Semantic Kernel ?"

Exécutez l'analyse de l'étape 4 pour trouver les fonctionnalités où maf_support = "yes" et sk_support = "no".

??? success "?" Révéler la réponse" Le nombre de fonctionnalités où MAF a le support yes mais SK a le support no.

Celles-ci représentent les capacités que les équipes gagnent en migrant de SK vers MAF ? comme les modèles de discussion multi-agents, les bacs à sable d'exécution de code et d'autres fonctionnalités issues d'AutoGen désormais disponibles dans le framework unifié.

question "Q5 (Exécutez le lab) : Combien de fonctionnalités au total sont suivies dans la matrice de migration ?"

Chargez le CSV et vérifiez le nombre total de lignes.

??? success "?" Révéler la réponse" 15 fonctionnalités

La matrice de migration suit 15 fonctionnalités réparties en 4 catégories : core (5), orchestration (4), tooling (3) et integration (3). Chaque fonctionnalité est évaluée pour le support dans SK, AutoGen et MAF, ainsi que l'effort de migration.

Résumé

Sujet | Ce que vous avez appris | Microsoft Agent Framework | Plateforme unifiée fusionnant Semantic Kernel et AutoGen (RC fév. 2026) | Agent Skills | Packages de compétences

portables .md pour des capacités d'agents réutilisables | Graph Workflows | Orchestration basée sur des DAG avec points de contrôle, remplaçant les pipelines linéaires | Matrice de migration | 15 fonctionnalités comparées entre SK, AutoGen et MAF | Stratégie de migration | Commencer par le faible effort (7 fonctionnalités), planifier l'effort élevé (3 fonctionnalités) | DevUI | Interface de développement intégrée pour déboguer et visualiser les workflows d'agents |

Prochaines étapes

Lab 034 ? Multi-Agent avec Semantic Kernel (comprendre ce que vous migrez) Lab 036 ? Bases d'AutoGen (comprendre les modèles AutoGen avant MAF) Lab 074 ? Foundry Agent Service (exploiter et surveiller des flottes d'agents en production) Lab 073 ? SWE-Bench (évaluer les agents MAF sur des tâches de codage réelles)

Modernisation de Phase 3

warning "Verified 2026-10-27" Utilisez Microsoft Agent Framework 1.0 pour les nouveaux agents. Semantic Kernel reste une source de migration héritée et AutoGen vise la maintenance. Remplacez les planners obsolètes par workflows, outils, mémoire, checkpoints et arrêts explicites.

[diagram or code sample]

Microsoft Agent Framework 1.0 GA AutoGen to Agent Framework migration

Phase 3 · Simulation / Parcours Réel

Simulation Parcours Réel

Preserve the existing free simulation and its fixtures. Estimated cost: US\$0.

Reuse the same acceptance criteria against the documented hosted service. Before running, record a service-specific estimate and a hard spending cap. After validation: Supprimez déploiements, points de terminaison, index, traces et données de test après validation.

Kit d'accompagnement

Le build du site génère des téléchargements PDF et ZIP stables. Le ZIP contient le guide du lab et tous les fichiers d'accompagnement déclarés.

Télécharger le PDF généré Télécharger le ZIP généré

Supporting file: broken_migration.py

```
/usr/bin/env python3 """Bug-fix: Broken Migration Analyzer - Lab 076.""" import io, pandas as pd def count_low_effort(matrix): return (matrix["migration_effort"] == "high").sum() def features_in_maf_only(matrix): return len(matrix[matrix["semantic_kernel"] == "not supported"]) def migration_coverage(matrix): return matrix["agent_framework"].nunique() / len(matrix) * 100 def run_tests(): csv="feature,semantic_kernel,agent_framework,migration_effort\nPlugins,KernelPlugin,AgentSkills,medium\nA2A,not supported,native\nA2A,n/a\nStreaming,IAsyncEnumerable,built-in,low\nDevUI,not available,DevUI,low" m=pd.read_csv(io.StringIO(csv));p,f=0,0 n=count_low_effort(m);exp=2 if n==exp:print(f"OK T1: {n}");p+=1 else:print(f"FAIL T1: {n} (exp {exp})");f+=1 a=features_in_maf_only(m);exp=1 if a==exp:print(f"OK T2: {a}");p+=1 else:print(f"FAIL T2: {a} (exp {exp})");f+=1 c=migration_coverage(m);exp=100.0 if abs(c-exp)<0.1:print(f"OK T3: {c:.0f}%");p+=1 else:print(f"FAIL T3: {c:.0f}% (exp {exp:.0f}%");f+=1 print(f"\n{'All passed!' if f==0 else f'{f} failed'}") if __name__=="__main__":print("Tests:\n");run_tests()
```

Supporting file: migration_matrix.csv

feature,semantic_kernel,autogen,agent_framework,migration_effort Agent Abstraction,ChatCompletionAgent,AssistantAgent,Agent (unified),low Plugin

System,KernelPlugin,Tools,AgentSkills (.md files),medium
Multi-Agent,AgentGroupChat,GroupChat / Swarm,Graph Workflows,medium Orchestration,manual
loop,round_robin / selector,DAG with data flows,high Streaming,IAsyncEnumerable /
AsyncGenerator,streaming parameter,built-in with checkpointing,low
Memory,SemanticTextMemory,ConversableAgent memory,pluggable memory providers,medium
Human-in-the-Loop,manual implementation,human_input_mode,built-in graph nodes,low
Observability,OpenTelemetry manual,logging,DevUI + OpenTelemetry built-in,low MCP
Integration,via plugin,via tools,native MCP support,low A2A Integration,not supported,not
supported,native A2A support,n/a AG-UI Integration,not supported,not supported,native AG-UI
support,n/a Model Support,OpenAI/Azure/Ollama,OpenAI/Azure,OpenAI/Azure/Anthropic/GitHub/Loc
al,low Language Support,.NET and Python,Python (.NET experimental),.NET and Python (stable
both),low State Management,manual,manual,checkpointing + time-travel,medium DevUI,not
available,AutoGen Studio,DevUI (built-in),low